WMT 2021

Sixth Conference on Machine Translation

Proceedings of the Conference

November 10-11, 2021

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 978-1-954085-94-7

Preface

The Sixth Conference on Machine Translation (WMT 2021) took place on Wednesday, November 10 and Thursday, November 11, 2021 immediately following the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021).

This is the sixth time WMT has been held as a conference. The first time WMT was held as a conference was at ACL 2016 in Berlin, Germany, the second time at EMNLP 2017 in Copenhagen, Denmark, the third time at EMNLP 2018 in Brussels, Belgium, the fourth time at ACL 2019 in Florence, Italy, and the fifth time at EMNLP-2020, which was held as an online event due to the COVID-19 pandemic. 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, 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 13 shared tasks. These consisted of 10 translation tasks: Machine Translation of News, Similar Language Translation, Biomedical Translation, Multilingual Low-Resource Translation for Indo-European Languages, Large-Scale Multilingual Machine Translation, Triangular MT: Using English to Improve Russian-to-Chinese Machine Translation, Translation Efficiency, Machine Translation using Terminologies, Unsupervised and Very Low Resource Supervised Translation, and Lifelong Learning for Machine Translation, two evaluation tasks: Quality Estimation of Translation and Metrics for Machine Translation, and the Automatic Post-Editing task.

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 2021 has received 49 full research paper submissions (not counting withdrawn submissions). In total, WMT 2021 featured 18 full research paper presentations and 96 shared task presentations.

The event hosted a panel discussion led by Markus Freitag (Google) on evaluation with Nitika Mathur (Univ. Melbourne), Benjamin Marie (NICT), Ricardo Rei (Unbabel), Tom Kocmi (Microsoft).

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.

Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Marco Turchi, and Marcos Zampieri

Co-Organizers

Organizing Committee

Organizers:

Loïc Barrault (University of Sheffield)
Ondřej Bojar (Charles University in Prague)
Fethi Bougares (University of Le Mans)
Rajen Chatterjee (Apple)
Marta R. Costa-jussà (Universitat Politècnica de Catalunya)
Christian Federmann (MSR)
Mark Fishel (University of Tartu)
Alexander Fraser (LMU Munich)
Markus Freitag (Google)
Yvette Graham (DCU)
Roman Grundkiewicz (MSR)
Paco Guzman (Facebook)
Barry Haddow (University of Edinburgh)
Matthias Huck (LMU Munich)
Antonio Jimeno Yepes (IBM Research Australia)
Philipp Koehn (University of Edinburgh / Johns Hopkins University)
Tom Kocmi (MSR)
André Martins (Unbabel)
Makoto Morishita (NTT)
Christof Monz (University of Amsterdam)
Masaaki Nagata (NTT)
Toshiaki Nakazawa (University of Tokyo)
Matteo Negri (FBK)
Aurélie Névéol (LIMSI, CNRS)
Mariana Neves (German Federal Institute for Risk Assessment)
Martin Popel (Charles University in Prague)
Matt Post (Johns Hopkins University)
Marco Turchi (FBK)
Marcos Zampieri (Rochester Institute of Technology)

Panelists:

Nitika Mathur (Univ. Melbourne)
Benjamin Marie (NICT)
Ricardo Rei (Unbabel)
Tom Kocmi (Microsoft)

Program Committee:

Tamer Alkhouli (AppTek)
Mihael Arcan (National Universith of Ireland Galway)
Duygu Ataman (New York University)

Eleftherios Avramidis (German Research Center for Artificial Intelligence (DFKI))
Amittai Axelrod (DiDi Labs)
Parnia Bahar (AppTek)
Petra Barancikova (Charles University in Prague, Faculty of Mathematics and Physics)
Rachel Bawden (Inria)
Meriem Beloucif (University of Hamburg)
Bill Byrne (University of Cambridge)
Ozan Caglayan (Imperial College London)
Sheila Castilho (Dublin City University)
Daniel Cer (Google Research; University of California at Berkeley)
Vishrav Chaudhary (Facebook AI)
Boxing Chen (Alibaba)
Pinzhen Chen (The University of Edinburgh)
Colin Cherry (Google)
Vishal Chowdhary (MSR)
Raj Dabre (NICT)
Steve DeNeefe (SDL Research)
Michael Denkowski (Amazon)
Mattia A. Di Gangi (AppTek GmbH)
Shuoyang Ding (Johns Hopkins University)
Miguel Domingo (Universitat Politècnica de València)
Kevin Duh (Johns Hopkins University)
Hiroshi Echizen-ya (Hokkai-Gakuen University)
Sergey Edunov (Faceook AI Research)
Miquel Esplà-Gomis (Universitat d'Alacant)
Angela Fan (Facebook AI Research)
Marcello Federico (Amazon AI)
Orhan Firat (Google AI)
George Foster (Google)
Atsushi Fujita (National Institute of Information and Communications Technology)
Ulrich Germann (University of Edinburgh)
Jesús González-Rubio (WebInterpret)
Isao Goto (NHK)
Cyril Goutte (National Research Council Canada)
Naman Goyal (Facebook)
Jeremy Gwinnup (Air Force Research Laboratory)
Nizar Habash (New York University Abu Dhabi)
Viktor Hangya (Ludwig-Maximilians-Universität München)
Greg Hanneman (Amazon)
Yifan He (Alibaba Group)
John Henderson (MITRE)
Christian Herold (RWTH Aachen University)
Cong Duy Vu Hoang (Oracle)
Mika Hämäläinen (University of Helsinki, Rootroo Ltd)
Kenji Imamura (National Institute of Information and Communications Technology)
Phillip Keung (Amazon)
Yunsu Kim (Lilt, Inc.)

Rebecca Knowles (National Research Council Canada)
Tom Kocmi (Microsoft)
Julia Kreutzer (Google)
Roland Kuhn (National Research Council of Canada)
Shankar Kumar (Google)
Anoop Kunchukuttan (Microsoft AI and Research)
Surafel Melaku Lakew (Amazon AI)
Ekaterina Lapshinova-Koltunski (Universität des Saarlandes)
Alon Lavie (Unbabel/Carnegie Mellon University)
Qun Liu (Huawei Noah's Ark Lab)
Samuel Läubli (University of Zurich)
Andreas Maletti (Universität Leipzig)
Sameen Maruf (Monash University)
Rebecca Marvin (Independent)
Antonio Valerio Miceli Barone (The University of Edinburgh)
Tomáš Musil (Charles University)
Mathias Müller (University of Zurich)
Preslav Nakov (Qatar Computing Research Institute, HBKU)
Jan Niehues (Maastricht University)
Xing Niu (Amazon AI)
Tsuyoshi Okita (Kyushu institute of technology)
Daniel Ortiz-Martínez (University of Barcelona)
Santanu Pal (Saarland University)
Carla Parra Escartín (Iconic Translation Machines)
Pavel Pecina (Charles University)
Stephan Peitz (Apple)
Sergio Penkale (Lingo24)
Marcis Pinnis (Tilde)
Maja Popović (ADAPT, Dublin City University)
Matt Post (Johns Hopkins University)
MatÄ«ss Rikters (University of Tartu)
Annette Rios (University of Zurich)
Elizabeth Salesky (Johns Hopkins University)
Hassan Sawaf (aixplain, inc.)
Rico Sennrich (University of Zurich)
Aditya Siddhant (Google)
Patrick Simianer (Lilt)
Felix Stahlberg (Google Research)
David Stap (University of Amsterdam)
Sara Stymne (Uppsala University)
Katsuhito Sudoh (Nara Institute of Science and Technology (NAIST))
Víctor M. Sánchez-Cartagena (Universitat d'Alacant)
Aleš Tamchyna (Memsource)
Gongbo Tang (Uppsala University)
Tristan Thrush (Facebook AI Research (FAIR))
Jörg Tiedemann (University of Helsinki)
Antonio Toral (University of Groningen)

Ke Tran (Amazon)
Masao Utiyama (NICT)
David Vilar (Google)
Ekaterina Vylomova (University of Melbourne)
Weiyue Wang (RWTH Aachen University)
Taro Watanabe (Nara Institute of Science and Technology)
Guillaume Wenzek (Facebook AI Research)
Joern Wuebker (Lilt, Inc.)
Hainan Xu (Google)
François Yvon (LISN CNRS & Univ. Paris Saclay)
Xuan Zhang (Johns Hopkins University)
Zhong Zhou (Carnegie Mellon University)

Table of Contents

Conference Program

Wednesday, November 10, 2021

9:00–9:10 ***Opening Remarks***

9:10–10:15 Session 1: Shared Task Overview Papers I (Session Chair: Philipp Koehn)

9:10–10:15 *Findings of the 2021 Conference on Machine Translation (WMT21)*
Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin and Marcos Zampieri

9:00–9:25 ***News Translation Task***

9:25–9:35 ***Similar Languages Translation Task***

9:35–9:45 ***Automatic Postediting Task***

9:45–9:55 ***Triangular Translation Task***

9:55–10:05 ***Indo-European Multilingual Translation Task***

10:05–10:15 *Findings of the WMT 2021 Shared Task on Large-Scale Multilingual Machine Translation*
Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush and Francisco Guzmán

10:15–10:30 ***Coffee Break***

Wednesday, November 10, 2021 (continued)

10:30–12:00 News Translation Task

10:30–12:00 *GTCOM Neural Machine Translation Systems for WMT21*
Chao Bei and Hao Zong

10:30–12:00 *The University of Edinburgh's English-German and English-Hausa Submissions to the WMT21 News Translation Task*
Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch and Kenneth Heafield

10:30–12:00 *Tune in: The AFRL WMT21 News-Translation Systems*
Grant Erdmann, Jeremy Gwinnup and Tim Anderson

10:30–12:00 *The TALP-UPC Participation in WMT21 News Translation Task: an mBART-based NMT Approach*
Carlos Escolano, Ioannis Tsiamas, Christine Basta, Javier Ferrando, Marta R. Costa-jussa and José A. R. Fonollosa

10:30–12:00 *CUNI Systems in WMT21: Revisiting Backtranslation Techniques for English-Czech NMT*
Petr Gebauer, Ondřej Bojar, Vojtěch Švandelík and Martin Popel

10:30–12:00 *Ensembling of Distilled Models from Multi-task Teachers for Constrained Resource Language Pairs*
Amr Hendy, Esraa A. Gad, Mohamed Abdelghaffar, Jailan S. ElMosalami, Mohamed Afify, Ahmed Y. Tawfik and Hany Hassan Awadalla

10:30–12:00 *Miðeind's WMT 2021 Submission*
Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Pétur Orri Ragnarson, Haukur Jónsson and Vilhjalmur Thorsteinsson

10:30–12:00 *Allegro.eu Submission to WMT21 News Translation Task*
Mikołaj Koszowski, Karol Grzegorczyk and Tsimur Hadeliya

10:30–12:00 *Illinois Japanese ↔ English News Translation for WMT 2021*
Giang Le, Shinka Mori and Lane Schwartz

10:30–12:00 *MiSS@WMT21: Contrastive Learning-reinforced Domain Adaptation in Neural Machine Translation*
Zuchao Li, Masao Utiyama, Eiichiro Sumita and Hai Zhao

10:30–12:00 *The Fujitsu DMATH Submissions for WMT21 News Translation and Biomedical Translation Tasks*
Ander Martinez

Wednesday, November 10, 2021 (continued)

10:30–12:00 *Adam Mickiewicz University's English-Hausa Submissions to the WMT 2021 News Translation Task*
Artur Nowakowski and Tomasz Dwojak

10:30–12:00 *eTranslation's Submissions to the WMT 2021 News Translation Task*
Csaba Oravecz, Katina Bontcheva, David Kolovratník, Bhavani Bhaskar, Michael Jellinghaus and Andreas Eisele

10:30–12:00 *The University of Edinburgh's Bengali-Hindi Submissions to the WMT21 News Translation Task*
Proyag Pal, Alham Fikri Aji, Pinzhen Chen and Sukanta Sen

10:30–12:00 *The Volctrans GLAT System: Non-autoregressive Translation Meets WMT21*
Lihua Qian, Yi Zhou, Zaixiang Zheng, Yaoming ZHU, Zehui Lin, Jiangtao Feng, Shanbo Cheng, Lei Li, Mingxuan Wang and Hao Zhou

10:30–12:00 *NVIDIA NeMo's Neural Machine Translation Systems for English-German and English-Russian News and Biomedical Tasks at WMT21*
Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams and Oleksii Kuchaiev

10:30–12:00 *Facebook AI's WMT21 News Translation Task Submission*
Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov and Angela Fan

10:30–12:00 *Tencent Translation System for the WMT21 News Translation Task*
Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng and Wen Zhang

10:30–12:00 *HW-TSC's Participation in the WMT 2021 News Translation Shared Task*
Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang and Ying Qin

10:30–12:00 *LISN @ WMT 2021*
Jitao Xu, Minh Quang Pham, Sadaf Abdul Rauf and François Yvon

10:30–12:00 *WeChat Neural Machine Translation Systems for WMT21*
Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu and Jie Zhou

10:30–12:00 *Small Model and In-Domain Data Are All You Need*
Hui Zeng

10:30–12:00 *The Mininglamp Machine Translation System for WMT21*
Shiyu Zhao, Xiaopu Li, Minghui Wu and Jie Hao

Wednesday, November 10, 2021 (continued)

10:30–12:00 *The NiuTrans Machine Translation Systems for WMT21*
Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao and Jingbo Zhu

10:30–12:00 Similar Languages Translation Task

10:30–12:00 *Improving Similar Language Translation With Transfer Learning*
Ife Adebara and Muhammad Abdul-Mageed

10:30–12:00 *T4T Solution: WMT21 Similar Language Task for the Spanish-Catalan and Spanish-Portuguese Language Pair*
Miguel Canals and Marc Raventós Tato

10:30–12:00 *Neural Machine Translation for Tamil–Telugu Pair*
Sahinur Rahman Laskar, Bishwaraj Paul, Prottay Kumar Adhikary, Partha Pakray and Sivaji Bandyopadhyay

10:30–12:00 *Low Resource Similar Language Neural Machine Translation for Tamil-Telugu*
Vandan Mujadia and Dipti Sharma

10:30–12:00 *Similar Language Translation for Catalan, Portuguese and Spanish Using Marian NMT*
Reinhard Rapp

10:30–12:00 *NITK-UoH: Tamil-Telugu Machine Translation Systems for the WMT21 Similar Language Translation Task*
Richard Saldanha, Ananthanarayana V. S, Anand Kumar M and Parameswari Krishnamurthy

10:30–12:00 *A3-108 Machine Translation System for Similar Language Translation Shared Task 2021*
Saumitra Yadav and Manish Shrivastava

10:30–12:00 Automatic Post-Editing Task

10:30–12:00 *Netmarble AI Center's WMT21 Automatic Post-Editing Shared Task Submission*
Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An and Insoo Oh

10:30–12:00 *Adapting Neural Machine Translation for Automatic Post-Editing*
Abhishek Sharma, Prabhakar Gupta and Anil Nelakanti

10:30–12:00 Triangular Translation Task

10:30–12:00 *ISTIC's Triangular Machine Translation System for WMT2021*
Hangcheng Guo, Wenbin Liu, Yanqing He, Tian Lan, Hongjiao Xu, Zhenfeng Wu and You Pan

10:30–12:00 *HW-TSC's Participation in the WMT 2021 Triangular MT Shared Task*
Zongyao Li, Daimeng Wei, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Zhengzhe Yu, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang and Ying Qin

10:30–12:00 *DUTNLP Machine Translation System for WMT21 Triangular Translation Task*
Huan Liu, Junpeng Liu, Kaiyu Huang and Degen Huang

10:30–12:00 *Pivot Based Transfer Learning for Neural Machine Translation: CFILT IITB @ WMT 2021 Triangular MT*
Shivam Mhaskar and Pushpak Bhattacharyya

10:30–12:00 *Papago's Submissions to the WMT21 Triangular Translation Task*
Jeonghyeok Park, Hyunjoong Kim and Hyunchang Cho

Wednesday, November 10, 2021 (continued)

10:30–12:00 Indo-European Multilingual Translation Task

10:30–12:00 *Machine Translation of Low-Resource Indo-European Languages*
Wei-Rui Chen and Muhammad Abdul-Mageed

10:30–12:00 *CUNI systems for WMT21: Multilingual Low-Resource Translation for Indo-European Languages Shared Task*
Josef Jon, Michal Novák, João Paulo Aires, Dusan Varis and Ondřej Bojar

10:30–12:00 *Transfer Learning with Shallow Decoders: BSC at WMT2021's Multilingual Low-Resource Translation for Indo-European Languages Shared Task*
Ksenia Kharitonova, Ona de Gibert Bonet, Jordi Armengol-Estapé, Mar Rodriguez i Alvarez and Maite Melero

10:30–12:00 *EdinSaar@WMT21: North-Germanic Low-Resource Multilingual NMT*
Svetlana Tchistiakova, Jesujoba Alabi, Koel Dutta Chowdhury, Sourav Dutta and Dana Ruiter

10:30–12:00 *TenTrans Multilingual Low-Resource Translation System for WMT21 Indo-European Languages Task*
Han Yang, Bojie Hu, Wanying Xie, ambyera han, Pan Liu, Jinan Xu and Qi Ju

10:30–12:00 Large-Scale Multilingual Translation Task

10:30–12:00 *The University of Maryland, College Park Submission to Large-Scale Multilingual Shared Task at WMT 2021*
Saptarashmi Bandyopadhyay, Tasnim Kabir, Zizhen Lian and Marine Carpuat

10:30–12:00 *To Optimize, or Not to Optimize, That Is the Question: TelU-KU Models for WMT21 Large-Scale Multilingual Machine Translation*
Sari Dewi Budiwati, Tirana Fatyanosa, Mahendra Data, Dedy Rahman Wijaya, Patrick Adolf Telnoni, Arie Ardiyanti Suryani, Agus Pratondo and Masayoshi Aritsugi

10:30–12:00 *MMTAfrica: Multilingual Machine Translation for African Languages*
Chris Chinenye Emezue and Bonaventure F. P. Dossou

10:30–12:00 *The LMU Munich System for the WMT 2021 Large-Scale Multilingual Machine Translation Shared Task*
Wen Lai, Jindřich Libovický and Alexander Fraser

10:30–12:00 *Back-translation for Large-Scale Multilingual Machine Translation*
Baohao Liao, Shahram Khadivi and Sanjika Hewavitharana

Wednesday, November 10, 2021 (continued)

10:30–12:00 *Maastricht University's Large-Scale Multilingual Machine Translation System for WMT 2021*
Danni Liu and Jan Niehues

10:30–12:00 *Data Processing Matters: SRPH-Konvergen AI's Machine Translation System for WMT'21*
Lintang Sutawika and Jan Christian Blaise Cruz

10:30–12:00 *TenTrans Large-Scale Multilingual Machine Translation System for WMT21*
Wanying Xie, Bojie Hu, Han Yang, Dong Yu and Qi Ju

10:30–12:00 *Multilingual Machine Translation Systems from Microsoft for WMT21 Shared Task*
Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song and Furu Wei

10:30–12:00 *HW-TSC's Participation in the WMT 2021 Large-Scale Multilingual Translation Task*
Zhengzhe Yu, Daimeng Wei, Zongyao Li, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang and Ying Qin

12:00–13:00 ***Lunch Break***

13:00–14:15 Session 3: Panel Discussion on Evaluation with Nitika Mathur (Univ. Melbourne), Benjamin Marie (NICT), Ricardo Rei (Unbabel), Tom Kocmi (Microsoft) and moderated by Markus Freitag (Google)

14:15–14:45 ***Mini Break***

14:45–16:15 Session 4: Research Papers on Evaluation (Session Chair: Antonis Anastasopoulos)

14:45–16:15 *On the Stability of System Rankings at WMT*
Rebecca Knowles

14:45–16:15 *To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation*
Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita and Arul Menezes

14:45–16:15 *Just Ask! Evaluating Machine Translation by Asking and Answering Questions*
Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens and Pavel Pecina

Wednesday, November 10, 2021 (continued)

14:45–16:15 *A Fine-Grained Analysis of BERTScore*
Michael Hanna and Ondřej Bojar

14:45–16:15 *Evaluating Multiway Multilingual NMT in the Turkic Languages*
Jamshidbek Mirzakhalov, Anoop Babu, Aigiz Kunafin, Ahsan Wahab, Bekhzodbek Moydinboyev, Sardana Ivanova, Mokhiyakhon Uzokova, Shaxnoza Pulatova, Duygu Ataman, Julia Kreutzer, Francis Tyers, Orhan Firat, John Licato and Sriram Chellappan

14:45–16:15 *Extending Challenge Sets to Uncover Gender Bias in Machine Translation: Impact of Stereotypical Verbs and Adjectives*
Jonas-Dario Troles and Ute Schmid

16:15-16:45 ***Coffee Break***

16:45–18:15 Session 5: Research Papers on Data (Session Chair: Mathias Müller)

16:45–18:15 *Continual Learning in Multilingual NMT via Language-Specific Embeddings*
Alexandre Berard

16:45–18:15 *DELA Corpus - A Document-Level Corpus Annotated with Context-Related Issues*
Sheila Castilho, João Lucas Cavalheiro Camargo, Miguel Menezes and Andy Way

16:45–18:15 *Multilingual Domain Adaptation for NMT: Decoupling Language and Domain Information with Adapters*
Asa Cooper Stickland, Alexandre Berard and Vassilina Nikoulina

16:45–18:15 *Translation Transformers Rediscover Inherent Data Domains*
Maksym Del, Elizaveta Korotkova and Mark Fishel

16:45–18:15 *Improving Machine Translation of Rare and Unseen Word Senses*
Viktor Hangya, Qianchu Liu, Dario Stojanovski, Alexander Fraser and Anna Korhonen

16:45–18:15 *Pushing the Right Buttons: Adversarial Evaluation of Quality Estimation*
Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan and Lucia Specia

Thursday, November 11, 2021

9:00–10:15 Session 6: Shared Task Overview Papers I

9:00–9:12 *Findings of the WMT 2021 Shared Task on Efficient Translation*
Kenneth Heafield, Qianqian Zhu and Roman Grundkiewicz

9:12–9:24 *Findings of the WMT Shared Task on Machine Translation Using Terminologies*
Md Mahfuz Ibn Alam, Ivana Kvapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu, Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn and Vassilina Nikoulina

9:24–9:36 *Findings of the WMT 2021 Biomedical Translation Shared Task: Summaries of Animal Experiments as New Test Set*
Lana Yeganova, Dina Wiemann, Mariana Neves, Federica Vezzani, Amy Siu, Inigo Jauregi Unanue, Maite Oronoz, Nancy Mah, Aurélie Névéol, David Martinez, Rachel Bawden, Giorgio Maria Di Nunzio, Roland Roller, Philippe Thomas, Cristian Grozea, Olatz Perez-de-Viñaspre, Maika Vicente Navarro and Antonio Jimeno Yepes

9:36–9:48 *Findings of the WMT 2021 Shared Task on Quality Estimation*
Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary and André F. T. Martins

9:48–10:00 *Findings of the WMT 2021 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT*
Jindřich Libovický and Alexander Fraser

10:00–10:15 *Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain*
Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie and Ondřej Bojar

10:15–10:30 ***Coffee Break***

10:30–12:00 Efficient Translation Task

10:30–12:00 *Efficient Machine Translation with Model Pruning and Quantization*
Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap and Roman Grundkiewicz

10:30–12:00 *HW-TSC's Participation in the WMT 2021 Efficiency Shared Task*
Hengchao Shang, Ting Hu, Daimeng Wei, Zongyao Li, Jianfei Feng, ZhengZhe Yu, Jiaxin Guo, Shaojun Li, Lizhi Lei, ShiMin Tao, Hao Yang, Jun Yao and Ying Qin

10:30–12:00 *The NiuTrans System for the WMT 2021 Efficiency Task*
Chenglong Wang, Chi Hu, Yongyu Mu, Zhongxiang Yan, Siming Wu, Yimin Hu, Hang Cao, Bei Li, Ye Lin, Tong Xiao and Jingbo Zhu

10:30–12:00 *TenTrans High-Performance Inference Toolkit for WMT2021 Efficiency Task*
Kaixin WU, Bojie Hu and Qi Ju

10:30–12:00 Terminology Translation Task

10:30–12:00 *Lingua Custodia's Participation at the WMT 2021 Machine Translation Using Terminologies Shared Task*
Melissa Ailem, Jingshu Liu and Raheel Qader

10:30–12:00 *Kakao Enterprise's WMT21 Machine Translation Using Terminologies Task Submission*
Yunju Bak, Jimin Sun, Jay Kim, Sungwon Lyu and Changmin Lee

10:30–12:00 *The SPECTRANS System Description for the WMT21 Terminology Task*
Nicolas Ballier, Dahn Cho, Bilal Faye, Zong-You Ke, Hanna Martikainen, Mojca Pecman, Guillaume Wisniewski, Jean-Baptiste Yunès, Lichao Zhu and Maria Zimina-Poirot

10:30–12:00 *Dynamic Terminology Integration for COVID-19 and Other Emerging Domains*
Toms Bergmanis and Mārcis Pinnis

10:30–12:00 *CUNI Systems for WMT21: Terminology Translation Shared Task*
Josef Jon, Michal Novák, João Paulo Aires, Dusan Varis and Ondřej Bojar

10:30–12:00 *PROMT Systems for WMT21 Terminology Translation Task*
Alexander Molchanov, Vladislav Kovalenko and Fedor Bykov

Thursday, November 11, 2021 (continued)

10:30–12:00 *SYSTRAN @ WMT 2021: Terminology Task*
Minh Quang Pham, Josep Crego, Antoine Senellart, Dan Berrebbi and Jean Senellart

10:30–12:00 *TermMind: Alibaba's WMT21 Machine Translation Using Terminologies Task Submission*
Ke Wang, Shuqin Gu, Boxing Chen, Yu Zhao, Weihua Luo and Yuqi Zhang

10:30–12:00 Biomedical Translation Task

10:30–12:00 *FJWU Participation for the WMT21 Biomedical Translation Task*
Sumbal Naz, Sadaf Abdul Rauf and Sami Ul Haq

10:30–12:00 *High Frequent In-domain Words Segmentation and Forward Translation for the WMT21 Biomedical Task*
Bardia Rafieian and Marta Ruiz Costa Jussa

10:30–12:00 *Huawei AARC's Submissions to the WMT21 Biomedical Translation Task: Domain Adaption from a Practical Perspective*
Weixuan Wang, Wei Peng, Xupeng Meng and Qun Liu

10:30–12:00 *Tencent AI Lab Machine Translation Systems for the WMT21 Biomedical Translation Task*
Xing Wang, Zhaopeng Tu and Shuming Shi

10:30–12:00 *HW-TSC's Submissions to the WMT21 Biomedical Translation Task*
Hao Yang, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Daimeng Wei, Zongyao Li, Hengchao Shang, Minghan Wang, Jiaxin Guo, Lizhi Lei, chuanfei xu, Min Zhang and Ying Qin

10:30–12:00 Quality Estimation Task

10:30–12:00 *RTM Super Learner Results at Quality Estimation Task*
Ergun Biçici

10:30–12:00 *HW-TSC's Participation at WMT 2021 Quality Estimation Shared Task*
Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, Yujia Liu and Shujian Huang

10:30–12:00 *Ensemble Fine-tuned mBERT for Translation Quality Estimation*
Shaika Chowdhury, Naouel Baili and Brian Vannah

10:30–12:00 *The JHU-Microsoft Submission for WMT21 Quality Estimation Shared Task*
Shuoyang Ding, Marcin Junczys-Dowmunt, Matt Post, Christian Federmann and Philipp Koehn

10:30–12:00 *TUDa at WMT21: Sentence-Level Direct Assessment with Adapters*
Gregor Geigle, Jonas Stadtmüller, Wei Zhao, Jonas Pfeiffer and Steffen Eger

10:30–12:00 *Quality Estimation Using Dual Encoders with Transfer Learning*
Dam Heo, WonKee Lee, Baikjin Jung and Jong-Hyeok Lee

10:30–12:00 *ICL's Submission to the WMT21 Critical Error Detection Shared Task*
Genze Jiang, Zhenhao Li and Lucia Specia

10:30–12:00 *Papago's Submission for the WMT21 Quality Estimation Shared Task*
Seunghyun Lim, Hantae Kim and Hyunjoong Kim

10:30–12:00 *NICT Kyoto Submission for the WMT'21 Quality Estimation Task: Multimetric Multilingual Pretraining for Critical Error Detection*
Raphael Rubino, Atsushi Fujita and Benjamin Marie

10:30–12:00 *QEMind: Alibaba's Submission to the WMT21 Quality Estimation Shared Task*
Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo and Yuqi Zhang

10:30–12:00 *Direct Exploitation of Attention Weights for Translation Quality Estimation*
Lisa Yankovskaya and Mark Fishel

10:30–12:00 *IST-Unbabel 2021 Submission for the Quality Estimation Shared Task*
Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, miguel vera, Fabio Kepler and André F. T. Martins

10:30–12:00 Unsupervised and Very Low Resource Translation Task

10:30–12:00 *The IICT-Yverdon System for the WMT 2021 Unsupervised MT and Very Low Resource Supervised MT Task*
Àlex R. Atrio, Gabriel Luthier, Axel Fahy, Giorgos Vernikos, Andrei Popescu-Belis and Ljiljana Dolamic

10:30–12:00 *Unsupervised Translation of German–Lower Sorbian: Exploring Training and Novel Transfer Methods on a Low-Resource Language*
Lukas Edman, Ahmet Üstün, Antonio Toral and Gertjan van Noord

10:30–12:00 *The LMU Munich Systems for the WMT21 Unsupervised and Very Low-Resource Translation Task*
Jindřich Libovický and Alexander Fraser

10:30–12:00 *Language Model Pretraining and Transfer Learning for Very Low Resource Languages*
Jyotsana Khatri, Rudra Murthy and Pushpak Bhattacharyya

10:30–12:00 *NRC-CNRC Systems for Upper Sorbian-German and Lower Sorbian-German Machine Translation 2021*
Rebecca Knowles and Samuel Larkin

10:30–12:00 *NoahNMT at WMT 2021: Dual Transfer for Very Low Resource Supervised Machine Translation*
Meng Zhang, Minghao Wu, Pengfei Li, Liangyou Li and Qun Liu

Thursday, November 11, 2021 (continued)

10:30–12:00 Metrics Task

10:30–12:00 *cushLEPOR: customising hLEPOR metric using Optuna for higher agreement with human judgments or pre-trained language model LaBSE*
Lifeng Han, Irina Sorokina, Gleb Erofeev and Serge Gladkoff

10:30–12:00 *MTEQA at WMT21 Metrics Shared Task*
Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens and Pavel Pecina

10:30–12:00 *Are References Really Needed? Unbabel-IST 2021 Submission for the Metrics Shared Task*
Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins and Alon Lavie

10:30–12:00 *Regressive Ensemble for Machine Translation Quality Evaluation*
Michal Stefanik, Vít Novotný and Petr Sojka

10:30–12:00 *Multilingual Machine Translation Evaluation Metrics Fine-tuned on Pseudo-Negative Examples for WMT 2021 Metrics Task*
Kosuke Takahashi, Yoichi Ishibashi, Katsuhito Sudoh and Satoshi Nakamura

10:30–12:00 *RoBLEURT Submission for WMT2021 Metrics Task*
Yu Wan, Dayiheng Liu, Baosong Yang, Tianchi Bi, Haibo Zhang, Boxing Chen, Weihua Luo, Derek F. Wong and Lidia S. Chao

10:30–12:00 Test Suites

10:30–12:00 *Linguistic Evaluation for the 2021 State-of-the-art Machine Translation Systems for German to English and English to German*
Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova and Sebastian Möller

12:00–13:00 ***Lunch Break***

13:00–14:15 Session 8: Research Papers on Training and Modelling (Session Chair: Mark Fishel)

13:00–14:15 *Pruning Neural Machine Translation for Speed Using Group Lasso*
Maximiliana Behnke and Kenneth Heafield

13:00–14:15 *Phrase-level Active Learning for Neural Machine Translation*
Junjie Hu and Graham Neubig

13:00–14:15 *Learning Feature Weights using Reward Modeling for Denoising Parallel Corpora*
Gaurav Kumar, Philipp Koehn and Sanjeev Khudanpur

13:00–14:15 *Monotonic Simultaneous Translation with Chunk-wise Reordering and Refinement*
HyoJung Han, Seokchan Ahn, Yoonjung Choi, Insoo Chung, Sangha Kim and Kyunghyun Cho

13:00–14:15 *Simultaneous Neural Machine Translation with Constituent Label Prediction*
Yasumasa Kano, Katsuhito Sudoh and Satoshi Nakamura

13:00–14:15 *Contrastive Learning for Context-aware Neural Machine Translation Using Coreference Information*
Yongkeun Hwang, Hyeongu Yun and Kyomin Jung

14:15–14:45 ***Mini Break***

Thursday, November 11, 2021 (continued)

14:45–15:45 Session 9: Machine Translation Papers from the Findings of the EMNLP (Session Chair: Matt Post)

14:45–15:00 ***The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation, Orevaoghene Ahia, Julia Kreutzer and Sara Hooker***

15:00–15:15 ***Subword Mapping and Anchoring across Languages, Giorgos Vernikos and Andrei Popescu-Belis***

15:15–15:30 ***Uncertainty-Aware Machine Translation Evaluation, Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei and André F. T. Martins***

15:30–15:45 ***Sometimes We Want Ungrammatical Translations, Prasanna Parasarathi***

Findings of the 2021 Conference on Machine Translation (WMT21)

Farhad Akhbardeh RIT
Arkady Arkhangorodsky DiDi Labs
Magdalena Biesialska UPC
Ondřej Bojar Charles University
Rajen Chatterjee Apple Inc.
Vishrav Chaudhary Facebook AI
Marta R. Costa-jussà UPC
Cristina España-Bonet DFKI
Angela Fan Facebook AI
Christian Federmann Microsoft Cloud + AI
Markus Freitag Google Research
Yvette Graham Trinity College Dublin
Roman Grundkiewicz Microsoft
Barry Haddow University of Edinburgh
Leonie Harter DFKI
Kenneth Heafield University of Edinburgh
Christopher M. Homan RIT
Matthias Huck SAP SE
Kwabena Amponsah-Kaakyire DFKI
Jungo Kasai U. of Washington
Daniel Khashabi Allen Institute for AI
Kevin Knight DiDi Labs
Tom Kocmi Microsoft
Philipp Koehn JHU
Nicholas Lourie New York University
Christof Monz University of Amsterdam
Makoto Morishita NTT
Masaaki Nagata NTT
Ajay Nagesh DiDi Labs
Toshiaki Nakazawa University of Tokyo
Matteo Negri FBK
Santanu Pal WIPRO AI
Allahsera Tapo RIT
Marco Turchi FBK
Valentin Vydrin INALCO
Marcos Zampieri RIT

Abstract

This paper presents the results of the news translation task, the multilingual low-resource translation for Indo-European languages, the triangular translation task, and the automatic post-editing task organised as part of the Conference on Machine Translation (WMT) 2021. In the news task, participants were asked to build machine translation systems for any of 10 language pairs, to be evaluated on test sets consisting mainly of news stories. The task was also opened up to additional test suites to probe specific aspects of translation. In the Similar Language Translation (SLT) task, participants were asked to develop systems to translate between pairs of similar languages from the Dravidian and Romance family as well as French to two similar low-resource Manding languages (Bambara and Maninka). In the Triangular MT translation task, participants were asked to build a Russian to Chinese translator, given parallel data in Russian-Chinese, Russian-English and English-Chinese. In the multilingual low-resource translation for Indo-European languages task, participants built multilingual systems to translate among Romance and North-Germanic languages. The task was designed to deal with the translation of documents in the cultural heritage domain for relatively low-resourced languages. In the automatic post-editing (APE) task, participants were asked to develop systems capable to correct the errors made by an unknown machine translation systems.

1 Introduction

The Sixth Conference on Machine Translation (WMT21)[1] was held online with EMNLP 2021 and hosted a number of shared tasks on various aspects of machine translation. This conference built on 15 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, 2016, 2017, 2018a; Barrault et al., 2019, 2020).

This year we conducted several official tasks. In this paper we report on the news task, the multilingual low-resource translation for Indo-European languages task, the triangular translation task, and the automatic post-editing task. Additional shared tasks are described in separate papers in these proceedings:

[1] http://www.statmt.org/wmt21/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1–88
November 10–11, 2021.

- biomedical translation (Yeganova et al., 2021)
- efficiency (Heafield et al., 2021)
- large-scale multilingual machine translation (Wenzek et al., 2021)
- machine translation using terminologies (Alam et al., 2021)
- metrics (Freitag et al., 2021b)
- quality estimation (Specia et al., 2021)
- unsupervised and very low-resource translation (Libovický and Fraser, 2021)

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 included 20 translation directions this year, with translation between English and each of Chinese, Czech, German, Japanese and Russian, as well as French↔German being repeated from last year, and English to and from Hausa and Icelandic being new for this year, along with Bengali↔Hindi and Xhosa↔Zulu. The translation tasks covered a range of language families, and included both low-resource and high-resource pairs. System outputs for each task were evaluated both automatically and manually, but we only include the manual evaluation here.

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. We collected additional assessments from a pool of linguists, as well as crowd-workers. This year, the official manual evaluation metric is again based on judgments of adequacy on a 100-point scale, a method (known as "direct assessment", DA) that we explored in the previous years with convincing results in terms of the trade-off between annotation effort and reliable distinctions between systems. In addition, other golden standards with this year's systems were collected. The human-in-the-loop GENIE leaderboard (Khashabi et al., 2021) conducted de→en evaluations independently in a Likert scale (Section 3.5). We refer the reader to Freitag et al. (2021b) for MQM scoring of en→de, en→ru, and zh→en.

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 translations are also available for interactive visualization and comparison of differences between systems at http://wmt.ufal.cz/ using MT-ComparEval (Sudarikov et al., 2016), and also on ExplainaBoard[3] (Liu et al., 2021b).

In order to gain further insight into the performance of individual MT systems, we again organized a call for dedicated "test suites". Test suites are custom additions to the inputs. Anyone can provide a test suite for any subset of news translation task languages and we ensure that the test suite is requested from all participating MT systems. The MT outputs are delivered back to test suite authors for evaluation, which can be manual, automatic or both, focusing on any possible aspect of the MT systems. This year, five test suites were acquired and translated by participating MT systems but only two were then analyzed in time for these proceedings:

- Freitag et al. (2021b), the metrics task paper, used TED talks as additional domain, scored them with MQM, and further used these outputs and scores to assess domain-dependence of MT evaluation metrics.

- Macketanz et al. (2021) reports on the fourth application of a fine-grained test suite for German↔English linguistic phenomena. The previous instances (Macketanz et al., 2018; Avramidis et al., 2019, 2020) use the same underlying collection of sentences and thus allow to observe the overall development of MT systems in clear categories. This year, the major jump was observed in the category of idioms, especially due to a few exceptional MT systems. Many phenomena are being solved almost perfectly, the difficult categories remain false friends, ambiguity and multi-word expressions.

The goal of the Similar Language Translation (SLT) task (Section 4) is to evaluate the perfor-

[2] http://statmt.org/wmt21/results.html
[3] http://explainaboard.nlpedia.ai/leaderboard/task-mt/index.php

mance of MT systems taking into account the similarity between pairs of closely-related languages from the same language family. Following the interest of the community in this topic (Costa-jussà et al., 2018; Popović et al., 2020) and the success of the past two editions of the SLT task task at WMT 2019 and WMT 2020, we organize a third iteration of the task at WMT 2021. SLT 2021 features a pair of similar Dravidian languages, namely Tamil - Telugu, and multiple pairs of Romance languages involving Catalan, Spanish, Portuguese, and Romanian in all possible combinations. A new track with French and two similar low-resource Manding languages: Bambara and Maninka was also included to encourage participants to take advantage of the similarity between Bambara and Maninka and explore data augmentation techniques, a typical scenario of low-resource languages. Finally, translations were evaluated in both directions using three automatic metrics: BLEU, RIBES, and TER.

The primary goals of the Triangular MT task (Section 5) are to promote translation between non-English languages, to optimally mix direct and indirect parallel resources and exploit noisy web data sources to build an MT system. Specifically, the task was Russian to Chinese machine translation, given parallel data comprising of direct (Russian-Chinese) and indirect (Russian-English and English-Chinese) sources. The submitted systems were evaluated on a (secret) mixed-genre test set, drawn from the web and curated manually for high-quality segment pairs.

The multilingual low-resource translation for Indo-European languages task (MLLR, Section 6) aims to investigate the best approaches to deal with multilingual translation. Usually, multilingual translation is done with the help of a high-resourced language, e.g. English. In MLLR, we evaluate translation quality for Icelandic–Norwegian Bokmål–Swedish (North-Germanic) and Catalan–Italian–Occitan–Romanian (Romance). Higher resourced languages (Danish, German, English, Spanish, French and Portuguese) are allowed for training but not evaluated. We focus on a specific domain: cultural heritage documents are extracted from Europeana and Wikipedia, a domain where named entities may also play a role in translation quality. The evaluation is done at language family level with a combination of automatic metrics (BLEU, TER, chrF, BertScore and COMET) and complemented by a manual evaluation on a subset of language pairs.

The automatic post-editing (APE) task (Section 7) focuses on another MT-related problem: the correction of machine-translated text generated by an unknown system. In continuity with last year, in this seventh iteration of the task at WMT we focused on two language pairs (English-German and English-Chinese), using data drawn from English Wikipedia articles and translated with neural MT systems. The evaluation was carried out both automatically – with TER and BLEU respectively used as primary and secondary metric - and manually – with the same direct assessment method used for the news translation task.

2 News Translation Task

This recurring WMT task assesses the quality of MT on text from the news domain. As in the previous year, we included Chinese, Czech, German, Japanese and Russian (to and from English) as well as French↔German. New language pairs for this year were Icelandic and Hausa (to and from English) as well as Bengali↔Hindi and Xhosa↔Zulu.

2.1 Test Data

As in previous years, the test sets consist of unseen translations prepared specially for the task. The test sets are publicly released to be used as translation benchmarks in the coming years. Here we describe the production and composition of the test sets.

The source texts for the test sets were all extracted from online news sites, with the exception of Bengali↔Hindi and Xhosa↔Zulu, which were part of the FLORES-101 benchmark (Goyal et al., 2021) and extracted from Wikipedia. The sources used for the online news are shown in Table 1, and all articles are from the second half of 2020. For the French↔German task, we specifically selected financial and economic news, whereas for the other news sources, we randomly selected articles from general online news, including politics, sports, international and local events.

For all language pairs, we aimed for a test set size of 1000 sentences, and to ensure that the test sets were "source-original", in that the source text is the original article and the target text is the translation. This is to avoid "translationese" effects on

the source language, which can have a detrimental effect on the accuracy of evaluation (Freitag et al., 2019; Laubli et al., 2020; Graham et al., 2020). The exceptions were Chinese→English, where we used a larger test set of 1948 sentences, and the FLORES-101 test sets which were around 500 sentences, and derived from English source documents. For language pairs that were new this year (i.e. Icelandic↔English and Hausa↔English) we prepared development sets using the same process as the test set, but concatenating both translation directions into the same set. For each translated article in the development set, the direction of translation is clearly identified.

For WMT20, we experimented with using test sources with line (segment) boundaries at paragraphs (not sentences) for some language pairs, but we found no evidence that translators used their new freedom to reorganise sentences, and the longer lines possibly made evaluation more difficult, so we reverted to a sentence-per-line format this year. For selected language sources (Czech, German and English, when translated into the recurring languages) we retained the paragraph boundaries from the original articles, but within the paragraphs, the sentences were in separate segments. It was up to the participating systems to make use of the paragraph breaks or not, but the systems were expected to preserve the segment boundaries.

The test sets for WMT21 were released using a new XML format, replacing the "pseudo xml" SGML format which had been used for many years. The advantages of the new format are: (i) it can be processed with standard XML tools, and there is no longer any doubt about how to treat special XML characters such as the ampersand ("&"); (ii) the source, all references and all submissions can be contained in one convenient XML file; (iii) the metadata better matches the needs of the task, and can be extended as necessary. We created simple tools for converting from text-based files to the new XML format.[4]

The translation of the test sets was performed by professional translation agencies, according to the brief supplied in Appendix B. Several language pairs got special attention. For Chinese↔English, Russian↔English and German↔English, we obtained a second reference in each direction from a different translation agency, labelled "B". For German↔English, the "B" reference was found to be a post-edited version of one of the participating online systems, so we had to discard it. Microsoft then sponsored a third independent translation, labelled "C", and the metrics task organizers with the support from Google later provided yet another German↔English reference, discussed only in Freitag et al. (2021b) as "D". For Czech↔English, the first reference (labelled "A") which served in reference-based manual evaluations, was provided by a translation agency in both directions. The second Czech↔English reference (labelled "B") which served as another system in the competition was provided by professional translators recruited from teachers and students of translation studies into Czech and three students and graduates of translation studies and one translator, English native speaker, into English.

2.2 Training Data

As in past years we provided a selection of parallel and monolingual corpora for model training, and development sets to tune system parameters. Participants were permitted to use any of the provided corpora to train systems for any of the language pairs. As well as providing updates on many of the previously released data sets, we included several new data sets, mainly to support the new language pairs.

Our training data includes the latest version of ParaCrawl (Bañón et al., 2020) for all language pairs where it is available. New for this year is a ParaCrawl corpus for Chinese↔English, which contains 14M sentences, as well as a small Hausa↔English ParaCrawl. The JParaCrawl corpus (for Japanese↔English) is constructed in a similar way to ParaCrawl, but by a different group (Morishita et al., 2020).

For Icelandic↔English we used the recently released ParIce (Barkarson and Steingrímsson, 2019) a source of parallel data, and the Icelandic Gigaword corpus for monolingual data (Steingrímsson et al., 2018).

For Hausa↔English, the data was mainly drawn from Opus (Tiedemann and Nygaard, 2004), which is mostly religious and IT localisation text. We added a small (< 6000) parallel sentence corpus extracted from the website of Ayatollah Khamenei,[5] now only accessible using the

[4] https://github.com/wmt-conference/wmt-format-tools

[5] https://english.khamenei.ir/

English	ABC News (5), Al Jazeera (1), All Africa (2), BBC (4), Brisbane Times (3), CBS LA (1), CBS News (3), CNBC (1), CNN (1), Daily Express (4), Daily Mail (1), Egypt Independent (3), Fox News (2), Guardian (6), LA Times (1), London Evening Standard (2), Metro (1), NDTV (7), New York Times (2), RTE (1), Russia Today (5), Seattle Times (4), Sky (1), The Independent (1), The Sun (2), UPI (1), VOA (1), news.com.au (1), novinite.com (1),
Chinese	China News (76), Hunan Ribao (5), Jingji Guancha Bao (3), Macao Government (2), Nhan Dan (3), RFI Chinese (6), VOA Chinese (3), Xinhua (57), tsrus.cn (1),
Czech	Aktuálně (4), Blesk (5), Denik (3), Dnes (1), E15 (1), Haló noviny (5), Hospodářské Noviny (1), Idnes (2), Lidovky (7), Mediafax (6), Novinky (6), Týden (1), Tydenek Homer Mostecka (1), ČT24 (4), Česká Pozice (6), Česká Televize (4), České Noviny (4), Český Rozhlas (1),
German	Aachener Nachrichten (1), Abendzeitung Mn̈chen (1), Abendzeitung Nürnberg (1), Allgemeine Zeitung (1), Augsburger-allgemeine (1), Braunschweiger Zeitung (1), Das Bild (3), Dresdner Neueste Nachrichten (1), Euronews (1), Frankfurter Allgemeine Zeitung (1), Freie Presse (1), Handelsblatt (1), Hessische/Niedersaechsische Allgemeine (1), Infranken (3), Kurier (2), Lampertheimer Zeitung (3), Landeszeitung (1), Main-Netz (1), Mainpost (1), Mittelbayerische Zeitung (2), Mitteldeutsche Zeitung (2), Morgenpost (2), Neue Presse (Coburg) (2), Nordbayerischer Kurier (3), OE24 (1), Passauer Neue Presse (2), Peiner Allgemeine Zeitung (2), Pforzheimer Zeitung (1), Potsdamer Neueste Nachrichten< (1), Rhein Zeitung (2), Rundschau online (1), Söster Anzeiger (1), Salzburger Nachrichten (1), Schwäbische (2), Schwäbische post (2), Schwarzwälder Bote (2), Tiroler Tageszeitung (2), Usinger Anzeiger (1), Westfälische Nachrichten (2), Wienerzeitung (1),
Hausa	Deutsche Welle (7), Freedom radio (22), Leadership (19), Premium Times (20), RFI Hausa (10), VOA Hausa (18), VON Hausa (4),
Japanese	Fukui Shimbun (1), Hokkaido Shimbun (5), Iwate Nippo (3), Saga Shimbun (3), Sanyo Shimbun (4), Shizuoka Shimbun (11), Ube nippo Shimbun (2), Yaeyama mainichi shimbun (1), Yahoo (49), Yamagata Shimbun (2),
Russian	Altapress (1), Altyn-orda (1), Argumenti Nedely (5), Argumenty i Fakty (6), Armenpress (1), BBC Russian (1), Delovoj Peterburg (1), ERR (5), Gazeta (4), Interfax (3), Izvestiya (11), Kommersant (1), Komsomolskaya Pravda (7), Lenta (6), Lgng (2), Moskovskij Komsomolets (9), Novye Izvestiya (1), Ogirk (1), Parlamentskaya Gazeta (3), Rossiskaya Gazeta (5), Russia Today (8), Russkaya Planeta (1), Sovsport (2), Sport Express (9), Tyumenskaya Oblast Segodnya (1), VOA Russian (1), Vedomosti (2), Vesti (6), Xinhua (3),
German (economic)	Aachener Nachrichten (1), Abendzeitung Mn̈chen (1), Das Bild (1), Der Spiegel (2), Epoch Times (1), Frankfurter Allgemeine Zeitung (6), Handelsblatt (17), Haz (2), Kurier (4), Lübecker Nachrichten (1), Mindener Tageblatt (1), Mittelbayerische Zeitung (1), NZZ (1), Neue Westfälische (1), Onetz (1), Passauer Neue Presse (2), Rheinische Post (1), Russia Today (3), Süddeutsche Zeitung (8), Salzburger Nachrichten (2), Tiroler Tageszeitung (1), Volksstimme (1), Yahoo (1), come-on.de (1),
French (econmic)	Algérie Presse Service (3), Aujourd'hui le Maroc (5), Dernière Heure (4), Dernières Nouvelles d'Alsace (1), Euronews (2), L'Independant (1), L'express (2), La Croix (4), La Meuse (3), La Tribune (4), La Venir (1), Le Devoir (3), Le Figaro (17), Le Monde (5), Le Quotidien (1), Les Echos (1), Liberté Algerie (1), Libre Belgium (1), Madagascar tribune (1), Metro Canada (1), Nice Matin (1), Nouvel Obs (6), Russia Today (4), VOA Afrique (2),

Table 1: Composition of the test sets. The economic arcticles were used for French↔German only. We did not record the sources for the Icelandic articles, and the Bengali, Hindi, Xhosa and Zulu articles were from Wikipedia.

Europarl Parallel Corpus

	Czech ↔ English		German ↔ English		German ↔ French	
Sentences	645,241		1,825,745		1,801,076	
Words	14,948,900	17,380,340	48,125,573	50,506,059	47,517,102	55,366,136
Distinct words	172,452	63,289	371,748	113,960	368,585	134,762

News Commentary Parallel Corpus

	Czech ↔ English		German ↔ English		Russian ↔ English	
Sentences	253,456		388,813		331,596	
Words	5,674,011	6,270,051	9,921,515	9,840,910	8,469,701	8,820,805
Distinct words	176,403	70,774	215,101	86,518	207,701	82,938

	Chinese ↔ English		Japanese ↔ English		German ↔ French	
Sentences	313,934		1,851		296,022	
Words	–	7,982,550	–	45,438	7,671,513	9,346,818
Distinct words	–	76,372	–	6,280	185,348	87,481

Common Crawl Parallel Corpus

	German ↔ English		Czech ↔ English		Russian ↔ English		French ↔ German	
Sentences	2,399,123		161,838		878,386		622,288	
Words	54,575,405	58,870,638	3,529,783	3,927,378	21,018,793	21,535,122	13,991,973	12,217,457
Distinct words	1,640,835	823,480	210,170	128,212	764,203	432,062	676,725	932,137

ParaCrawl Parallel Corpus

	German ↔ English		Czech ↔ English		Chinese ↔ English	
Sentences	82,638,202		14,083,311		14,170,585	
Words	1,543,410,882	1,613,780,145	240,233,151	260,801,934	–	253,776,811
Distinct Words	15,256,769	7,765,311	2,655,118	1,972,030	–	1,871,639

	Japanese ↔ English		Russian ↔ English		French ↔ German	
Sentences	10,120,013		12,654,509		7,222,574	
Words	–	274,368,443	232,950,488	266,368,340	145,190,707	123,205,701
Distinct Words	–	2,051,246	2,913,181	1,816,590	1,534,068	2,368,682

	Icelandic ↔ English		Hausa ↔ English	
Sentences	2,392,422		158,968	
Words	39,528,080	42,454,372	4,041,027	3,957,605
Distinct Words	709,945	416,986	102,962	101,049

EU Press Release Parallel Corpus

	Czech ↔ English		German ↔ English	
Sentences	452,411		1,631,639	
Words	7,214,324	7,748,940	26,321,432	27,018,196
Distinct words	141,077	83,733	402,533	197,030

Yandex 1M Parallel Corpus

	Russian ↔ English	
Sentences	1,000,000	
Words	24,121,459	26,107,293
Distinct	701,809	387,646

CzEng v2.0 Parallel Corpus

	Czech ↔ English	
Sentences	60,980,645	
Words	757,316,261	848,016,692
Distinct	3,684,081	2,493,804

WikiTitles Parallel Corpus

	Chinese ↔ English		Czech ↔ English		German ↔ English		Hausa ↔ English	
Sentences	922,194		410,977		1,474,196		7,501	
Words	–	2,549,611	990,191	1,065,417	3,219,123	3,763,461	14,285	14,629
Distinct	–	380,234	218,992	186,375	674,927	573,280	7,855	7,827

	Icelandic ↔ English		Japanese ↔ English		Russian ↔ English		German ↔ French	
Sentences	50,181		757,052		1,189,097		1,006,563	
Words	90,620	100,847	–	2,016,400	3,244,102	3,261,299	2,142,193	2,543,265
Distinct	40,570	34,440		281,880	534,392	457,933	503,342	444,330

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 Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library).

CCMT Corpus

	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

Extra Japanese-English Parallel Data

	Subtitles		Kyoto		TED	
Sentences	2,801,388		443,849		223,108	
Words	–	23,933,060	–	11,622,252		4,554,409
Distinct	–	161,484	–	191,885	–	60,786

Extra Hausa-English Parallel Data

	Khamenei		Opus	
Sentences	5,837		584,004	
Words	217,543	167,466	8,385,179	8,994,622
Distinct	6,075	7,942	219,203	193,518

CC-Aligned

	Bengali↔Hindi		Xhosa↔Zulu	
Sentences	3,365,142		94,323	
Words	40,782,432	45,609,689	1,689,086	1,658,266
Distinct	996,612	860,033	186,070	173,148

United Nations Parallel Corpus

	Russian ↔ English		Chinese ↔ English	
Sentences	23,239,280		15,886,041	
Words	570,099,284	601,123,628	–	425,637,920
Distinct	1,446,782	1,027,143	–	769,760

Synthetic parallel data (both directions combined)

	Czech ↔ English		Russian ↔ English		Chinese ↔ English	
Sentences	126,828,081		76,133,209		19,763,867	
Words	2,351,230,606	2,655,779,234	1,511,996,711	1,698,428,744	–	416,567,173
Distinct	5,745,323	3,840,231	5,928,141	3,889,049	–	1,188,933

Wikimatrix Parallel Data

	Czech ↔ English		German ↔ English		Japanese ↔ English		Icelandic ↔ English	
Sentences	2,094,650		6,227,188		3,895,992		313,875	
Words	34,801,119	39,197,172	113,445,806	118,077,685	–	72,320,248	5,395,042	6,475,011
Distinct	1,068,844	798,095	2,855,263	1,827,785	–	1,106,529	328,369	231,192

	Russian ↔ English		Chinese ↔ English		German ↔ French	
Sentences	5,203,872		2,595,119		3,350,816	
Words	93,828,313	102,937,537	–	58,615,891	68,249,384	59,422,699
Distinct	2,233,043	1,592,190	–	1,059,537	1,067,450	1,844,533

Figure 2: 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 Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library).

News Language Model Data

	English	German	Czech	Russian	Japanese
Sentences	274,929,980	386,987,716	97,396,609	111,118,861	14,389,733
Words	6,782,988,670	7,951,191,279	1,760,715,133	2,010,171,968	–
Distinct words	8,329,647	39,524,377	5,960,637	5,679,507	–

	Icelandic	Chinese	French	Hausa	Hindi	Bengali
Sentences	534,647	10,771,382	96,402,399	272,966	46,187,245	10,101,626
Words	9,653,929	–	2,338,364,059	7,305,501	872,106,937	148,586,981
Distinct words	308,924	–	3,975,116	125,350	2,752,071	1,091,788

Document-Split News LM Data (not dedudped)

	Czech	English	German
Sentences	142,478,129	531,904,913	739,041,709
Words	2,221,995,079	11,472,609,712	12,524,314,673
Distinct words	5,744,574	8,595,778	26,849,693

Common Crawl Language Model Data

	English	German	Czech	Russian
Sent.	3,074,921,453	2,872,785,485	333,498,145	1,168,529,851
Words	65,104,585,881	65,147,123,742	6,702,445,552	23,332,529,629
Dist.	342,149,665	338,410,238	48,788,665	90,497,177

	Chinese	Icelandic	Hausa	French
Sent.	1,672,324,647	24,627,579	1,467,326	4,898,012,445
Words	–	595,998,326	20,082,665	126,364,574,036
Dist.	–	7,483,421	688,610	363,878,959

Figure 3: Statistics for the monolingual training sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library).

Test Sets

	Czech → EN			EN → Czech			German → EN			EN → German		
Lines.	1000			1002			1000			1002		
Words	17,914	22,080	22,570	27,454	25,907	27,190	18,190	20,668	20,541	27,454	28,273	28,673
Distinct words	6,457	4,032	4,425	5,374	8,295	8,577	5,115	4,012	3,980	5,374	6,841	6,697

	Chinese → EN		EN → Chinese			Russian → EN			EN → Russian		
Lines.	1948		1002			1000			1002		
Words	–	72,334	27,454	–	–	17,796	21,400	21,185	27,454	26,413	26,253
Distinct words	–	8,290	5.374	–	–	6,315	4,214	4,230	5,374	8,591	8,377

	Icelandic → EN		EN → Icelandic		Japanese → EN		EN → Japanese		Hausa ↔ EN	
Lines.	1000		1000		1005		1000		997	
Words	19,930	22,749	26,467	25,557	–	28,846	26,467	–	31,362	27,519
Distinct words	5,282	3,773	5,258	6,614	–	5,001	5,258	–	4,032	4,240

	EN ↔ Hausa		Bengali → Hindi		Hindi → Bengali		Xhosa → Zulu		Zulu ↔ Xhosa	
Lines.	1000		503		509		503		509	
Words	26,467	33,915	11,439	14,133	14,286	11,136	9,180	9,314	9,320	9,065
Distinct words	5,258	4,713	4,514	3,686	3,402	4,091	5,499	5,265	4,961	5,093

	French → German		German → French	
Lines.	1026		1000	
Words	30,143	26,353	18,801	26,407
Distinct words	5,395	6,021	5,198	4,613

Figure 4: Statistics for the test sets used in the translation task. In the cases that there are three word counts, these are for source, first target translation, and second target translation. The number of words and the number of distinct words (case-insensitive) is based on the Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library).

Wayback Machine.[6]

For the two FLORES-101 language pairs (i.e. Bengali↔Hindi and Xhosa↔Zulu) all training data is from the CC-Aligned corpus (El-Kishky et al., 2020).

Other language pairs used the same data sets as last year, with updates wherever available.

The monolingual data we provided was similar to last year's, with a 2020 news crawl[7] added to all the news corpora. Note that news crawl now includes 59 languages, so is not limited to languages used in WMT. In addition, we provided versions of the news corpora for Czech, English and German, with both the document and paragraph structure retained. In other words, we did not apply sentence splitting to these corpora, and we retained the document boundaries and text ordering of the originals.

Some statistics about the training and test materials are given in Figures 1, 2, 3 and 4.

2.3 Submitted Systems

In 2021, we received a total of 173 submissions. The participating institutions 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 online MT systems (originating from 5 services), which we anonymized as ONLINE-A,B,G,W,Y. All submissions, sources and references are made available via github[8].

The collect submissions, we used the submission tool, OCELoT,[9] replacing the matrix that has been used up until 2019. Using OCELoT gives us more control over the submission and scoring process, for example we are able to limit the number of test submissions by each team, and we also display the submissions anonymously to avoid publishing any automatic scores.

For presentation of the results, systems are treated as either *constrained* or *unconstrained*. When the system submitters report that they were only trained on the provided data, we class them as constrained. The online systems are treated as unconstrained during the automatic and human evaluations, since we do not know how they were built.

In Appendix C, we provide brief details of the submitted systems, for those where the authors provided such details.

3 Human Evaluation

A human evaluation campaign is run each year to assess translation quality and to determine the official ranking of systems taking part in the news translation task. This section describes how data for the human evaluation is prepared, the process of collecting human assessments, and computation of the official results of the shared task.

3.1 Direct Assessment

We have employed Direct Assessment (DA, Graham et al., 2013, 2014, 2016) as the primary mechanism for evaluating systems since running a comparison of DA and relative ranking in 2016 (Bojar et al., 2016). DA has several important features including accurate quality control of crowdsourcing. With DA human evaluation, human assessors are asked to rate a given translation by how adequately it expresses the meaning of the corresponding reference translation or source language input on an analogue scale, which corresponds to an underlying absolute 0–100 rating scale.[10]

3.1.1 Source and Reference-based Evaluations

The original definition of DA provides human assessors with a reference translation. The benefit of this reference-based evaluation is that only speakers of the target language are needed, but the quality of the reference translation becomes critical and even if flawless, evaluating against a single reference translation could bias evaluators towards that reference.

In 2018, we trialled source-based (or "bilingual") evaluation for the first time, for English to Czech translation. In this configuration, the human assessor is shown the source input and system output only (with no reference translation shown). The assessor thus has to understand both the source and target languages very well but the quality of the reference is no longer vital. In fact, the human-generated reference can be included in the evaluation as an additional system to provide an estimate of human performance.

[6] https://archive.org/web/

[7] http://data.statmt.org/news-crawl

[8] https://github.com/wmt-conference/wmt21-news-systems

[9] https://github.com/AppraiseDev/OCELoT

[10] No sentence or document length restriction is applied during manual evaluation. Direct Assessment is also employed for evaluation of video captioning systems at TRECvid (Graham et al., 2018; Awad et al., 2019, 2021) and multilingual surface realisation (Mille et al., 2018, 2019).

Team	Language Pairs	System Description
AFRL	ru-en	(Erdmann et al., 2021)
ALLEGRO.EU	en-is,is-en	(Koszowski et al., 2021)
AMU	ha-en,en-ha	(Nowakowski and Dwojak, 2021)
BJTU-NMT	en-zh	(no associated paper)
BORDERLINE	en-zh,de-en,zh-en	(Wang et al., 2021)
BUPT-RUSH	en-zh,en-ja,en-de	(no associated paper)
CAPITALMARVEL	en-zh,en-ja,ja-en	(no associated paper)
CUNI-DOCTRANSFORMER	en-cs,cs-en	(Gebauer et al., 2021)
CUNI-MARIAN-BASELINES	en-cs	(Gebauer et al., 2021)
CUNI-TRANSFORMER2018	en-cs,cs-en	(Gebauer et al., 2021)
DIDI-NLP	zh-en	(no associated paper)
EPHEMERALER	en-zh,en-ja	(no associated paper)
ETRANSLATION	fr-de,en-cs,en-de	(Oravecz et al., 2021)
FACEBOOK-AI	ha-en,en-zh,en-ha,en-is,en-ja,de-en, zh-en,en-ru,en-cs,cs-en,ru-en,en-de, ja-en,is-en	(Tran et al., 2021)
FJDMATH	xh-zu	(Martinez, 2021)
GTCOM	ha-en,bn-hi,en-ha,zu-xh,hi-bn,xh-zu	(Bei and Zong, 2021)
HAPPYNEWYEAR	en-zh,zh-en	(no associated paper)
HAPPYPOET	en-zh,de-en,en-de	(no associated paper)
HW-TSC	ha-en,en-zh,bn-hi,en-ha,en-is,en-ja, zu-xh,de-en,zh-en,hi-bn,xh-zu,en-de, ja-en,is-en	(Wei et al., 2021)
ICL	en-zh,de-en,zh-en,en-de	(no associated paper)
IIE-MT	zh-en,ja-en	(no associated paper)
ILLINI	en-ja,ja-en	(Le et al., 2021)
KWAINLP	zh-en,ja-en	(no associated paper)
LAN-BRIDGE-MT	en-zh,en-is	(no associated paper)
LISN	fr-de,de-fr	(Xu et al., 2021)
MACHINE-TRANSLATION	en-zh,zh-en	(no associated paper)
MANIFOLD	ha-en,en-ha,en-is,de-en,en-ru,de-fr, ru-en,en-de,is-en	(no associated paper)
MIDEIND	en-is,is-en	(Jónsson et al., 2021)
MISS	en-zh,en-ja,zh-en,ja-en	(Li et al., 2021b)
MOVELIKEAJAGUAR	en-zh,en-ja,ja-en	(no associated paper)
MS-EGDC	ha-en,bn-hi,en-ha,zu-xh,hi-bn,xh-zu	(Hendy et al., 2021)
NIUTRANS	ha-en,en-zh,en-ha,en-is,en-ja,zh-en, en-ru,ru-en,ja-en,is-en	(Zhou et al., 2021)
NJUSC-TSC	en-zh,zh-en	(no associated paper)
NUCLEAR-TRANS	en-zh,en-de	(no associated paper)
NVIDIA-NEMO	de-en,en-ru,ru-en,en-de	(Subramanian et al., 2021)
P3AI	ha-en,en-zh,en-ha,fr-de,de-en,zh-en, de-fr,en-de	(Zhao et al., 2021)
SMU	en-zh,de-en,zh-en	(no associated paper)
TALP-UPC	fr-de,de-fr	(Escolano et al., 2021)
TRANSSION	ha-en,bn-hi,en-ha,zu-xh,hi-bn,xh-zu	(no associated paper)
TWB	ha-en,en-ha	(no associated paper)
UEDIN	ha-en,bn-hi,en-ha,de-en,hi-bn,en-de	(Chen et al., 2021; Pal et al., 2021)
UF	en-zh,de-en,zh-en,en-de	(no associated paper)
VOLCTRANS-AT	de-en,en-de	(Qian et al., 2021)
VOLCTRANS-GLAT	de-en,en-de	(Qian et al., 2021)
WATERMELON	de-en	(no associated paper)
WECHAT-AI	en-zh,en-ja,en-de,ja-en	(Zeng et al., 2021)
WINDFALL	en-zh	(no associated paper)
XMU	zh-en,ja-en	(no associated paper)
YYDS	en-zh,zh-en	(no associated paper)
ZENGHUIMT	en-zh,zh-en	(Zeng, 2021)
ZMT	ha-en,en-ha	(no associated paper)

Table 2: Participants in the shared translation task. The translations from the 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.

For both reference and source-based evaluation, we require human assessors to only evaluate translation *into* their native language. Following WMT19 and WMT20, we thus again use the source-based evaluation only for out-of-English language pairs. This is especially relevant since we have a large group of volunteer human assessors with native language fluency in non-English languages and high fluency in English, while we generally lack the reverse, i.e. native English speakers with high fluency in non-English languages.

We use different implementation and human annotators for into-English and out-of-English. We describe the approaches separately. Reference-based (monolingual) into-English human evaluation is described in Section 3.2, while source-based (bilingual) out-of-English and non-English human evaluation is described in Section 3.3. A third, simplified annotation was used for Bengali↔Hindi and Xhosa↔Zulu, Section 3.4.

3.1.2 Translationese

Prior to WMT19, all the test sets included a mix of sentence pairs that were originally in the source language, and then translated to the target language, and sentence pairs that were originally in the target language but translated to the source language. The inclusion of the latter "reverse-created" sentence pairs has been shown to introduce biases into the evaluations, particularly in terms of BLEU scores (Graham et al., 2020). Therefore we have avoided it for all language pairs, apart from Bengali↔Hindi and Xhosa↔Zulu, where the texts are all translated from English.

3.1.3 Document Context

As mentioned already in our discussion in WMT18 and as also established within the community (Läubli et al., 2018b; Toral et al., 2018a), evaluating sentences out of their document context can skew the results. The effect is particularly pronounced when comparing human and machine translation, where it is observed that evaluators tend to rate the human translation higher (relative to the machine translation) when the translations are viewed in context. Human translators always have access to the document context when translating to create the references.

In WMT19, we experimented with a DA style that considers document context in a simple way.

Language Pair	Sys.	Assess.	Assess/Sys
Czech→English	9	10,651	1,183.4
German→English	20	25,718	1,285.9
Hausa→English	14	17,321	1,237.2
Icelandic→English	10	11,124	1,112.4
Japanese→English	16	17,055	1,065.9
Russian→English	11	11,499	1,045.4
Chinese→English	24	44,268	1,844.5
Total to-English	104	137,636	1,323.4

Table 3: Amount of data collected in the WMT21 manual evaluation campaign for evaluation into-English; after removal of quality control items.

Dubbed "SR+DC" (segment rating with document context), this method presents one segment at a time but the segments are no longer shuffled (as in "SR−DC", segment rating without document context). Instead, they are provided in the order in which they appear in the document. The implementation still has the limitation that the assessors cannot go back to the previous segment.

An improved alternative to "SR+DC" is to offer the full document and allow the assessors to review their segment-level ratings. We call this setup "SR+FD" (segment ranking in a full document) and illustrate the user interface in Appraise in Figure 5.[11]

This year, for all language pairs for which document context was available, we include it when evaluating translations. Note that the ratings are nevertheless collected on the segment level, motivated by the power analysis described in Graham et al. (2019) and Graham et al. (2020). The particular details on how document context is made available to assessors depends on the translation direction, as described in more detail in Sections 3.2 to 3.4.

3.2 Human Evaluation of Translation into-English

In terms of the News translation task manual evaluation for into-English language pairs, a total of 589 turker accounts were involved.[12] 488,396 translation assessment scores were submitted in total by the crowd, of which 170,194 were provided by workers who passed quality control.[13]

System rankings are produced from a large set of human assessments of translations, each of which indicates the absolute quality of the out-

[11] Compare with Figures 3 and 4 in Bojar et al. (2019).

[12] Numbers do not include the 1,078 workers on Mechanical Turk who did not pass quality control.

[13] Numbers include quality control segments.

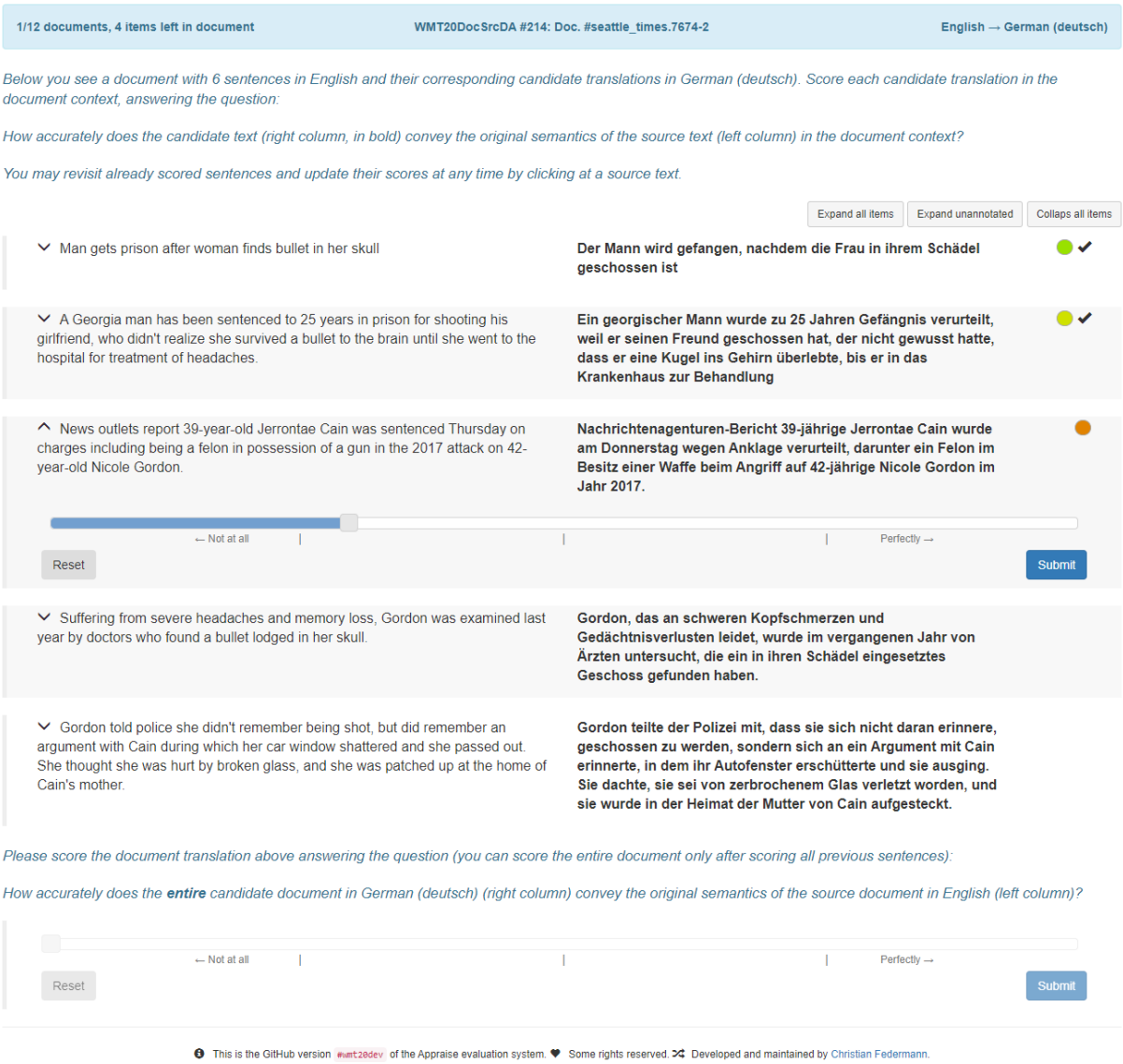


Figure 5: Screen shot of the document-level DA (SR+FD, segment rating within the full document) configuration in the Appraise interface for an example assessment from the human evaluation campaign. The annotator is presented with the entire translated document randomly selected from competing systems (anonymized) and is asked to rate the translation of individual segments and then entire document on sliding scales.

put of a system. Table 3 shows total numbers of human assessments collected in WMT21 for into-English language pairs contributing to final scores for systems.[14]

3.2.1 Crowd Quality Control

Collection of segment-level ratings with document context (SR+DC, Segment Rating + Document Context) involved constructing HITs so that each sentence belonging to a given document (produced by a single MT system) was displayed to and rated in turn by the human annotator.

We then injected the three kinds of quality control translation pairs described in Table 4: we repeat pairs expecting a similar judgment (Repeat Pairs), damage MT outputs expecting significantly worse scores (Bad Reference Pairs) and use references instead of MT outputs expecting high scores (Good Reference Pairs). For each of these three types, we include the MT output, along with its corresponding control item.

HITs were then constructed as follows, with as close as possible to 100 segments in a single HIT:

1. All documents produced by all systems are pooled;[15]

[14] Number of systems for WMT21 includes four "human" systems comprising human-generated reference translations used to provide human performance estimates.

[15] If a "human" system is included to provide a human per-

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 4: Standard DA HIT structure quality control translation pairs hidden within 100-translation HITs, numbers of items are provided in parentheses.

2. Documents are then sampled at random (without replacement) and assigned to the current HIT until the current HIT contains close to (but less than) 70 segments

3. Once documents amounting to close to 70 segments have been assigned to the current HIT, we select a subset of these documents to be paired with quality control documents; this subset is selected by repeatedly checking if the addition of the number of the segments belonging to a given document (as quality control items) will keep the total number of segments in the HIT below 100; if this is the case, it is included; otherwise it is skipped until the addition of all documents has been checked. In doing this, the HIT is structured to bring the total number of segments as close as possible to 100 segments.

4. Once we have selected a core set of original system output documents and a subset of them to be paired with quality control versions for each HIT, quality control documents are automatically constructed by altering the sentences of a given document into a mixture of three kinds of quality control items used in the original DA segment-level quality control: bad reference translations, reference translations and exact repeats (see below for details of bad reference generation and Table 5 for numbers of words replaced in document segments);

5. Finally, the documents belonging to a HIT are shuffled.

Construction of Bad References 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 mostly 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 listed in Table 5.

formance estimate, it is also considered a system during quality control set-up.

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 $\rfloor$

Table 5: Number of words replaced when constructing quality control items.

Quality Filtering 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-source worker who has not demonstrated that he/she can reliably score bad reference translations significantly lower than corresponding genuine system

	All	(A) Sig. Diff. Bad Ref.	(A) & No Sig. Diff. Exact Rep.
Czech→English	290	73 (25%)	68 (93%)
German→English	605	162 (27%)	150 (93%)
Hausa→English	423	109 (26%)	101 (93%)
Icelandic→English	273	75 (27%)	67 (89%)
Japanese→English	315	103 (33%)	91 (88%)
Russian→English	187	84 (45%)	77 (92%)
Chinese→English	617	195 (32%)	178 (91%)
Total	**1,694**	**589 (35%)**	**544 (92%)**

Table 6: Number of crowd-sourced workers taking part in the reference-based SR+DC campaign; (A) those whose scores for bad reference items were significantly lower than corresponding MT outputs; those of (A) whose scores also showed no significant difference for exact repeats of the same translation; note: many workers evaluated more than one language pair.

output translations are filtered out. A paired significance test is applied to test if degraded translations are consistently scored lower 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.

Table 6 shows the number of workers participating in the into-English translation evaluation who met our filtering requirement in WMT21 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. We removed data from the non-reliable workers in all language pairs.

3.2.2 Producing the Human Ranking

This year all rankings (for to-English translation) were arrived at via segment ratings presented one at a time in their original document order (SR+DC).

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.

Average standardized scores for individual segments belonging to a given system were then computed, before the final overall DA score for a given system is computed as the average of its segment scores (Ave z in Table 7). Results are also reported for average scores for systems, computed in the same way but without any score standardization applied (Ave % in Table 7).

Human performance estimates arrived at by evaluation of human-produced reference translations are denoted by "HUMAN" in all tables.

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. Rank ranges are based on the same head-to-head statistical significance tests. For instance, if a system is statistically significantly worse than 2 other systems, and not statistically different from 4 other systems, its rank is reported as 3–6 (the top of the rank range is 2+1, the bottom 2+4).

All data collected during the human evaluation is available at `http://www.statmt.org/wmt21/results.html`. Appendix A shows the underlying head-to-head significance test official results for all pairs of systems and also reports BLEU, chrF, and COMET scores.

3.3 Bilingual Human Evaluation

Human evaluation for nine out-of-English and non-English translation directions used a source-based (sometimes called "bilingual") direct assessment of individual segments in the full document context (SR+FD), as established in WMT20 (Barrault et al., 2020).

In an attempt to break more ties among the participating systems, we also ran a second stage of annotation using segment-level contrastive source-based DA ignoring document context (labelled "contr:SR−DC") for top-10 systems (plus human references) for 3 out-of-English language pairs. Details on the second stage are in Section 3.3.5.

In the source-based DA campaign, we collected 303,627 assessments in total after excluding quality control items and users who did not pass the quality control. The contrastive source-based DA campaign provided 64,031 translation assessments. The total numbers of collected assessments per language pair are presented in Table 8. For data collection, we used the open-source Appraise Evaluation Framework (Federmann, 2012) for both assessment types.

Czech→English

Rank	Ave.	Ave. z	System
1–2	77.8	0.111	Facebook-AI
1–2	78.4	0.081	Online-A
3–6	72.0	0.008	CUNI-DocTransf
3–6	74.0	−0.005	Online-B
3–8	71.5	−0.008	CUNI-Trf2018
3–8	74.5	−0.032	Online-W
5–9	67.2	−0.039	Online-G
7–9	74.4	−0.084	Online-Y
5–9	75.6	−0.085	HUMAN-B

German→English

Rank	Ave.	Ave. z	System
1–5	71.9	0.126	Borderline
1–6	73.5	0.124	Online-A
1–4	78.6	0.122	Online-W
4	79.5	0.113	UF
3–8	73.2	0.106	VolcTrans-AT
4–9	77.5	0.100	Facebook-AI
5–12	75.8	0.068	ICL
4–12	73.4	0.048	Online-G
8–17	69.7	0.016	Online-B
7–17	71.3	0.016	Online-Y
7–17	71.6	0.010	VolcTrans-GLAT
5–16	69.6	0.007	P3AI
9–19	70.6	−0.008	SMU
9–17	73.1	−0.008	UEdin
9–17	69.1	−0.010	NVIDIA-NeMo
10–19	69.9	−0.035	Manifold
15–20	67.0	−0.043	Watermelon
7–17	71.8	−0.061	happypoet
16–20	66.8	−0.081	HUMAN-C
18–20	66.0	−0.120	HW-TSC

Hausa→English

Rank	Ave.	Ave. z	System
1	74.4	0.248	Facebook-AI
2–4	68.8	0.118	Online-B
3–7	66.6	0.062	TRANSSION
2–6	66.5	0.059	ZMT
3–6	69.0	0.059	GTCOM
3–9	65.3	0.029	HW-TSC
5–19	65.2	0.002	MS-EgDC
6–10	60.1	−0.031	P3AI
6–10	62.4	−0.032	NiuTrans
8–11	63.5	−0.090	Online-Y
10–12	59.6	−0.112	Manifold
11–13	60.4	−0.173	AMU
12–13	58.2	−0.205	UEdin
14	56.9	−0.267	TWB

Icelandic→English

Rank	Ave.	Ave. z	System
1	74.5	0.293	Facebook-AI
2	74.8	0.112	Manifold
3–7	75.1	0.045	NiuTrans
3–8	71.3	0.028	Online-B
3–7	76.6	0.013	HW-TSC
3–7	69.7	0.009	Mideind
3–9	75.4	0.003	Online-A
6–9	70.1	−0.037	Allegro.eu
7–9	71.7	−0.080	Online-Y
10	65.2	−0.256	Online-G

Japanese→English

Rank	Ave.	Ave. z	System
1	73.8	0.141	HW-TSC
2–5	65.1	0.082	IIE-MT
2–6	68.6	0.046	NiuTrans
2–9	67.8	0.033	KwaiNLP
2–6	66.2	0.032	Facebook-AI
5–11	63.5	0.025	XMU
3–10	66.8	0.011	capitalmarvel
5–11	60.9	0.001	Online-B
6–11	61.5	−0.031	MiSS
5–11	66.7	−0.039	Online-W
7–12	59.3	−0.062	WeChat-AI
11–14	59.0	−0.080	Online-A
12–16	55.0	−0.140	Online-G
12–16	64.8	−0.157	movelikeajaguar
13–16	62.2	−0.189	Online-Y
13–16	55.4	−0.193	Illini

Russian→English

Rank	Ave.	Ave. z	System
1–5	77.5	0.137	NVIDIA-NeMo
1–4	73.9	0.130	Online-W
3–7	73.1	0.108	Online-B
1–7	73.3	0.089	HUMAN-B
2–7	71.7	0.060	Manifold
1–7	70.4	0.056	Facebook-AI
3–8	68.5	0.044	NiuTrans
7–10	65.1	0.016	Online-G
8–11	65.5	−0.014	AFRL
8–11	63.9	−0.022	Online-A
9–12	69.1	−0.123	Online-Y

Chinese→English

Rank	Ave.	Ave. z	System
1–5	75.0	0.042	NiuTrans
1–6	77.0	0.039	KwaiNLP
1–6	75.6	0.031	DIDI-NLP
1–9	74.1	0.019	HUMAN-B
1–9	71.7	0.016	HappyNewYear
2–19	74.0	−0.001	P3AI
4–18	70.5	−0.023	Borderline
4–19	72.6	−0.026	ICL
6–17	70.1	−0.029	MiSS
3–24	73.1	−0.031	IIE-MT
9–22	72.8	−0.032	Machine-Translation
7–21	70.6	−0.034	SMU
7–21	70.7	−0.036	yyds
6–20	70.1	−0.037	Facebook-AI
7–21	73.6	−0.042	Online-B
7–21	73.5	−0.050	ZengHuiMT
7–21	73.0	−0.062	HW-TSC
7–22	67.6	−0.068	XMU
12–24	76.0	−0.072	NJUSC-TSC
11–24	72.1	−0.082	Online-G
8–22	72.9	−0.087	Online-W
17–24	70.1	−0.103	UF
20–24	66.7	−0.106	Online-A
20–24	69.0	−0.174	Online-Y

Table 7: Official results of WMT21 News Translation Task for translation into-English (SR+DC). Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; rank ranges are based on the same test (for details, see Section 3.2.2); grayed entry indicates resources that fall outside the constraints provided.

Language Pair	Sys.	Assess.	Assess/Sys
English-Czech	12	50,491	4,207.6
English-German	22	24,689	1,122.2
English-Hausa	15	18,656	1,243.7
English-Icelandic	12	16,940	1,411.7
English-Japanese	16	43,991	2,749.4
English-Russian	11	31,632	2,875.6
English-Chinese	31	84,322	2,720.1
German-French	10	21,018	2,101.8
French-German	10	11,888	1,188.8
Total standard DA	139	303,627	2,184.4
English-Czech	12	19,279	1,606.6
English-German	12	23,212	1,934.3
English-Chinese	12	21,540	1,795.0
Total contrastive DA	36	64,031	1,778.6

Table 8: Amount of data collected in the WMT21 manual document- and segment-level evaluation campaigns for bilingual source-based evaluation out-of-English and non-English language pairs. The system counts include the human references (either 1 or 2 references, depending on language pair).

3.3.1 Sources of Human Annotators

We used three groups of annotators: participants in the News Shared Task, crowd-workers from the Toloka platform, and paid professional annotators sponsored by Microsoft.

We asked participants of the news task to contribute around 9 hours of annotation time (which we estimated at 12 HITs) per each primary system submitted, with each HIT including roughly 100 segment translations. Furthermore, we collected information about the classification of their annotators type. Unfortunately, only 65% of the requested annotations were finished by participating teams.

The second annotator group was provided by Toloka AI.[16] Toloka AI is a global data labeling company that helps its customers generate machine learning data at scale by harnessing the wisdom of the crowd from around the world. It relies on a geographically diverse crowd of several million registered users (Pavlichenko et al., 2021).[17] Toloka tests proficiency of their annotator crowd and excludes from future annotations anyone who does not pass quality control in the Appraise tool.

The last part of annotations is sponsored by Microsoft, who contributed with their crowd of qualified paid bilingual speakers experienced in the annotation process. Moreover, Microsoft tracks the performance of the annotators, and those who fail quality control are permanently removed from the pool of annotators. This increases the overall quality of the human assessment.

For bilingual human evaluation, Microsoft contributed with 42%, WMT News participants contributed with 37%, and Toloka platform with 21% of all valid annotations (after removal of annotators that do not pass quality control). The distribution of individual groups of annotators per each language is presented in Table 9.

3.3.2 Document-Level Assessment

This year's human evaluation for out-of-English and non-English language pairs features a document-level direct assessment configuration as presented last year (Barrault et al., 2020). We again use the segment level rating but provide the full document at once (SR+FD, segment rating within a full document), for a more reliable evaluation (Castilho et al., 2020; Laubli et al., 2020).

Figure 5 above shows a screenshot of the fully document-level interface. In the default scenario, an annotator scores individual segments one by one and, after scoring all of them, on the same screen, the annotator then judges the translation of the entire document displayed. Annotators can, however, revisit and update scores of previously assessed segments at any point of the annotation of the given document. It has been shown that presenting the entire document context on a screen may lead to higher quality segment- and document-level assessments (Grundkiewicz et al., 2021) improving the correlation between segment and document scores and increasing inter-annotator agreement for document scores. A similar setup has been used by Popel et al. (2020) even for more than two systems compared at once.

3.3.3 Quality Control

For the document-level evaluation of out-of-English translations, HITs were generated using the same method as described for the SR+DC evaluation of into-English translations in Section 3.2.1 with a minor modification: Since the annotations are made by researchers and professional translators who ensure a better quality of assessments than the crowd-sourced workers, only bad references are used as quality control items.

[16] https://toloka.ai/

[17] https://hackernoon.com/evolution-of-the-data-production-paradigm-in-ai

	Microsoft annotators	Toloka paid crowd	Participants linguists	annotators	researchers	students
English - Chinese	33%	11%	2%	20%	17%	17%
English - Czech	27%	18%	-	54%	-	-
English - German	56%	29%	13%	-	2%	-
English - Hausa	63%	35%	3%	-	-	-
English - Icelandic	82%	5%	13%	-	-	-
English - Japanese	43%	20%	1%	26%	4%	8%
English - Russian	29%	39%	9%	-	23%	-
French - German	76%	14%	11%	-	-	-
German - French	43%	45%	11%	-	-	-
Total	42%	21%	37%			

Table 9: Distribution of annotation crowds for each language pair in bilingual human evaluation. Annotator types are self-classified by participants.

3.3.4 Including Human Translations

Source-based DA allows us to include human references in the evaluation as another system to provide an estimate of human performance. Human references were added to the pool of system outputs prior to sampling documents for tasks generation. Each reference is assessed individually if multiple references are available, which is the case for English→German, English→Czech, English→Russian, and English→Chinese.

3.3.5 Contrastive Direct Assessment

This year we extended the bilingual source-based human evaluation with contrastive evaluation using segment-level pairwise direct assessments (Novikova et al., 2018; Sakaguchi and Van Durme, 2018). It has been pointed out (Freitag et al., 2021a) that standard direct assessment may not be able to properly differentiate high-quality MT system outputs. The contrastive approach to DA can strengthen the discriminative power as annotators judge translations in relation to each other. When standard DA can likely provide better *absolute* quality assessment, the contrastive evaluation can provide better *relative* quality assessments between system pairs. This may help create a more reliable ranking of systems if used on top of the standard approach described in Section 3.3.

The contrastive evaluation is similar to the relative ranking used from WMT08 (Callison-Burch et al., 2008) to WMT16 (Bojar et al., 2016), where annotators were presented with up to five system outputs and corresponding source and reference sentence and asked to rank these systems between each other. The main differences in this year's contrastive evaluation to the relative rankings are that 1) the evaluation is source-based, i.e. without the reference, 2) the continuous scale is used instead of ranks, and 3) only two system outputs are judged at the same time instead of five.

To reduce the cognitive load on annotators, we decided to trial this contrastive approach evaluating individual sentences independent of their context. This is a very important difference compared to the the first stage (Section 3.3).

We ran the contrastive evaluation for English→Chinese, English→Czech and English→German, and we selected top-10 best performing systems based on DA z-score from the ranking created using standard direct assessment for those languages (Table 10), and two human references.

This contrastive evaluation was sponsored by Microsoft and performed by the bilingual paid annotator group as described in Section 3.3.1. Assessments were collected using the open-source Appraise Evaluation Framework (Federmann, 2012). A screenshot of the user interface used in this stage is shown in Figure 6. Each annotator is presented with two randomly selected translated segments from competing systems (anonymized) and asked to rate both of them on a continuous scale of 0-100. Upon request by the annotator, the differences between the two translations were highlighted at the word level to help avoid missing differences. This highlighting may however reduced the effectiveness of control items.

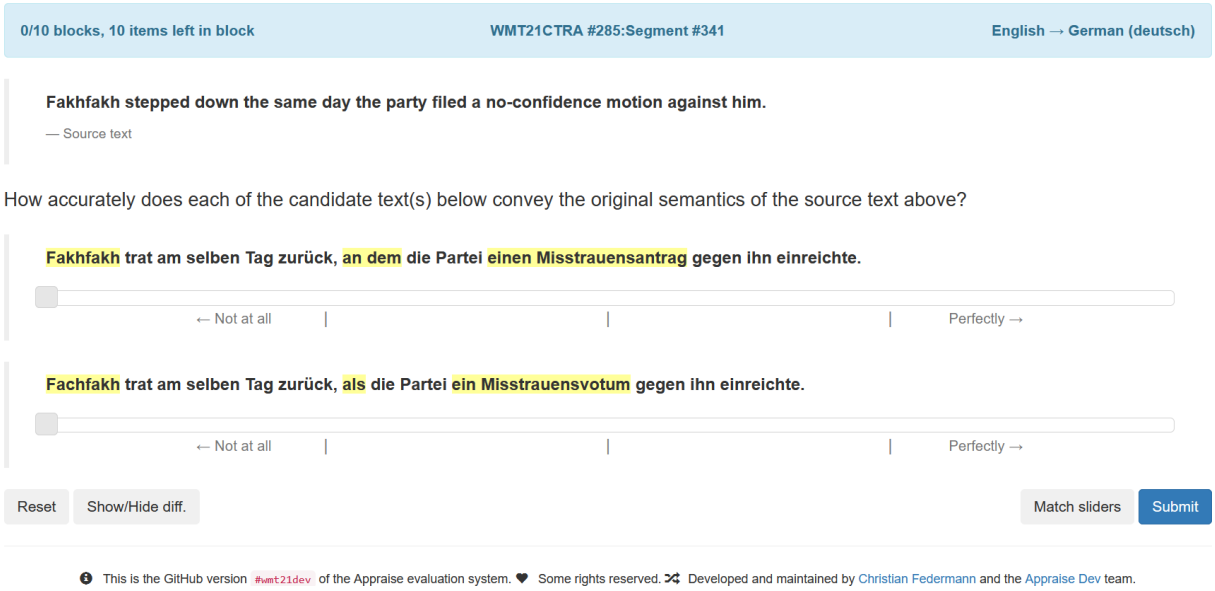


Figure 6: Screen shot of the contrastive DA configuration in the Appraise interface for an example assessment from the 2nd stage of human evaluation campaign. The annotator is presented with two translated segments randomly selected from competing system outputs (anonymized) and is asked to rate both of them on sliding scales.

3.3.6 Human Rankings

Table 10 shows official news task results for translation out-of-English, where lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$.

Source-based DA scores were collected based on the document-level annotation interface, so context was available during annotation. All systems are evaluated in isolation, based on the annotators' perception of translation quality given the source text and document context. Across all language pairs, human reference translations end up in the top-scoring cluster, indicative of a (relatively) high quality of these references. For language pairs with large numbers of submissions, we observe little to no clustering. Notably English→German has only two clusters, one of which contains all but one of the submitted systems, and English→Chinese ends up with a huge mono cluster containing all submissions. While there are differences in average scores and z scores these are not statistically significant enough for effective clustering. As a substitute, rank ranges give an indication of the respective system's translation quality.

Table 11 shows contrastive news task results for translation out-of-English, where lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$.

Contrastive, source-based DA scores (contr:SR−DC) were collected using a segment-level annotation interface, so context was *not* been available to annotators. Results for the source-based DA annotation phase (SR+FD) in Table 11 were computed on the subset of data for the ten systems and two references for which we have run the contrastive, source-based DA annotation phase.

We generally observe better clustering for the contr:SR−DC. This is especially noteworthy as the number of annotations collected per system is much higher for the first, SR+FD, DA phase (for two of the three language pairs on which contr:SR−DC was run). It seems that pairwise comparison of system outputs is beneficial for determining whether differences between systems are statistically significant.

In contrast to the first annotation phase, we find that human reference translations are scored worse, and significantly worse than the top cluster. We explain this by the fact that our contrastive setup was run on segment-level while the source-based DA annotators had access to the full document context. A simple explanation that should nevertheless be empirically validated is that the wording of the sentence created for and within the context of the document does not sound flawless and natural when evaluated in isolation (Läubli et al., 2018a; Toral et al., 2018b). Some machine translation systems do consider the surrounding sentences but their capacity of 'contextualizing' the candidate sentences is probably limited.

Observing the striking difference in system

English→Czech

Rank	Ave.	Ave. z	System
1	90.2	0.397	HUMAN-A
2–4	87.9	0.284	HUMAN-B
2–4	87.6	0.263	Facebook-AI
2–4	86.1	0.214	Online-W
5–7	83.0	0.122	eTranslation
5–6	82.1	0.047	CUNI-Transformer2018
6–8	79.2	-0.120	CUNI-DocTransformer
7–9	79.3	-0.154	CUNI-Marian-Baselines
8–10	77.8	-0.183	Online-B
9–10	74.6	-0.308	Online-A
11	76.2	-0.373	Online-Y
12	65.6	-0.674	Online-G

English→German

Rank	Ave.	Ave. z	System
1–17	83.3	0.266	Online-B
1–5	84.7	0.243	Online-W
1–14	86.6	0.217	WeChat-AI
1–6	87.6	0.145	Facebook-AI
1–10	89.4	0.116	UF
2–17	85.2	0.089	HW-TSC
3–17	86.8	0.072	UEdin
3–18	86.5	0.041	P3AI
3–18	86.4	0.030	HUMAN-A
5–19	83.3	0.013	happypoet
4–19	86.1	0.010	eTranslation
4–19	84.4	0.001	Online-A
3–18	84.5	0.001	HUMAN-C
5–19	78.8	-0.053	VolcTrans-AT
5–19	86.7	-0.055	NVIDIA-NeMo
8–21	83.1	-0.058	Manifold
4–20	84.3	-0.062	Online-G
12–20	84.5	-0.072	Online-Y
18–21	73.9	-0.130	ICL
4–20	85.0	-0.140	VolcTrans-GLAT
16–21	78.3	-0.179	nuclear_trans
22	80.0	-0.415	BUPT_rush

English→Hausa

Rank	Ave.	Ave. z	System
1–2	84.1	0.362	HUMAN-A
1–4	82.7	0.264	Facebook-AI
2–5	80.8	0.263	NiuTrans
3–6	81.2	0.175	Online-B
4–6	80.1	0.128	TRANSSION
2–6	79.2	0.124	ZMT
7–10	78.0	0.018	P3AI
7–10	78.7	0.006	HW-TSC
8–12	75.2	-0.026	AMU
7–10	78.8	-0.036	GTCOM
9–12	75.0	-0.128	MS-EgDC
12–15	70.2	-0.227	UEdin
11-15	73.4	-0.243	Manifold
12–15	70.5	-0.340	TWB
11-15	67.7	-0.448	Online-Y

English→Icelandic

Rank	Ave.	Ave. z	System
1	88.1	0.872	HUMAN-A
2	84.5	0.594	Facebook-AI
3–4	68.2	0.277	NiuTrans
3–4	72.7	0.240	Manifold
5–9	75.2	0.200	Online-A
5–7	65.6	0.130	Lan-Bridge-MT
5–9	62.6	0.063	Mideind
6–9	73.9	0.026	Online-B
6–9	75.6	-0.034	HW-TSC
10	62.0	-0.236	Online-Y
11	48.7	-0.470	Allegro.eu
12	33.9	-1.082	Online-G

English→Japanese

Rank	Ave.	Ave. z	System
1–2	86.4	0.430	Facebook-AI
1–2	85.3	0.314	HUMAN-A
3–5	84.2	0.266	Online-W
3–5	81.3	0.168	WeChat-AI
3–5	82.6	0.148	NiuTrans
6–8	77.8	0.017	HW-TSC
6–8	71.8	-0.042	MiSS
8–13	78.5	-0.051	Online-Y
6–10	77.8	-0.067	BUPT_rush
8–13	70.9	-0.129	Online-A
9–13	67.4	-0.184	Online-B
9–14	74.2	-0.284	ephemeraler
9–14	72.5	-0.339	capitalmarvel
12–14	70.1	-0.373	movelikeajaguar
15–16	63.5	-0.440	Illini
15–16	65.7	-0.541	Online-G

English→Russian

Rank	Ave.	Ave. z	System
1–3	86.0	0.317	HUMAN-B
1–3	83.3	0.277	Online-W
1–3	82.5	0.093	HUMAN-A
4–6	79.4	0.056	Online-B
4–7	75.3	0.032	Online-A
4–7	80.1	-0.001	Facebook-AI
7–10	74.5	-0.123	NiuTrans
7–10	72.3	-0.153	Manifold
7–10	75.4	-0.161	NVIDIA-NeMo
5–10	76.0	-0.180	Online-G
11	62.7	-0.541	Online-Y

English→Chinese

Rank	Ave.	Ave. z	System
1–3	82.5	0.325	HUMAN-B
2–14	74.9	0.284	HappyNewYear
1–7	81.2	0.250	Facebook-AI
1–8	80.0	0.216	HUMAN-A
4–19	75.3	0.164	Borderline
2–19	81.0	0.161	bjtu_nmt
3–14	75.5	0.151	Lan-Bridge-MT
4–21	79.3	0.124	BUPT_rush
2–18	79.2	0.098	NiuTrans
4–18	75.7	0.091	Machine_Translation
2–15	80.9	0.078	SMU
6–22	81.4	0.064	capitalmarvel
4–19	79.5	0.056	WeChat-AI
6–22	78.1	0.026	Online-W
7–22	75.2	0.004	ICL
9–23	75.9	-0.008	HW-TSC
5–23	78.2	-0.025	ZengHuiMT
11–22	81.2	-0.026	yyds
10–26	79.7	-0.050	P3AI
17–27	77.1	-0.061	windfall
6–24	78.9	-0.075	Online-B
13–26	76.8	-0.080	NJUSC_TSC
9–24	77.7	-0.100	MiSS
19–27	77.0	-0.101	UF
22–28	72.7	-0.123	Online-A
22–28	79.3	-0.160	happypoet
20–28	76.9	-0.185	nuclear_trans
25–29	76.4	-0.247	ephemeraler
28–31	67.5	-0.257	Online-G
29–31	67.1	-0.463	Online-Y
29–31	68.3	-0.613	movelikeajaguar

French→German

Rank	Ave.	Ave. z	System
1–5	87.7	0.088	Online-W
1–7	89.2	0.052	Online-A
1–4	89.5	0.035	HUMAN-A
2–8	85.7	0.002	LISN
1–8	86.9	-0.014	Online-B
4–10	85.0	-0.021	talp_upc
3–8	85.0	-0.064	eTranslation
7–10	84.1	-0.154	Online-G
3–10	86.6	-0.210	Online-Y
7–10	86.4	-0.229	P3AI

German→French

Rank	Ave.	Ave. z	System
1–3	87.9	0.160	Online-B
1–3	86.5	0.126	HUMAN-A
3–6	83.4	0.018	Manifold
1–6	84.8	0.006	Online-W
3–6	84.5	0.004	Online-A
6–10	83.0	-0.084	Online-G
3–10	83.5	-0.148	P3AI
6–10	81.3	-0.149	LISN
6–10	83.7	-0.177	Online-Y
6–10	81.0	-0.190	talp_upc

Table 10: Official results of WMT21 News Translation Task for translation out-of-English (SR+FD). Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; rank ranges are based on the same test (for details, see Section 3.2.2); grayed entry indicates resources that fall outside the constraints provided. DA scores are collected using a document-level annotation interface, so context is available to annotators.

ranking by SR+FD vs. contr:SR−DC, esp. the discrepancy in the ranking of human translations, we conclude that evaluating MT systems without document context is no longer reliable for mid- and high-quality MT systems. This is also supported by the surprising observation in Czech→English in Table 7 where humans seemed to be surpassed by *all* participating MT systems. (Considering statistical significance, the claim is arguably weaker: humans share the second cluster with the majority of the systems.) We acknowledge that it is possible that the Czech→English HUMAN-B references are of much worse quality than the English→Czech ones,[18] but we tend to put more trust in the reference quality than in the SR+DC method for two reasons: (1) The annotators did not see the whole document at once and cannot go back in their annotation, so their effective capability to consider context is limited. (2) It is possible that other effects of reference-based DA in the Czech→English start playing role when both the candidate and reference are human vs. when only the reference is human. One possibility would be a stronger confidence of assessors when scoring human translations, leading e.g. to more polarized scores. A detailed investigation into manual evaluation methods that word reliably for both human and machine translations is thus still needed.

3.4 Human Evaluation of Bengali↔Hindi and Xhosa↔Zulu Translation (Wikipedia Data)

Translation quality for Bengali↔Hindi and Xhosa↔Zulu was evaluated using Direct Assessment without considering document context (SR−DC) with a scoring scale of 1-100 by vetted human evaluators. The human evaluators were asked to provide a judgment that they felt most accurately reflected the perceived quality of each corresponding translation of the give source sentence. Definitions of translation quality within several scoring ranges were provided to assist evaluators in providing consistent annotations.

A participating system translation was displayed on the right next to its corresponding source sentence on the left. The sentence pairs were then randomized and passed to a human evaluator for a single direct assessment. The evaluation was performed on the sentence level and evaluators provided a direct assessment score for each sentence-translation pair. The user interface was simpler than the one shown in Figure 5: instead of a slider, the annotators had to enter the scores numerically.

Because evaluators were extremely difficult to recruit for these language pairs and the evaluation was thus low resource, no quality control items were injected and we focused on the vetting process of the evaluators prior to performing any assessment. The only sanity check was that evaluators enter an integer between 1 and 100 as the scores.

All segments from the FLORES Wikimedia test set were included for the evaluation. Each segment was annotated and assessed by one evaluator only once.

All four language directions were assessed by trusted evaluators who have been vetted by a localization vendor specializing in translation evaluation services, to have native fluency of the target language, fluent to native understanding of the source language, have lived in the target region for at least five years recently, and have had at least two to five years of professional translation experience. For Hindi→Bengali and Bengali→Hindi, two human evaluators were used with the translation data being split in half and randomly assigned to the respective evaluators. Two human evaluators assessed for Xhosa→Zulu data and one evaluator assessed for Zulu→Xhosa. The number of evaluators and judgments they made is provided in Table 12.

The final scores for Bengali↔Hindi and Xhosa↔Zulu are provided in Table 13.

3.5 GENIE DE-EN Evaluation

This year, human evaluations for German→English translation with the GENIE leaderboard were also carried out. GENIE is an ongoing effort that centralizes and facilitates human evaluations for natural language generation tasks (Khashabi et al., 2021). In

[18] The quality assurance for each of "A" and "B" references for English↔Czech was comparable; not that the same translators would be producing both directions. In fact, we expected the "B" translations to be *better*, because they were created by experienced students and teachers of translation studies, who are active translators themselves and who *specifically attempted to produce as good translations as possible*. As the to-Czech scores suggest, our annotators preferred the translation agency "A" translations significantly more. But even if the "A" translations were also better than "B" in from-Czech, we see it as very unlikely that the translatologist translations would be worse than all systems.

Source-based DA

(on document level)

SR+FD

English→Czech

Rank	Ave.	Ave. z	System
1	90.2	0.397	HUMAN-A
2-4	87.9	0.284	HUMAN-B
2-4	87.6	0.263	Facebook-AI
2-4	86.1	0.214	Online-W
5-7	83.0	0.122	eTranslation
5-6	82.1	0.047	CUNI-Transformer2018
6-8	79.2	-0.120	CUNI-DocTransformer
7-9	79.3	-0.154	CUNI-Marian-Baselines
8-10	77.8	-0.183	Online-B
9-10	74.6	-0.308	Online-A
11	76.2	-0.373	Online-Y
12	65.6	-0.674	Online-G

Five clusters

English→German

Rank	Ave.	Ave. z	System
1-10	83.3	0.209	Online-B
1-6	84.7	0.179	Online-W
1-10	86.6	0.109	WeChat-AI
1-6	87.6	0.077	Facebook-AI
3-11	86.8	0.008	UEdin
1-11	86.5	-0.014	P3AI
3-11	86.4	-0.031	HUMAN-A
3-11	86.1	-0.038	eTranslation
1-11	84.5	-0.063	HUMAN-C
10-12	84.5	-0.109	Online-Y
5-12	83.3	-0.131	happypoet
3-12	86.7	-0.134	NVIDIA-NeMo

Single cluster

English→Chinese

Rank	Ave.	Ave. z	System
1-8	74.9	0.205	HappyNewYear
1-5	82.5	0.186	HUMAN-B
1-7	81.2	0.139	Facebook-AI
1-5	80.0	0.105	HUMAN-A
3-9	75.5	0.045	Lan-Bridge-MT
2-11	81.0	0.019	bjtu_nmt
2-9	80.9	-0.012	SMU
7-12	75.3	-0.066	Borderline
4-12	75.7	-0.068	Machine_Translation
7-12	81.4	-0.074	capitalmarvel
8-12	79.3	-0.090	BUPT_rush
5-12	79.2	-0.105	NiuTrans

Single cluster

Contrastive, source-based DA

(segment level ignoring doc. context)

contr:SR−DC

English→Czech

Rank	Ave.	Ave. z	System
1-2	87.8	0.281	Facebook-AI
1-2	87.6	0.237	Online-W
3-5	85.6	0.091	CUNI-DocTransformer
3-6	84.9	0.067	CUNI-Transformer2018
4-7	84.3	0.026	HUMAN-A
3-6	84.1	-0.003	HUMAN-B
6-7	83.4	-0.057	eTranslation
8-9	82.7	-0.119	CUNI-Marian-Baselines
8-10	81.3	-0.219	Online-A
9-10	81.1	-0.238	Online-B
11-12	77.7	-0.489	Online-Y
11-12	75.8	-0.630	Online-G

Four clusters

English→German

Rank	Ave.	Ave. z	System
1–3	89.6	0.093	Facebook-AI
1–3	88.5	0.067	WeChat-AI
1–3	88.4	0.035	Online-W
4–9	87.2	-0.044	NVIDIA-NeMo
4–11	87.9	-0.058	HUMAN-C
4–10	86.7	-0.062	P3AI
4–9	86.5	-0.080	UEdin
4–10	87.1	-0.088	Online-B
4–10	86.9	-0.102	eTranslation
6–12	85.7	-0.190	happypoet
10–12	85.7	-0.192	Online-Y
10–12	85.8	-0.226	HUMAN-A

Two clusters

English→Chinese

Rank	Ave.	Ave. z	System
1–5	82.6	0.072	Borderline
1–5	82.3	0.071	bjtu_nmt
1–5	82.5	0.062	SMU
1–5	82.4	0.048	Facebook-AI
1–5	82.5	0.011	NiuTrans
6–11	82.0	-0.016	HappyNewYear
6–11	82.0	-0.016	Machine_Translation
6–10	82.0	-0.056	Lan-Bridge-MT
6–11	81.6	-0.094	BUPT_rush
6–11	81.2	-0.126	capitalmarvel
6–11	81.7	-0.149	HUMAN-A
12	79.3	-0.393	HUMAN-B

Three clusters

Table 11: Contrastive results of WMT21 News Translation Task for translation out-of-English. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; rank ranges are based on the same test (for details, see Section 3.2.2); grayed entry indicates resources that fall outside the constraints provided. DA scores collected using a segment-level annotation interface, so context is not available to annotators.

Language Pair	Sys.	Assess.	Evaluators
Bengali→Hindi	9	4,461	2
Hindi→Bengali	9	4,512	2
Xhosa→Zulu	6	2,952	2
Zulu→Xhosa	5	2,502	1
Total	29	14,437	7

Table 12: Amount of data collected in the WMT21 manual evaluation campaign for evaluation Hindi to/from Bengali and Zulu to/from Xhosa

addition to all German→English submissions, four original transformer baselines with varying sizes and depths were trained and evaluated: GENIE-large-6-6 (transformer large with a 6-layer encoder and a 6-layer decoder), GENIE-base-6-6, GENIE-base-3-3, and GENIE-base-1-1.[19] These models were trained solely on the given training data without ensembling, backtranslation, or any other data augmentation method.

Similar to the official into-English evaluations, evaluations are done monolingually where Human-A is used as the reference. Each HIT contains 5 segments that are randomly shuffled, and no document context is considered during evaluations. Turkers are asked to decide whether they agree or disagree that the prediction adequately expresses the meaning of the reference. Turkers are given the following additional instructions: *a prediction is adequate if in the absence of the reference, the prediction perfectly conveys the meaning intended by the reference*. The user interface for annotating one candidate segment in the HIT is illustrated in Figure 7.

For quality control, we first selected Amazon Mechanical Turkers who had completed at least 5000 HITs with a 99+% approval rate and had a locale of US, GB, AU, or CA. They were then asked to carefully read the instructions and finish 10 sample questions created from WMT 2019 submissions and references. They were allowed to participate only when they correctly annotate 9 instances at least. In addition to this quality control at the entry point, we kept monitoring to detect spamming behavior. In particular, we randomly replaced 5% of the model predictions with sentences identical to the corresponding reference (Perfect Ref., similar to *good reference* in Section 3.2.1), and 5% of the model predictions with the reference from a different question (Wrong Ref.). We then randomly selected 800 examples from the test set to annotate. During annotation, we monitored how annotators labeled the Perfect Ref. and Wrong Ref. questions. Annotators that failed to both assign a high score to the Perfect Ref. and a low score to the Wrong Ref. questions were removed from the annotator pool, and all of their annotations were discarded. This qualification resulted in removing 5% of the participants. Since spammers invest little effort into completing each HIT, they can complete many more than other annotators (we found they would have completed up to 50% of the HITs in our preliminary experiments). Therefore, removing the 5% of participants that spammed annotations substantially improved the quality of our assessment.

In summary, there are several major differences from the setup used in the official evaluations:

- Turkers assess the adequacy by a five-category Likert scale, which is later converted to scalar values: *strongly agree* (1.0), *agree* (0.75), *neutral* (0.5), *disagree* (0.25), and *strongly disagree* (0.0).
- All 5 segments are randomly chosen for each HIT, and the document context is disregarded.
- For evaluating each system, we randomly sample 800 segments from the test set. The randomly selected instances are shared across all systems.
- To maximize the number of segments annotated for a given budget, each segment is annotated only once (*unilabeling*). Under a fixed annotation budget, unilabeling results are shown to be relatively stable compared to *multilabeling* (i.e., evaluating one segment by multiple annotators. See Section 5.1 of Khashabi et al., 2021).
- The overall scores are calculated by averaging raw numbers over the 800 segments. No standardization is applied.
- Different quality controls are applied as discussed above.

Table 14 shows results from the GENIE evaluation for German to English translation. There are systems that are ranked highly, both in the official and GENIE evaluations, such as Online-A and VolcTrans-AT. Conversely, happypoet and Manifold are given low scores consistently. Further, the

[19] The leaderboard is public at https://leaderboard.allenai.org/genie-mt21/submissions/public. All models and code to reproduce are available at https://github.com/jungokasai/GENIE_wmt2021-de-en.

Bengali→Hindi

Rank	Ave.	Ave. z	System
1–2	82.1	0.202	GTCOM
1–2	79.1	0.163	Online-B
3–5	77.5	0.080	TRANSSION
3–5	78.0	0.076	MS-EgDC
3–6	78.0	0.054	UEdin
4–8	76.1	−0.015	Online-Y
6–8	75.7	−0.080	HW-TSC
6–8	75.7	−0.107	Online-A
9	70.8	−0.373	Online-G

Hindi→Bengali

Rank	Ave.	Ave. z	System
1–4	95.0	0.245	HW-TSC
1–4	94.8	0.236	Online-A
1–4	94.5	0.233	GTCOM
1–4	94.6	0.214	UEdin
5–6	92.3	0.080	Online-Y
7	92.0	0.045	TRANSSION
6–7	91.3	0.029	Online-B
8	90.9	−0.008	MS-EgDC
9	73.5	−1.100	Online-G

Xhosa→Zulu

Rank	Ave.	Ave. z	System
1–3	68.4	0.331	HW-TSC
1–3	67.9	0.287	TRANSSION
1–3	63.7	0.240	GTCOM
4–5	61.5	0.144	MS-EgDC
4–5	62.6	0.107	FJDMATH
6	19.4	−1.135	Online-G

Zulu→Xhosa

Rank	Ave.	Ave. z	System
1	80.7	0.502	TRANSSION
2–3	74.3	0.310	HW-TSC
2–4	72.6	0.258	MS-EgDC
3–4	69.3	0.162	GTCOM
5	21.9	−1.253	Online-G

Table 13: Official results of WMT21 Translation Task for Hindi to/from Bengali and Zulu to/from Xhosa translation (Wikipedia data, SR−DC). Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; rank ranges are based on the same test (for details, see Section 3.2.2); grayed entry indicates resources that fall outside the constraints provided.

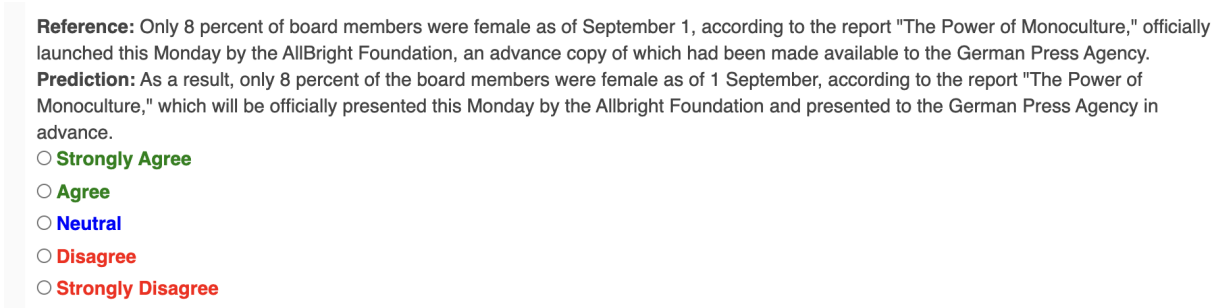


Figure 7: GENIE annotation interface for one segment.

transformer baselines are ranked in the expected order: large-6-6, base-6-6, base-3-3, followed by base-1-1. This confirms the validity of the evaluations. Nonetheless, we see some noticeable difference from the official ranking. In particular, HUMAN and the Watermelon systems are ranked high in contrast to the official evaluations. It is left to future work to analyze which parts of the crowdsourcing setup are contributing to the diverging system rankings; these analyses would help us improve our human evaluation method in the future.

4 Similar Language Translation

In this section we present the findings of the third SLT shared task organized at WMT 2021. The task follows the success of the two past SLT shared tasks organized at WMT 2019 and WMT 2020. SLT 2021 is motivated by the growing interest of the community in translating between similar languages, low-resource languages, dialects, and language varieties, and the challenges faced by state-of-the-art systems in these settings evidenced in recent studies (Hassani, 2017; Costa-jussà et al., 2018; Popović et al., 2020; Tapo et al., 2020).

The main goal of the task is to evaluate the performance of state-of-the-art MT systems on translating between closely-related language pairs of languages from the same language family. Past editions of the task (Barrault et al., 2019, 2020) featured language pairs such as Spanish - Portuguese, Czech - Polish, and Hindi - Nepali to name a few. This year's SLT features multiple pairs of similar languages from the Indo-Aryan and Romance family.

Finally, SLT 2021 also features a track including French and two similar low-resource Manding languages spoken in West Africa, namely Bambara and Maninka, where participants were pro-

GENIE German→English			
Ave. Score	Lower	Upper	System
0.757	0.737	0.776	Watermelon
0.752	0.732	0.772	VolcTrans-AT
0.752	0.732	0.772	HUMAN
0.743	0.724	0.764	Online-B
0.742	0.721	0.760	Online-A
0.740	0.720	0.759	Facebook-AI
0.738	0.721	0.756	Online-W
0.738	0.717	0.757	Online-G
0.737	0.717	0.757	VolcTrans-GLAT
0.735	0.714	0.756	UF
0.734	0.713	0.754	HuaweiTSC
0.733	0.710	0.753	NVIDIA-NeMo
0.712	0.691	0.734	ICL
0.704	0.684	0.723	GENIE-large-6-6
0.704	0.684	0.722	P3AI
0.700	0.680	0.721	UEdin
0.692	0.670	0.712	SMU
0.690	0.669	0.711	GENIE-base-6-6
0.685	0.664	0.705	Manifold
0.676	0.655	0.696	Borderline
0.665	0.645	0.684	Online-Y
0.653	0.630	0.676	GENIE-base-3-3
0.643	0.620	0.667	happypoet
0.507	0.483	0.530	GENIE-base-1-1

Table 14: GENIE DE-EN results. Lower and upper bounds for 95% confidence intervals are calculated by bootstrapping (Koehn, 2004; Khashabi et al., 2021). Grayed entries indicate unconstrained settings.

vided with the opportunity to combine datasets of the two Manding languages taking advantage of their similarity. As in past editions of the task, translations at SLT 2021 are evaluated in all directions using three automatic evaluation metrics: BLEU, RIBES, and TER.

4.1 Data

Training We have made available a number of data sources for the SLT shared task. Some training datasets were used in the previous editions of the WMT News Translation shared task and were updated (News Commentary v16, Wiki Titles v3), while some corpora were newly introduced. We also used data collected from Opus (Tiedemann and Nygaard, 2004; Tiedemann, 2012)[20].

For the Spanish–Catalan language pair we used parallel corpora: Wiki Titles v3, ParaCrawl (Bañón et al., 2020), DOGC v2, and monolingual: Europarl v10 (Koehn, 2005), News Commentary v16, News Crawl, caWaC (Ljubešić and Toral, 2014) (see Table 15). Released corpora for the Spanish–Portuguese language pair included parallel datasets: Europarl v10 (Koehn, 2005), News Commentary v16, Wiki Titles v3, Tilde MODEL (Rozis and Skadiņš, 2017), JRC-Acquis (Steinberger et al., 2006), and monolingual corpora: Europarl v10 (Koehn, 2005), News Commentary v16, News Crawl (see Table 16). Moreover, corpora for the Romanian–Spanish language pair (see Table 17) and the Romanian–Portuguese language pair (see Table 18) contained parallel datasets: Europarl v8 (Koehn, 2005), Wiki Titles v3, Tilde MODEL (Rozis and Skadiņš, 2017), JRC-Acquis (Steinberger et al., 2006), and monolingual data: Europarl v10 (Koehn, 2005), News Commentary v16, News Crawl, Common Crawl.

The released parallel Tamil–Telugu dataset was collected from news (Siripragada et al., 2020), PMIndia (Haddow and Kirefu, 2020) and MKB (Man Ki Baat) datasets. All data were initially combined, tokenized using indic-nlp tokenizer (Kunchukuttan, 2020) and randomly shuffled. A subset of data extracted from the dataset are used for test and development set. The remaining data were considered as training set (cf. Table 21).

Finally, the parallel Bambara-French corpus is a part of the Bambara Reference Corpus [21].

Development and Test Data The development and test sets for Spanish–Catalan, Spanish–Portuguese, Romanian–Spanish and Romanian–Portuguese language pairs were created from a corpus provided by Pangeanic[22]. Catalan translations were provided by the Directorate-General for Language Policy at the Ministry of Culture, Government of Catalonia. Each dev and test dataset was cleaned, deduplicated and shuffled, resulting in 969 and 999 sentences in dev and test sets respectively.

4.2 Participants and Approaches

SEBAMAT SEBAMAT submitted their system for two language pairs, Spanish–Catalan and Spanish–Portuguese, in both directions. The SEBAMAT approach is based on the Marian NMT toolkit that leverages the Transformer architecture. The systems were trained using only the parallel corpora that were made available for the participants. For all the language pairs and directions, SEBAMAT submitted PRIMARY and CONTRASTIVE systems with different vocabulary sizes (40,000 and 85,000, respectively). Interestingly, in all the cases, the PRIMARY systems with a smaller vocabulary size performed better in terms of BLEU scores.

[20] http://opus.nlpl.eu/

[21] http://cormand.huma-num.fr/index.html

[22] https://www.pangeanic.com/

		Corpus	Sentences
Parallel	Spanish ↔ Catalan	Wiki Titles v3	476,475
	Spanish ↔ Catalan	ParaCrawl	6,870,183
	Spanish ↔ Catalan	DOGC v2	10,933,622
Monolingual	Spanish	Europarl v10	2,038,042
	Spanish	News Commentary v16	503,255
	Spanish	News Crawl 2007-2020	65,365,886
	Catalan	caWaC	24,745,986
Dev	Spanish ↔ Catalan		969
Test	Spanish ↔ Catalan		999

Table 15: Corpora for the Spanish ↔ Catalan language pair.

		Corpus	Sentences
Parallel	Spanish ↔ Portuguese	Europarl v10	1,801,845
	Spanish ↔ Portuguese	News Commentary v16	48,259
	Spanish ↔ Portuguese	Wiki Titles v3	649,833
	Spanish ↔ Portuguese	Tilde MODEL	13,464
	Spanish ↔ Portuguese	JRC-Acquis	1,650,126
Monolingual	Spanish	Europarl v10	2,038,042
	Spanish	News Commentary v16	503,255
	Spanish	News Crawl 2007-2020	65,365,886
	Portuguese	Europarl v10	2,016,635
	Portuguese	News Commentary v16	89,111
	Portuguese	News Crawl 2008-2020	10,900,924
Dev	Spanish ↔ Portuguese		969
Test	Spanish ↔ Portuguese		999

Table 16: Corpora for the Spanish ↔ Portuguese language pair.

		Corpus	Sentences
Parallel	Romanian ↔ Spanish	Europarl v8	387,653
	Romanian ↔ Spanish	Wiki Titles v3	253,770
	Romanian ↔ Spanish	Tilde MODEL	3,770
	Romanian ↔ Spanish	JRC-Acquis v2	451,849
Monolingual	Spanish	Europarl v10	2,038,042
	Spanish	News Commentary v16	503,255
	Spanish	News Crawl 2007-2020	65,365,886
	Romanian	Common Crawl	288,806,234
	Romanian	News Crawl 2015-2020	29,538,472
Dev	Romanian ↔ Spanish		969
Test	Romanian ↔ Spanish		999

Table 17: Corpora for the Romanian ↔ Spanish language pair.

T4T The T4T team participated in the SLT 2021 Romance languages track, submitting their system for Spanish ↔ Catalan and Spanish ↔ Portuguese. While their systems are built using out-of-the-box OpenNMT toolkit, the team developed custom cleaning scripts and an adhoc tokenizer. SentencePiece library was used for pre-processing and reducing the vocabulary size to 16,000 symbols.

UBC-NLP The UBC-NLP team submitted their Spanish ↔ Portuguese, Catalan → Spanish and French ↔ Bambara systems to the SLT 2021 task. Their systems are built using Transformers from the HuggingFace library. The UBC-NLP team experimented with tokenized (PRIMARY) and untokenized (CONTRASTIVE) systems and compared them with models developed by fine-tuning pre-trained models as well as models trained from

		Corpus	Sentences
Parallel	Romanian ↔ Portuguese	Europarl v8	381,404
	Romanian ↔ Portuguese	Wiki Titles v3	251,834
	Romanian ↔ Portuguese	Tilde MODEL	3,860
	Romanian ↔ Portuguese	JRC-Acquis v2	451,737
Monolingual	Portuguese	Europarl v10	2,016,635
	Portuguese	News Commentary v16	89,111
	Portuguese	News Crawl 2008-2020	10,900,924
	Romanian	Common Crawl	288,806,234
	Romanian	News Crawl 2015-2020	29,538,472
Dev	Romanian ↔ Portuguese		969
Test	Romanian ↔ Portuguese		999

Table 18: Corpora for the Romanian ↔ Portuguese language pair.

		Corpus	Sentences
Parallel	French ↔ Bambara	Dokotoro/Bible/SIL Dictionary Sentences/Corpus Référence de Bambara	9,939
Dev	French ↔ Bambara		5,972
Test	French ↔ Bambara		2,984

Table 19: Corpora for the French ↔ Bambara language pair.

		Corpus	Sentences
Parallel	French ↔ Maninka	3000 training sentences/Constitution of Guinea	3,243
Dev	French ↔ Maninka		540
Test	French ↔ Maninka		270

Table 20: Corpora for the French ↔ Maninka language pair.

		Corpus	Sentences
Parallel	Tamil ↔ Telugu	MKB	3,100
	Tamil ↔ Telugu	News	11,038
	Tamil ↔ Telugu	PM India	26,009
Dev	Tamil ↔ Telugu		1,261
Test	Tamil ↔ Telugu		1,735

Table 21: Corpora for the Tamil ↔ Telugu language pair.

scratch. The pre-trained models were developed using Marian NMT by Helsinki-NLP on HuggingFace.

A3-108 The A3-108 team submitted 3 systems (one PRIMARY and two CONTRASTIVEs) based on statistical machine translation for Tamil ↔ Telugu language pair. The team explores various tokenization schemes for their submissions. Their PRIMARY run achieved top rank in Telugu → Tamil and ranked 3^{rd} in Tamil → Telugu translation task.

oneNLP oneNLP team participation on Tamil ↔ Telugu system is based on transformer based NMT. The team explored different subword configurations, script conversion and single model training for both directions. Their primary submission achieved 2.05 BLEU for Tamil → Telugu and 5.03 for Telugu → Tamil.

CNLP-NITS The team submitted their run for Tamil ↔ Telugu similar language translation task. The CNLP-NITS system used pre-train word embeddings from monolingual data and applied in transformer based neural machine translation. The model achieved BLEU score 4.05 for both Tamil → Telugu and Telugu → Tamil.

NITK-UOH NITK-UoH's submission system is based on vanilla Transformer model initialized with MultiBPEmb – a collection of multilingual subword segmentation based pretrained embeddings. NITK-UoH performs top in Tamil → Tel-

ugu translation task.

4.3 Results

Similarly to the previous edition of the SLT shared task, participants could submit systems for the Spanish–Catalan and Spanish–Portuguese language pairs (in both directions). The best systems for Spanish-to-Portuguese (see Table 25) achieved over 40 BLEU and around 85 RIBES. While in the opposite direction (Portuguese-to-Spanish) the best performing system reached 47.71 of BLEU (see Table 24). As the Spanish–Catalan dev and test sets were aligned with Spanish–Portuguese ones, we noticed that the best results for the Spanish–Catalan language pair are in general much better than for Spanish–Portuguese. For Spanish-to-Catalan the best system attained over 79 BLEU and below 15 TER (see Table 27). However, its RIBES score (95.76) was lower than the runner-up system's (96.24). In the case of Catalan-to-Spanish, the best system scored over 82 BLEU and less than 11 TER (see Table 26). As there were no submissions for Romanian–Spanish and Romanian–Portuguese, we do not provide any evaluations for these language pairs.

4.4 Summary

This section presented the results and findings of the third edition of the SLT shared task at WMT. The third iteration of this competition featured data from multiple language pairs from three different language families: Dravidian, Manding, and Romance languages. We evaluated the systems translating in both directions of the language pair using three automatic metrics: BLEU, RIBES, and TER. Most teams this year participated in the Dravidian language pairs. Following a trend observed in the past editions of the task, we observed that the performance varies widely between language pairs and domains.

5 Triangular MT

This section presents an overview of the Triangular MT shared task. Given a low-resource language pair (X/Y), the bulk of previous MT work has pursued one of two strategies.

- Direct: Collect parallel X/Y data from the web, and train an X-to-Y translator , OR
- Pivot (Utiyama and Isahara, 2007; Wu and Wang, 2009): Collect parallel X/English and Y/English data (often much larger than X/Y data), train two translators (X-to-English + English-to-Y), and pipeline them to form an X-to-Y translator

However, there are many other possible strategies for combining such resources. These may involve, for example, ensemble methods, multi-source training methods, multi-target training methods, or novel data augmentation methods. For eg. (Zoph et al., 2016; Dholakia and Sarkar, 2014; Kim et al., 2019).

5.1 The Task

The goals of this shared task is to promote:

- translation between non-English languages,
- optimally mixing direct and indirect parallel resources, and
- exploiting noisy, parallel web corpora

The task is Russian-to-Chinese machine translation. We provided parallel corpora to the participating teams. We evaluate system translations on a (secret) mixed-genre test set, drawn from the web and curated for high quality segment pairs. After receiving test data, participants had one week to submit translations. After all submissions are received, we posted a populated leaderboard that will continue to receive post-evaluation submissions.[23] The evaluation metric for the shared task is 4-gram character Bleu. The script to be used for Bleu computation is Moses `multi-bleu-detok.perl`. Instructions to run the script were released as part of the shared task.[24] The participants indicated their intent to participate via registration on the Codalab website for the shared task[25] and obtained the instructions and links to various resources.

5.2 Training Data

We provided three parallel corpora:

- Chinese/Russian: crawled from the web and aligned at the segment level, and combined with different public resources.

[23] https://competitions.codalab.org/competitions/30446#results

[24] https://github.com/didi/wmt2021_triangular_mt/tree/master/eval

[25] https://competitions.codalab.org/competitions/30446#participate

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
NITK-UOH	PRIMARY	6.09	17.03	-
A3-108	CONTRASTIVE1	5.54	40.58	98.082
A3-108	PRIMARY	5.23	42.37	98.662
CNLP-NITS	PRIMARY	4.05	24.80	97.241
oneNLP	CONTRASTIVE2	3.67	22.28	99.122
oneNLP	CONTRASTIVE	3.57	23.54	99.034
A3-108	CONTRASTIVE2	3.32	34.42	-
oneNLP	PRIMARY	2.05	21.68	-
NITK-UOH	CONTRASTIVE	0.00	0.03	-

Table 22: Evaluation results for Tamil to Telugu.

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
A3-108	PRIMARY	8.37	43.55	95.884
A3-108	CONTRASTIVE1	7.89	46.24	95.627
A3-108	CONTRASTIVE2	7.43	42.54	94.964
NITK-UOH	PRIMARY	6.55	19.61	98.356
oneNLP	PRIMARY	5.03	23.98	97.551
CNLP-NITS	PRIMARY	4.05	24.80	97.241
oneNLP	CONTRASTIVE	3.63	27.05	97.534
oneNLP	CONTRASTIVE2	3.61	26.12	96.772
NITK-UOH	CONTRASTIVE	0.04	1.00	-

Table 23: Evaluation results for Telugu to Tamil.

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
UBC-NLP	PRIMARY	47.71	87.11	39.213
SEBAMAT	PRIMARY	46.51	86.31	41.235
T4T	PRIMARY	46.29	87.04	40.181
UBC-NLP	CONTRASTIVE	43.86	85.10	43.801
SEBAMAT	CONTRASTIVE	43.12	84.99	45.068

Table 24: Evaluation results for Portuguese to Spanish.

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
T4T	PRIMARY	40.74	85.69	43.343
SEBAMAT	PRIMARY	40.35	84.99	45.258
SEBAMAT	CONTRASTIVE	38.90	83.89	47.044
UBC-NLP	PRIMARY	38.10	85.35	46.556
UBC-NLP	CONTRASTIVE	35.61	82.48	52.612

Table 25: Evaluation results for Spanish to Portuguese.

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
UBC-NLP	PRIMARY	82.79	96.98	10.918
SEBAMAT	PRIMARY	78.65	94.76	15.805
T4T	PRIMARY	77.93	96.04	16.502
UBC-NLP	CONTRASTIVE	76.8	95.19	15.421
SEBAMAT	CONTRASTIVE	76.78	94.46	17.067

Table 26: Evaluation results for Catalan to Spanish.

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
SEBAMAT	PRIMARY	79.69	95.76	14.632
T4T	PRIMARY	78.60	96.24	16.133
SEBAMAT	CONTRASTIVE	77.32	95.35	16.744

Table 27: Evaluation results for Spanish to Catalan.

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
UBC-NLP	PRIMARY	1.32	24.79	97.899

Table 28: Evaluation results for French to Bambara.

- Chinese/English: combining several public resources.
- Russian/English: combining several public resources.

The details of the training resources provided are shown in Table 30. The provenance of the collected parallel data is as follows. We used a parallel data harvesting pipeline developed at DiDi (Zhang et al., 2020) to harvest Russian/Chinese parallel data on the Internet. We downloaded parallel datasets available from Opus (Tiedemann, 2009) for all the three language pairs - Russian/Chinese, Russian/English and English/Chinese. Since united nations data and subtitles data (Ru/En) are very large sources of parallel data, we report statistics on these two types of Opus parallel sources. In addition to Opus, we also curate parallel data from Wikimatrix (Schwenk et al., 2019) in all three language pairs and social media parallel data - Weibo and Twitter (Ling et al., 2013). We also release the provenance of each parallel segment, in case teams want to use this information to filter noisy data sources.

5.3 Creating the Test Dataset

We spent a considerable amount of time to curate high quality, parallel data online to be used as development and evaluation datasets. This was a completely manual process undertaken by a native speaker of Russian who consulted with a native Chinese speaker from our team to ensure good quality translations (that does not contain tell-tale signs of automatic translation). Our workflow entailed finding websites and large chunks of parallel text, not necessarily from the same pages. The sources selected were also hard to be harvested from a parallel data pipeline due to their difference in URL structure. The sources selected were from a diverse range of non-traditional sources, and have a balance of different types of documents. The topics would be famous works of literature, or tourism related news stories, and so on. We copied large chunks of text from such sources and manually aligned the paragraphs, followed by manual sentence alignment, each done manually to ensure top quality parallel segments. This was followed by a final filtering step to remove sentences and entire sources which had a significant overlap with training and development data. The details of the development and test datasets are shown in Tables 31 and 32.

5.4 Baselines and Final Results

We released a baseline system[26] as part of the shared task. This is based on the Google Tensor2tensor[27] toolkit to train a Transformer-based NMT system. We also provided the baseline bleu score on the development dataset ahead of the evaluation phase. We had 2 simple baselines - (1) Direct - Transformer model trained on the entire Russian/Chinese parallel dataset and decoded with $\alpha = 1.0$ and $beam\_size$=4. (2) Pivot model - 2 MT systems - Russian-to-English and English-to-Chinese - each trained with the corresponding parallel data. Both the Russian-to-English and the English-to-Chinese systems were decoded with $alpha$=1.0 and $beam\_size$=4. The baseline results on the development dataset as shown in Table 33.

We had a total of six teams submitting their system outputs on the test dataset. The evaluation metric was 4-gram character bleu score. The final evaluation results are shown in Table 34.

[26] https://github.com/didi/wmt2021_triangular_mt/

[27] https://github.com/tensorflow/tensor2tensor

Team Name	System Type	BLEU ↑	RIBES ↑	TER ↓
UBC-NLP	PRIMARY	3.62	36.17	-

Table 29: Evaluation results for Bambara to French.

Russian/Chinese parallel data	**Segment pairs**	**Characters (Chinese side)**
DiDi parallel data harvesting pipeline	5,403,157	82,552,922
Opus (no UN) + Weibo + Wikimatrix	430,302	20,954,541
Opus (UN)	27,551,996	1,362,478,536
Total	*33,385,455*	*1,465,985,999*
Russian/English parallel data	**Segment pairs**	**Words (Russian side)**
Opus (no UN, no subtitles) + Twitter + Wikimatrix	6,340,245	97,537,275
Opus (UN, subtitles)	62,811,986	909,476,736
Total	*69,152,231*	*1,007,014,011*
English/Chinese parallel data	**Segment pairs**	**Characters (Chinese side)**
Opus (no UN) + Twitter + Weibo + Wikimatrix	1,435,132	69,894,886
Opus (UN)	27,089,931	1,333,732,823
Total	*28,525,063*	*1,403,627,709*

Table 30: Triangular MT: Training data statistics

5.5 Overview of the Submitted Systems

Five out of the six participating systems submitted system description papers. In this section we briefly discuss the outline of these systems. For more details please refer to the proceedings.

- istic-team-2021 (Guo et al., 2021) The team's system is based on the Transformer architecture. They used several corpus pre-processing steps such as special symbol filtering and filtering based on segment length. In addition, they used context-based system combination - which is a multi-encoder to encode source sentence and contextual information from the machine translation results on the source sentence. They tried with both a direct and pipeline-based pivot system and report that the latter outperforms the former.

- HW_TSC (Li et al., 2021a) Huawei's submission used a multilingual model which is a single neural machine translation model to translate among multiple languages. Upon adding more parallel data, they report an increase in bleu score of upto 2 points using the multilingual model compared to the baseline model. In addition they used several data pre-processing techniques to denoise the training data and data augmentation techniques such as back-translation to improve overall system performance.

- Papago (Park et al., 2021) Naver's system reports that they get better performance by treating this as a bilingual machine translation task rather than as a multilingual translation task, based on their early experiments. They use the transformer model with extensive data pre-processing, filtering and data augmentation. To augment the direct bilingual data they synthetically generate bilingual sentence pairs using monlingual Chinese back-translated to Russian and the 2 sets of indirect parallel dataset provided.

- DUT-MT (Liu et al., 2021a) This team experimented with 2 different multilingual training models called mBART and mRASP, both of them based on underlying Transformer architecture. They report boosted performance especially on rare words when using mRASP. In addition, they also carry out data preprocessing and filtering to improve system performance.

- CFILT-IITB (Mhaskar and Bhattacharyya, 2021) CFLIT-IITB team's system used a pivot-based transfer learning technique. In this technique they have 2 encoder-decoder models, source-pivot (Russian-to-English) and pivot-target (English-to-Chinese), each of them trained on the respective training datasets. They use the encoder of the former and the decoder of the latter to initialize a third encoder-decoder for the actual task of Russian-to-Chinese translation. They fine tune this decoder using the given parallel data for Russian/Chinese. They report this system has a better performance compared to either a direct or pivot-based cascaded system. They do not experiment much with data pre-processing and filtering.

Source	Genre	Parallel segments
Anna Karenina, dialog	Literature	98
Art Academy	Biography	67
Isaac Babel interview	Literature	104
Master and Margarita	Literature	106
MPMCMS	International news	71
Potato system	International news	97
Visit Amur	Tourism	250
Chinese Embassy in Russia	International news	172
Total	-	*965*

Table 31: Triangular MT: Development dataset details

Source	Genre	Parallel segments
Aeroflot	Tourism	99
Isaac Babel - salt	Literature	47
A Day Without Lies	Literature	200
Everything is Normal, Everything is Fine	Literature	98
Hujiang	Language Learning	236
Kazinform	Tourism	21
Lotos shopping centre	Tourism	17
Alexandra Marinina novel	Literature	55
Private Museum Catalog	Tourism	196
Solzhenitsyn Nobel speech	Literature	240
Russia Beyond	Biography	329
Shenyang consulate	International news	113
War and Peace	Literature	3
Russian Embassy in China	Tourism, International News	97
Total	-	*1751*

Table 32: Triangular MT: Test dataset details

5.6 Conclusion

The triangular machine translation shared task set out to explore various modeling possibilities when building a machine translation system for a non-English language pair. We received enthusiastic participation from the participants. Almost all of them performed data filtering and preprocessing to denoise the training datasets and that seemed to substantially help improve system performance. The transformer model and its variants were used in all the system submissions confirming Transformer's ubiquitous acceptance as the model of choice for building machine translation systems. Many teams explored model ensembling and model averaging in addition to model reranking strategies. Several teams explored back-translation as an effective data-augmentation strategy. There was a wide variety of modeling architectures experimented by the participants. Almost everyone used all the parallel datasets provided underlining the importance of using parallel data in all directions to build a better machine translation system. Overall we are happy that the shared task provided a platform to the participants to experiment with different modeling strategies. We hope practitioners will find these techniques useful when working on machine translation between non-English language pairs.

6 Multilingual Low-Resource Translation for Indo-European Languages Task

Massively multilingual machine translation has shown impressive results, including zero and few-shot translation of low-resource languages. However, these models are often evaluated from or into English, where the most data is available, and one assumes that the models would generalise to other language pairs and low-resource languages. This shared task focuses explicitly on checking this assumption and aims to explore multilingual archi-

System	BLEU
Google Translate API	33.04
BASELINE-DIRECT	20.24
BASELINE-PIVOT	19.33

Table 33: Triangular MT: Baseline results on the development dataset

	Team name	BLEU
	Google Translate API	30.2
Team 1	HW_TSC	27.7
Team 2	Papago	26.8
Team 3	DUT-MT	21.7
Team 4	istic-team-2021	19.2
Team 5	CFILT-IITB	18.8
-	BASELINE-PIVOT	17.9
-	BASELINE-DIRECT	17.0
Team 6	mcairt	16.6

Table 34: Triangular MT: Results on the test dataset

tectures for languages in a same family and evaluate only low-resource pairs even if using the high-resourced pairs in the same language family is not forbidden. We work in the cultural heritage domain, where we can consider full documents, and in two Indo-European language families: North-Germanic and Romance. With these goals in mind (multilinguality, specific domain and document-level translation) we define two tasks, one per family:

Task 1. Europeana thesis abstracts and descriptions. North-Germanic languages: from/to Icelandic (is), Norwegian Bokmål (nb) and Swedish (sv). Danish (da), German (de) and English (en) data is allowed for training but translation quality is not evaluated.

Task 2. Wikipedia cultural heritage articles. Romance languages: from Catalan (ca) to Occitan (oc), Romanian (ro) and Italian (it). Spanish (es), French (fr) and Portuguese (pt) data (+ English) is allowed for training but translation quality is not evaluated.

6.1 Data and Resources

6.1.1 Training Corpora

One of the purposes of the shared task is to obtain state-of-the-art systems for the language pairs in the domain involved. In principle, this would imply an unconstrained data setting but, we also want to be able to compare systems and architectures among themselves. For this, we constrain the amount of parallel and monolingual corpora to be used but we allow pretrained open-source systems which might use more data than allowed for the languages considered. All the sources listed below apply to the following languages (except for pretrained models): Icelandic, Norwegian Bokmål, Swedish, Danish, German and English (Task 1); and Catalan, Italian, Occitan, Romanian, Spanish, French, Portuguese and English (Task 2).

- Corpora available at ELRC.[28] This data includes Paracrawl and Global voices.
- Europarl, JW300, WikiMatrix, MultiCCAligned, OPUS-100, Books, the Bible and TED talks.
- Common Crawl, Wikipedia and Wikidata dumps.
- Wordnets with open license, BabelNet.
- (Multiligual) pre-trained embeddings or other models that can be found freely available online (Hugging Face).
- Additional resources in Section 6.1.2 (multilingual lexicons).

6.1.2 Additional Resources

Given the importance of named entities in the cultural heritage domain, we provide participants with parallel/multilingual lexicons from Wikidata, Wikipedia titles and Wiktionary. The figures for each source are summarised in Table 35.

[28] https://elrc-share.eu/repository/search/

	Wikidata		Wikipedia		Wiktionary
	all	cleaner	all	cleaner	all
is2nb/nb2is	1,141,891	–	–	–	3,304/6,552
is2sv/sv2is	1,149,894	–	–	–	15,369/17,321
nb2sv/sv2nb	2,648,493	–	–	–	9,390/7,124
is-nb-sv	1,139,493	23,574	–	–	–
ca2it/it2ca	3,072,380	–	323,055	–	18,684/19,050
ca2oc/oc2ca	1,300,979	–	71,854	–	3,999/3,538
ca2ro/ro2ca	1,608,860	–	123,215	–	11,990/12,034
it2oc/oc2it	1,285,771	–	75,542	–	7,225/6,332
it2ro/ro2it	4,547,649	–	215,296	–	20,898/20,442
ro2oc/oc2ro	1,230,752	–	64,800	–	4,586/4,350
ca-it-ro	1,579,345	123,543	117,543	97,484	–

Table 35: Number of entries of the parallel/multilingual lexicons extracted from Wikidata, Wikipedia titles and Wiktionary for the multilingual low-resource translation task.

	Validation				Test			
	Docs.	Sents.	Src toks.	Tgt toks.	Docs.	Sents.	Src toks.	Tgt toks.
is2nb	26	467	6,096	6,932	24	563	8,256	9,301
is2sv	26	467	6,096	6,611	24	563	8,256	8,819
nb2is	19	502	7,673	7,495	16	540	9,218	8,867
nb2sv	19	502	7,673	7,499	16	540	9,218	8,804
sv2is	43	516	9,097	9,524	44	547	9,642	9,733
sv2nb	43	516	9,097	9,232	44	547	9,642	9,787
ca2it	41	1,269	30,363	29,725	42	1,743	38,868	37,649
ca2oc	41	1,269	30,363	30,184	42	1,743	38,868	38,662
ca2ro	41	1,269	30,363	29,842	42	1,743	38,868	37,379

Table 36: Statistics on the validation and test sets of the multilingual low-resource translation task. Source (Src) are original documents and target (Tgt) are human translations.

Wikidata. We extract aligned lexicons from the wikidata-20210301-all.json dump and provide two versions. The complete ("all") version includes all the entries, including duplicates. The "cleaner" version excludes duplicates, most of the terms that are equal in all the languages, terminology related to Wikimedia and a naïve cleaning on terms including years, parenthesis, and others.

Wikipedia titles. We extract aligned titles for the languages in Task 2 from the May 2020 Wikipedia dumps using the Wikitailor Toolkit[29] (Barrón-Cedeño et al., 2015; España-Bonet et al., 2020). We also provide two versions: the complete version ("all") includes all the entries. The "cleaner" version results from a naïve cleaning on titles including years, dates, parenthesis, and others.

Wiktionary. Each Wiktionary entry contains a word, its translation into several languages and its part of speech. We extract bilingual entries from April 2021 dumps for adjectives, adverbs, nouns and verbs from the Icelandic, Swedish, English and German Wiktionaries (Task 1) and from the Catalan and English ones (Task 2). The part of speech is kept in the dictionaries. Since the xlm dump contains the information in a text element with different structure for different dictionaries, we provide the extraction scripts for reproducibility.[30]

6.1.3 Validation and Test Sets

The documents used for constructing the validation and test sets are obtained from the Europeana collection (Task 1) and Wikipedia (Task 2).

Europeana kindly provided us with thesis abstracts, descriptions of archaeological sites and bibliographic entries for Icelandic, Norwegian Bokmål and Swedish. These monolingual documents are available at the Europeana portal but no intra-family parallel data exists and even the monolingual extraction is not straightforward for two main reasons: (i) collections with pan-Scandinavian labels and descriptions are uncommon, and (ii) language attributes in general are uncommon. For documents tagged as Norwe-

[29] github.com/cristinae/WikiTailor

[30] github.com/LeHarter/Extracting-translations-from-wiktionary

gian there is no distinction between Bokmål and Nynorsk, so texts where classified according to simple heuristics based on lexicons.

The original Europeana crawl obtained 1,192 documents (150,080 tokens) for Icelandic, 2,000 documents (166,303 tokens) for Norwegian Bokmål and 2,046 bilingual documents in English and Swedish with 443,111 tokens for Swedish. From these sets, we eliminate very similar documents (specially for Icelandic) and split documents at sentence level manually; we selected documents to collect around 1,000 sentences per language. Documents are finally divided evenly to build a validation set and a test set (Table 36).

The Wikipedia sets were built from articles in the Catalan edition. We selected original articles in Catalan that have no comparable article in any other language and that cover the cultural heritage domain (food, locations, sport, literature, traditions, people and animals). We selected 83 articles which were sentence-split manually to gather 3,013 sentences and 69,231 tokens. Similarly to the North-Germanic family, documents are divided evenly to build a validation set and a test set (Table 36). In this case, we also marked some entities in the source test documents (dates and locations) for further analysis in the manual evaluation (see Section 6.4).

Validation and test sets were sent to professional translators. A first translation was done by a native professional translator and afterwards there was a quality evaluation check by a second native professional translator. For the North-Germanic languages, we translated the source texts in Icelandic, Norwegian Bokmål and Swedish into the other two languages. For the Romance languages, we translated the source texts in Catalan into Italian, Romanian and Occitan. Translators were asked to keep the same sentence division as in the source and no indications were given on the translation of named entities.

6.2 Baselines and Submitted Systems

Nine different teams downloaded the validation data set but only five of them participated: BSC, CUNI, EdinSaar, Tencent and UBCNLP. We allowed two submissions per group and task, a primary (P) and a contrastive (C) system. With these constraints, we received four submissions for Task 1 and seven submissions for Task 2. We also prepared two baseline systems for comparison purposes.

6.2.1 M2M-100 (baseline)

We use M2M-100 without any modification, a multilingual model trained on a data set with 7.5 billion sentences for 100 languages including all the languages in our task (Fan et al., 2020). The sequence-to-sequence system is trained with parallel data enriched with backtranslations. We use the model with 1.2 B parameters available at the Hugging Face site.[31]

6.2.2 mT5-devFinetuned (baseline)

mT5 is a sequence-to-sequence model pretrained on a masked language modeling span-corruption objective with 8.5 billion monolingual sentences from 101 languages (Xue et al., 2021). As baseline, we use the model with 580 M parameters from Hugging Face. We finetune mT5-base only with the multilingual validation sets for each task described in Section 6.1.3. For Task 1, that involves 5,500 sentences, where we use the parallel sentences L_1–$L_{2\text{dev}}$ in both directions $L_1 2 L_2$ and $L_2 2 L_1$ (that is, we use $\text{is2nb}_{\text{dev}}$ sentences as is2nb and nb2is, and $\text{nb2is}_{\text{dev}}$ sentences as nb2is and is2nb because $\text{is2nb}_{\text{dev}}$ and $\text{nb2is}_{\text{dev}}$ are different; the same for the other pairs). We prepend one of the extra_id tokens in mT5 vocabulary to the source sentences to indicate the language of the target sentences. The remaining 440 sentences are used for validation. We repeat the process for Task 2, but in this case the training is multilingual but not bidirectional, so sentences are only used in one direction with a total of 3,600 sentences (1,200 ca2it, 1,200 ca2ro and 1,200 ca2oc) for finetuning and 207 for validation.

6.2.3 BSC (Kharitonova et al., 2021) – Task 2

BSC submission is a multilingual semi-supervised machine translation model. It is based on a pretrained language model, XLM-RoBERTa, that is later finetuned with parallel data obtained mostly from OPUS (5.1 M sentences). XLM-RoBERTa is only used to initialize the encoder while the shallow decoder is randomly initialised.

6.2.4 CUNI (Jon et al., 2021) – Task 2

Multilingual supervised machine translation model (primary) enriched with backtranslated data (contrastive). The multilingual systems

[31] https://huggingface.co/facebook/m2m100_1.2B

	Average Ranking	BLEU	TER	chrF	COMET	BertScore
M2M-100 (baseline)	1.0±0.0	31.5	0.54	0.55	0.399	0.862
EdinSaar-Contrastive	2.2±0.4	27.1	0.57	0.54	0.283	0.856
EdinSaar-Primary	2.8±0.4	27.5	0.58	0.52	0.276	0.849
UBCNLP-Primary	4.0±0.0	24.9	0.60	0.50	0.076	0.847
UBCNLP-Contrastive	5.0±0.0	24.0	0.61	0.49	-0.068	0.837
mT5-devFinetuned (baseline)	6.0±0.0	18.5	0.78	0.42	-0.102	0.810

Table 37: Official ranking according to the automatic metric average for the multilingual low-resource translation task of Europeana documents for North-Germanic languages (Task 1).

	Average Ranking	BLEU	TER	chrF	COMET	BertScore
CUNI-Primary	1.2±0.4	50.1	0.401	0.694	0.566	0.901
CUNI-Contrastive	1.6±0.5	49.5	0.404	0.693	0.569	0.901
TenTrans-Contrastive	3.0±0.0	43.5	0.460	0.670	0.444	0.894
TenTrans-Primary	3.8±0.4	43.3	0.462	0.668	0.442	0.894
BSC-Primary	5.0±0.7	41.3	0.402	0.647	0.363	0.884
M2M-100 (baseline)	5.8±0.4	40.0	0.478	0.634	0.414	0.878
UBCNLP-Primary	7.2±0.4	35.4	0.528	0.588	0.007	0.854
mT5-devFinetuned (baseline)	8.0±0.7	29.3	0.592	0.553	0.059	0.850
UBCNLP-Contrastive	8.6±0.5	28.5	0.591	0.529	-0.374	0.825

Table 38: Official ranking according to the automatic metric average for the multilingual low-resource translation task of Wikipedia articles in the cultural heritage domain for Romance languages (Task 2).

use 41 M original parallel sentences including all language pairs in the task plus French and English. Besides leveraging multilingual training data, various subword granularities are explored and phonemic representation of texts are added via multi-task learning. For Catalan–Occitan, character-level rescoring on the translations n-best lists is applied and Apertium is used for backtranslations when included.

6.2.5 EdinSaar (Tchistiakova et al., 2021) – Task 1

Semi-supervised systems with multilingual pretraining, backtranslation, finetuning and checkpoint ensembling. The primary system is a semi-supervised machine translation model. mT5 is finetuned with 1.2 M parallel sentences in the languages of the task plus Danish, German and English. The contrastive system is a transformer base architecture trained with 422 M parallel sentence pairs in all 30 language directions (including Danish, German and English) and finetuned only with pairs with the languages of the task as target language.

6.2.6 TenTrans (Yang et al., 2021) – Task 2

TenTrans submissions are semi-supervised multilingual systems based on a transformer base architecture. The basic system is an 8-to-4 multilingual model with Catalan–Italian–Romanian–Occitan as the target side and the inclusion of the high resource languages Spanish, French, Portuguese and English on the source side. In-domain finetuning is done with data selected using a domain classifier trained with multilingual BERT. Knowledge transfer is achieved with knowledge distillation of the M2M 1.2B model previously finetuned on the languages of the task. The primary submission is an ensemble between the in-domain multilingual and the distilled M2M. The contrastive submission adds a multilingual base model enriched with backtranslations to the ensemble and pivot-based methods to augment the training corpus.

6.2.7 UBCNLP (Chen and Abdul-Mageed, 2021) – Task 1, Task 2

Supervised bilingual systems based on a transformer base architecture where the Helsinki-NLP pretrained models available at the Hugging Face site are finetuned to the languages of the shared task. The primary submission finetunes the

	sv2nb					is2nb				
	BLEU	**TER**	**chrF**	**COMET**	**BertSc**	**BLEU**	**TER**	**chrF**	**COMET**	**BertSc**
M2M-100	**56.8**	**0.29**	**0.77**	**1.048**	**0.935**	19.3	0.67	0.42	-0.133	0.825
mT5-dFT	36.3	0.46	0.63	0.716	0.891	**22.3**	**0.64**	**0.47**	**0.120**	**0.853**
EdinSaar-C	48.2	0.35	0.73	0.980	0.923	13.0	0.71	0.41	-0.250	0.820
EdinSaar-P	45.4	0.38	0.70	0.919	0.912	16.3	0.72	0.39	-0.287	0.812
UBCNLP-C	51.8	0.33	0.74	0.996	0.931	9.5	0.77	0.33	-0.827	0.778
UBCNLP-P	49.8	0.35	0.73	0.952	0.927	12.8	0.74	0.36	-0.628	0.799

	nb2is					sv2is				
	BLEU	**TER**	**chrF**	**COMET**	**BertSc**	**BLEU**	**TER**	**chrF**	**COMET**	**BertSc**
M2M-100	**21.5**	**0.64**	**0.47**	**0.259**	**0.833**	19.0	0.66	0.48	0.501	0.832
mT5-dFT	3.6	1.26	0.21	-0.986	0.705	9.4	0.82	0.35	-0.138	0.777
EdinSaar-C	18.3	0.66	0.46	0.155	0.829	20.2	0.65	0.50	0.469	0.836
EdinSaar-P	19.5	0.65	0.46	0.258	0.829	**22.4**	**0.64**	**0.51**	**0.509**	0.836
UBCNLP-C	7.8	0.78	0.32	-0.924	0.771	20.5	0.66	0.49	0.348	**0.838**
UBCNLP-P	15.7	0.68	0.43	-0.074	0.822	14.8	0.71	0.45	0.144	0.825

	nb2sv					is2sv				
	BLEU	**TER**	**chrF**	**COMET**	**BertSc**	**BLEU**	**TER**	**chrF**	**COMET**	**BertSc**
M2M-100	**50.9**	**0.34**	**0.72**	**0.826**	**0.921**	**21.2**	**0.63**	0.45	-0.110	0.826
mT5-dFT	18.6	0.82	0.40	-0.368	0.790	21.1	0.69	**0.46**	**0.047**	**0.844**
EdinSaar-C	45.4	0.37	0.69	0.690	0.911	17.3	0.66	0.42	-0.348	0.815
EdinSaar-P	42.9	0.40	0.65	0.615	0.898	18.8	0.68	0.41	-0.357	0.805
UBCNLP-C	36.8	0.43	0.63	0.422	0.893	17.6	0.69	0.40	-0.425	0.810
UBCNLP-P	42.7	0.39	0.67	0.636	0.906	14.0	0.70	0.38	-0.572	0.804

Table 39: Automatic evaluation per language pair in the North-Germanic family of the multilingual low-resource translation task (Task 1). Best scores boldfaced. Notice that the final ranking is done per family and not per language pair as shown in Table 37.

Catalan–Spanish Helsinki-NLP model with WikiMatrix data (1.1 M sentences for ca-it, 139 k for ca-oc and 490 k for ca-ro). The same data is used to finetune the Catalan–English Helsinki-NLP model in the contrastive submission.

6.3 Automatic Evaluation

Recently, automatic metrics based on contextual embeddings have been shown to correlate better than string matching ones with human judgments (Kocmi et al., 2021). COMET was shown to be the best performing metric for languages with Latin script and chrF the best performing string-based method. Still, BLEU is used as *de facto* metric in most papers. As we cannot perform human evaluation for the 9 language pairs involved in this shared task, for the official ranking we use a combination of several metrics including the ones just mentioned plus BertScore as representative of contextual embedding-based metrics and TER as representative of plain string methods.

We evaluate the submissions and the baseline systems for the two tasks using BLEU,[32] TER,[33] chrF,[34] (all with SacreBLEU) COMET,[35] and BertScore.[36] The final ranking is done according to the average ranking of the individual metrics per family, ties on individual metrics are considered.

We report the results for Task 1 in Table 37 and for Task 2 in Table 38. M2M-100 resulted in a very strong baseline for North-Germanic languages. EdinSaar systems are second and third, followed by UBCNLPs. The ranking is consistent

[32] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a++version.1.4.14

[33] TER+tok.tercom-nonorm-punct-noasian-uncased+version.1.4.14

[34] chrF2+numchars.6+space.false+version.1.4.14

[35] wmt-large-da-estimator-1719 model(comet=0.1.0)

[36] bert-base-multilingual-cased_L9_no-idf_version=0.3.9(hug_trans=4.9.0.dev0)

	ca2it					ca2oc				
	BLEU	**TER**	**chrF**	**COMET**	**BertSc**	**BLEU**	**TER**	**chrF**	**COMET**	**BertSc**
M2M-100	46.6	0.390	0.694	0.743	0.913	40.2	0.405	0.673	0.341	0.892
mT5-dFT	30.4	0.551	0.571	0.235	0.872	40.1	0.395	0.680	0.402	0.897
BSC-P	42.0	0.420	0.670	0.651	0.908	57.1	0.272	0.780	0.514	0.929
CUNI-C	49.5	0.366	0.714	**0.813**	0.916	**67.1**	**0.201**	**0.832**	**0.724**	**0.952**
CUNI-P	**50.5**	**0.360**	**0.717**	0.810	**0.917**	66.9	0.202	0.829	0.719	0.951
TenTrans-C	44.1	0.410	0.680	0.667	0.912	56.1	0.309	0.813	0.617	0.941
TenTrans-P	43.2	0.418	0.671	0.640	0.910	56.5	0.304	0.817	0.640	0.944
UBCNLP-C	25.7	0.574	0.539	-0.263	0.844	51.7	0.316	0.736	0.259	0.905
UBCNLP-P	35.1	0.477	0.622	0.391	0.886	59.9	0.254	0.787	0.538	0.928

	ca2ro				
	BLEU	**TER**	**chrF**	**COMET**	**BertSc**
M2M-100	**33.1**	**0.640**	**0.535**	0.159	0.831
mT5-dFT	17.3	0.830	0.407	-0.461	0.784
BSC-P	24.9	0.695	0.490	-0.076	0.814
CUNI-C	31.8	0.644	0.533	**0.169**	**0.835**
CUNI-P	32.8	**0.640**	**0.535**	0.168	0.834
TenTrans-C	30.2	0.661	0.517	0.047	0.830
TenTrans-P	30.2	0.664	0.516	0.047	0.829
UBCNLP-C	8.6	0.884	0.311	-1.119	0.725
UBCNLP-P	11.2	0.855	0.354	-0.908	0.749

Table 40: Automatic evaluation per language pair in the Romance family of the multilingual low-resource translation task (Task 2). Best scores boldfaced. Notice that the final ranking is done per family and not per language pair as shown in Table 38.

across metrics. The quality of the second baseline, the finetuned version of mT5, is low as compared to the other systems because it has only been trained for machine translation with 5,500 parallel sentences for the 6 language pairs. EdinSaar-Primary is also a version of mT5 finetuned with 1.2 M parallel sentences and that improves translation quality significantly, but still, it lies below the multilingual baseline system trained with huge amounts of parallel data, M2M-100.

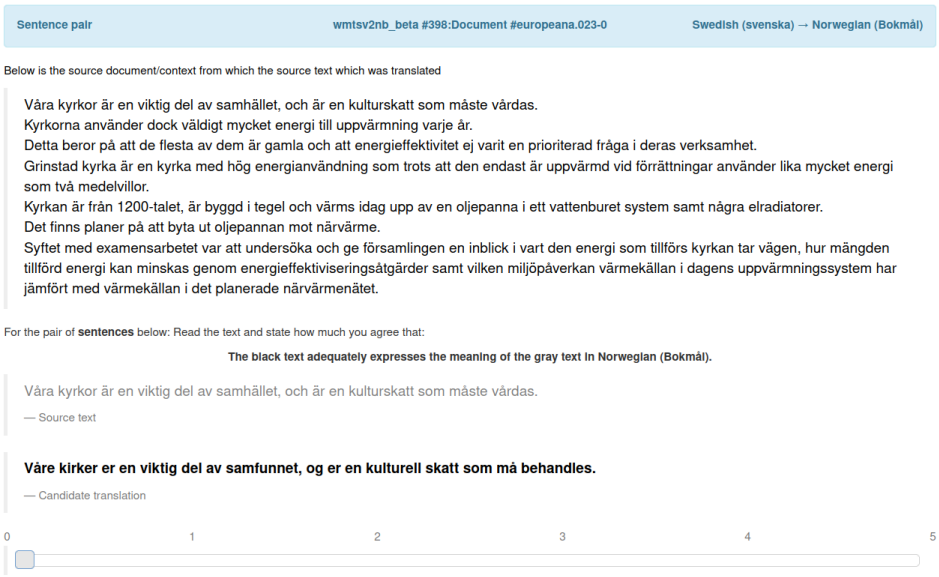
A more fine-grained analysis (Table 39) shows that translation into Icelandic is difficult for all the systems, and also translation from Icelandic into Swedish (Norwegian) is more difficult than translation from Norwegian (Swedish) into Swedish (Norwegian). Systems do not behave consistently across language pairs: mT5-devFinetuned (mT5-dFT in the table) achieves top performance when translating from Icelandic but performs poorly for the remaining pairs; UBCNLP-Contrastive (UBCNLP-C) is specially good for translating from Swedish.

For Task 2, the Romance family, the CUNI systems are significantly better than the rest, both at family and language pair levels (Tables 38 and 40). Only for ca2ro, M2M-100 is better according to some metrics; however, this system performs comparatively bad for ca2it. TenTrans and BSC perform very close one to each other. Globally, TenTrans performs better with BSC showing good performance for ca2oc. For this language pair, the reranking strategy via a character-based model by CUNI achieves very good results.

6.4 Human Evaluation

In order to complement and corroborate the automatic evaluation, we also perform human evaluation on a subset of the languages. However, since not all language pairs are covered, we cannot use the manual evaluation results for the official ranking of the systems.

The type of evaluation has been conditioned by the number and expertise of the raters we could attract. We hired a total of 14 raters: 5 Swedish annotators to rate nb2sv and is2sv documents; 3 bilingual Catalan–Occitan annotators to rate ca2oc documents and 6 bilingual Catalan–Italian annotators to rate ca2it documents. With these numbers in mind, we decided to do ratings on a Likert-like scale but following the philosophy of direct assess-



(a)

(b)

Figure 8: Modifications to the Appraise Evaluation Framework (Federmann, 2018) for the multilingual low-resource translation task. (a) We conduct reference document-level direct assessments on a discrete scale [1,5]. (b) For languages where we can conduct source document-level assessments, we we also evaluate term translation (dates and locations).

ments (DAs). We do source DA for Italian and Occitan, and reference DA for Swedish.

Following the conclusions in (Graham et al., 2020) and (Castilho et al., 2020), we perform sentence level evaluation with document context. Figure 8(a) shows that evaluators rate each sentence in context and when all the sentences in document are evaluated, the whole document is also scored. The evaluation is done using the Appraise Evaluation Framework (Federmann, 2018) with several modifications. Appraise implements document direct assessments as used in the WMT News Task evaluation campaign (Barrault et al., 2020). In our case, we have fewer annotators so we cannot expect > 15 ratings per sentence to get statistically significant results with a 100 points DA scale. To tackle this limitation, we constrain the DA scale to a 5 points Likert-like scale [1,5]. This resembles an adequacy+fluency evaluation where raters still answer the question "*The black text adequately expresses the meaning of the gray text.*", but they do not evaluate adequacy and fluency separately. After a small pilot experiment (see below), the guidelines to the evaluators were the following:

Rank a sentence with a 5 if it completely expresses the same meaning as the source/reference. Notice that we do not ask for a literal translation but for a sentence that preserves the meaning and it is grammatically correct. For a 3 score, the sentence should convey part of the meaning

of the original sentence but some relevant parts are missing or not well translated. For a 4, only non-relevant parts are not OK. For a 2, most of the sentence is wrong but still some bits, probably non-relevant, are well translated. Finally, rate the sentence with a 1 if none of the content is preserved.

Bilingual raters allow us to do a small term translation evaluation for Catalan to Italian and Occitan. Figure 8(b) shows that we boldface some terms in the source text and evaluators are asked to say if *(i) The phrase is not translated*, *(ii) The phrase is well translated* or *(iii) The phrase is mistranslated*.

6.4.1 Data Preparation

We select test documents or parts of them to cover 100 sentences per language. Table 36 shows that considering full documents would limit the evaluation to very few texts so we select a subset of contiguous sentences in documents to make the evaluation more heterogeneous. For Catalan to Italian and Occitan, we selected fragments in 9 documents with lengths between 5 and 15 sentences; for Icelandic to Swedish fragments in 7 documents with lengths between 8 and 20 sentences; and for Norwegian to Swedish fragments in 7 documents with lengths between 7 and 22 sentences.

We extract the same 100 sentences from the participants primary submissions and from the reference. For source DA evaluation (Catalan and Occitan), the reference is also rated and used to establish human performance. For reference DA (Swedish), the reference is just used for rating translations.

Finally, we mark 60 of the source sentences in Catalan with one term each. Selected terms[37] are mostly named entities (dates, locations or titles) and might be multi-word. Named entities that appear only a few times in training data are a challenge for neural systems, so the aim is to check the quality of these translations. Since professional translators did not receive any instructions on how to translate these terms, we can observe a mixture of untranslated and translated named entities, which makes it difficult to assess its quality in an automatic way.

[37]List of terms which translation is evaluated manually: Plaça del Mercadal, segle XV, segle XIX i XX, la Casa Pinyol, Festes de Maig, Rambla de Badalona, la Cremada, la Segona República, Josep Maria Cuyàs, Baró de Maldà, 11 de maig de 1940, Francesc de Paula Giró i Prat, Aristeus antennatus, Productes de l'Empordà, 400 metres, mitjan segle XX, Canyó de Palamós, Confraria de Pescadors de Palamós, finals del segle XIX, Xat de Benaiges, començaments del segle XX, "salvitxada", la calçotada, Alt Camp, Congrés de Cultura Catalana, Valls, Concurs de salsa de la "calçotada", Fogueres de Sant Antoni, Nadal, Sant Antoni, Química Orgànica, Universitat de Barcelona, Junta d'Energia Nuclear, Universitat de Chicago, Universitat de València, Física Teòrica, Mecànica Teòrica, Premi d'Investigació Ramón y Cajal, Manaies de Girona, any 1751, Dijous Sant, Setmana Santa, segles xviii i xix, 1851, mitjans de segle XIX, finals del XVIII, port del Masnou, dos quilòmetres i mig, Club Nàutic del Masnou, Creu Roja, festival Ple de Riure, Masnou, N-II, Premià de Mar, any 2019, platja d'Ocata, Michelin, Ferran Adrià, El Celler de Can Roca, Can Fabes

	nb2sv		is2sv	
System	z-score	raw	z-score	raw
M2M-100	0.7±0.6	4.2±0.8	0.1±1.0	2.0±1.1
EdinSaar	0.2±0.7	3.6±1.1	-0.1±0.8	1.9±1.0
UBCNLP	0.2±0.8	3.5±1.2	-0.4±1.0	1.6±1.1
mT5-dFT	-1.2±0.7	1.5±1.1	0.4±1.1	2.4±1.2

Table 41: Average DA and standard deviation of raw- and z-scores for all primary submissions of Task 1 in the language pairs manually evaluated.

	ca2it		ca2oc	
System	z-score	raw	z-score	raw
HUMAN	0.8±0.4	4.8±0.6	0.8±0.7	4.0±1.0
CUNI	0.5±0.7	4.4±0.9	0.5±0.8	3.6±1.1
M2M-100	0.4±0.7	4.2±1.0	-0.7±0.8	2.0±1.0
TenTrans	0.0±0.8	3.8±1.1	0.3±0.8	3.4±1.2
BSC	-0.1±0.8	3.7±1.1	0.3±0.9	3.4±1.2
UBCNLP	-0.5±1.0	3.1±1.3	0.0±0.9	3.0±1.2
mT5-dFT	-1.2±0.9	2.3±1.2	-1.0±0.7	1.7±0.9

Table 42: Average DA and standard deviation of raw- and z-scores for all primary submissions of Task 2 in the language pairs manually evaluated. HUMAN refers to the evaluation of the reference.

6.4.2 Pilot Experiment

We prepared a pilot experiment with two goals: (i) provide some training to the raters and (ii) check the feasibility of the task. For this, we prepared a manual with instructions to work with the modified Appraise interface and the guidelines for rating the translations. We populate the task with 20 translated sentences from one of the submissions. Sentences come from two test documents so that the annotators go through the full document annotation process twice.

After the pilot, we made the guidelines more concrete to accommodate the raters questions. These annotations are discarded for the final analysis described in the next section.

6.4.3 Results

The results of the evaluation task are the average DA scores per system. In order to take into account that some raters might be more strict than others, we rank the systems according to the z-score, where the DA score is mean-centered and normalised per rater.

Inter-annotator agreement as measured by Fleiss' κ (Fleiss, 1971) is moderate: 0.32±0.03 (nb2sv, fair agreement), 0.16±0.04 (is2sv, slight agreement), 0.28±0.03 (cat2it, fair agreement) and 0.16±0.02 (ca2oc, slight agreement). These values are in agreement with previous analyses (Castilho, 2020). Intra-annotator agreement ranges from 0.88±0.06 to 0.24±0.09 for the North-Germanic languages and from 0.56±0.09 to -0.04±0.07 for the Romance family. We discard raters with $\kappa \sim 0$ and report results with 4 raters for Swedish, 3 for Catalan–Occitan and 4 for Catalan–Italian. Tables 41 and Table 42 show the results for Task 1 and Task 2 respectively.

For Task 1, we obtain very different scores depending on the language pair. This is in line with the automatic evaluation: translations from Icelandic do not behave in the same way as Swedish and Norwegian which are closer languages. Baselines perform very well on this family, but not simultaneously. M2M-100 offers good translation quality for nb2sv while mT5-dFT is specially good for is2sv. For is2sv, systems are not statistically significantly different, for nb2sv mt5-dTF is significantly worse than the others and EdinSaar and UBCNLP show similar performance.

For Task 2, the reference (HUMAN) is ranked first in both language pairs, but the deviation is large and it is not significantly better than the CUNI system. For ca2it, HUMAN is not significantly better than the baseline system M2M-100 either. In some cases though, the distinction seemed to be easy. Raters pointed out several reasons: (i) mistranslations of very frequent words —*got* in Catalan (cup, glass) translated into Italian as *getto* (jet), *grigio* (gray) or *vetro* (glass, the material); (ii) bad translation in context of ambiguous words —*quarentena* in Catalan translates into Italian as *quarantina* (about fourty) or *quarantena* (quarantine); (ii) mistaken roots (this can be related to BPE subunits as explained below) —*calçots* (a local vegetable) translated as *calzatura* (footwear); or changing words —*un físic català* (a Catalan physicist) translated as *un fisico spagnolo* (a Spanish physicist).

	ca2it				ca2oc			
System	**well**	**mis**	**no**	Σ	**well**	**mis**	**no**	Σ
HUMAN	53	0	3	56	40	0	2	42
CUNI	39	3	5	47	30	7	1	38
M2M-100	33	2	6	41	26	9	0	35
TenTrans	37	0	9	46	32	4	1	37
BSC	27	7	5	39	33	4	0	37
UBCNLP	29	16	1	46	19	1	0	20
mT5-dFT	20	17	10	47	25	11	4	40

Table 43: Number of **well** translated, **mis**-translated and **no**t translated terms for the language pairs manually evaluated for Task 2. The last column per language shows the total number of terms considered from the maximum of 60 bold faced terms (see text).

Similar to the automatic evaluation, TenTrans and BSC are very close to each other according to the human ratings although the two architectures are completely different. The evaluation also confirms the bad performance of M2M-100 on ca2oc but its good performance on ca2it. In general, all the systems perform worse on ca2oc than ca2it according to the raw scores in Table 42, but the trend is reversed when analysing the z-scores. This result points to differences between the scale that annotators used in the two tasks even if they received the same instructions. Notice that almost all automatic metrics but COMET tend to score higher ca2oc than ca2it for most systems.

Term translation. The evaluation against the source for the Romance languages allows us to study the translation quality of selected terms. For ca2it we use the annotations from 5 raters but only 2 were considered for ca2oc as the remaining raters did not do the task properly. The agreement for this task is 0.34±0.05 (ca2it) and 0.19±0.05 (ca2oc). Table 43 shows the number of well translated, mis-translated and untranslated terms for both pairs.

For each term, we sum the votes from all the raters per class (well translated, mis-translated or untranslated) and consider the winning class the one with the majority of votes. In case there is a tie with 2 or more classes, the term is not considered in the analysis, this is why the last columns Σ in Table 43 differ from 60. The disagreement is high, and one of the causes is the ambiguity in the annotation of toponyms. For instance, the name of the city of "Valls" has been evaluated 17 times: 7 times as well translated and 10 times as not translated being always the translation

"Valls". The same happens with other toponyms and years. This ambiguity damages specially the majority voting for Occitan (low Σ) since we only consider 2 raters.

The systems with the largest number of mistranslations are those with less access to the task languages, that is, the baselines. mT5-devFinetuned and M2M-100 (specially for Occitan) do the most mistakes. A curious case is UBNLP which only produces 1 mistranslation for Occitan but 16 for Italian. Also BSC generates more errors for Italian (7) than for Occitan (4) even though translation quality into Italian is higher than into Occitan. Looking at some examples, we hypothesise that this can be related to the subunit segmentation strategy. For instance, the word "calçotada" is translated as *calzotada, calzolata* or as we have seen before *calzatura* in Italian, where no Italian word for this concept exists. For Occitan, it is always translated by *calçotada* (BPE units in Catalan and Occitan might be the same, but not for Italian), only two times it is mistranslated as *escòla*.

Besides these errors that might be due to the split in subunits, we also observe multi-word named entities where one of the words has been literally translated and the others have not. Also, in few occasions, a number (specially centuries) is translated by another one.

6.5 Discussion

This shared task faced three challenges: multilingual translation, document translation and in-domain (cultural heritage) translation. 60% of the submissions approached multilinguality with a single system while 40% used a combination of several bilingual systems. None of the participants focused on the document-level aspect of the task, and those who dealt with the specific domain did not use any of the in-domain multilingual lexicons but selected in-domain data from the available training corpus.

More details and comparisons among the submissions can be found in Figures 9 and 10. Figure 9 focuses on the resources. Participants did not use all the data available, probably because of its heterogeneous nature and the difference of language pairs available in the different corpora. WikiMatrix is the favourite corpus, with 80% of the submissions trained on it. 90% of the systems used some kind of pretrained model:

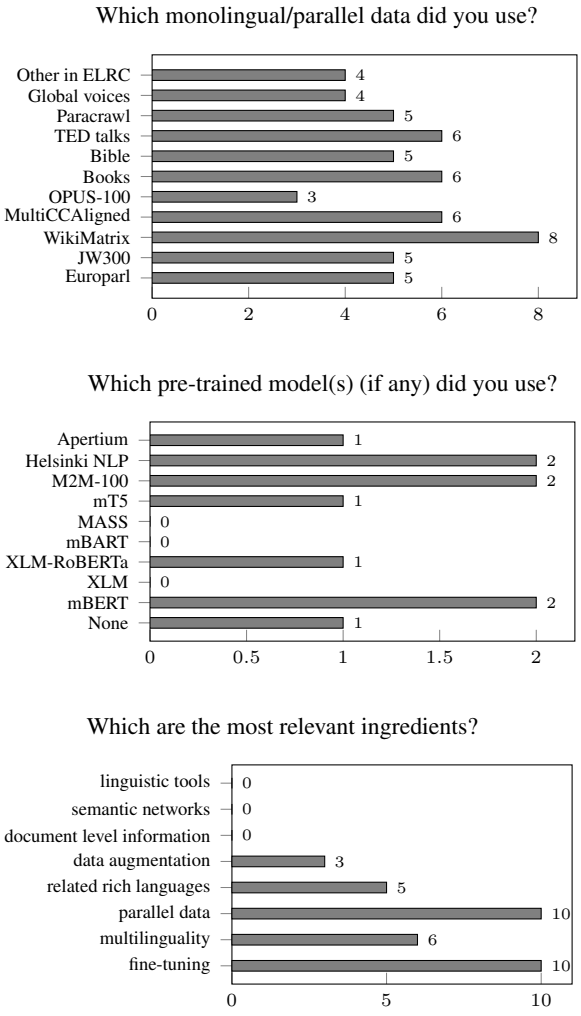


Figure 9: Resources used by the participants to train the systems submitted to the multilingual low-resource translation task (10 responses).

from language models such as mBERT (TenTrans, EdinSaar) or XLM-RoBERTa (BSC) to machine translation models such as M2M-100 (TenTrans) or Helsinki's NLP (UBCNLP). There is no clear favourite system here, and each team followed a different approach. In all cases, systems were finetuned with language specific data, either data made available for the task or backtranslations made by themselves. 50% of the submissions also used data from the related high resourced languages for training.

Figure 10 compares the architectures. As expected, neural systems dominate the number of submissions. In fact, all of them where 100% neural, without any hybridisation with any non-neural component. All participants used direct translation, either multilingual (60%) or bilingual (40%), but none of them submitted translations done

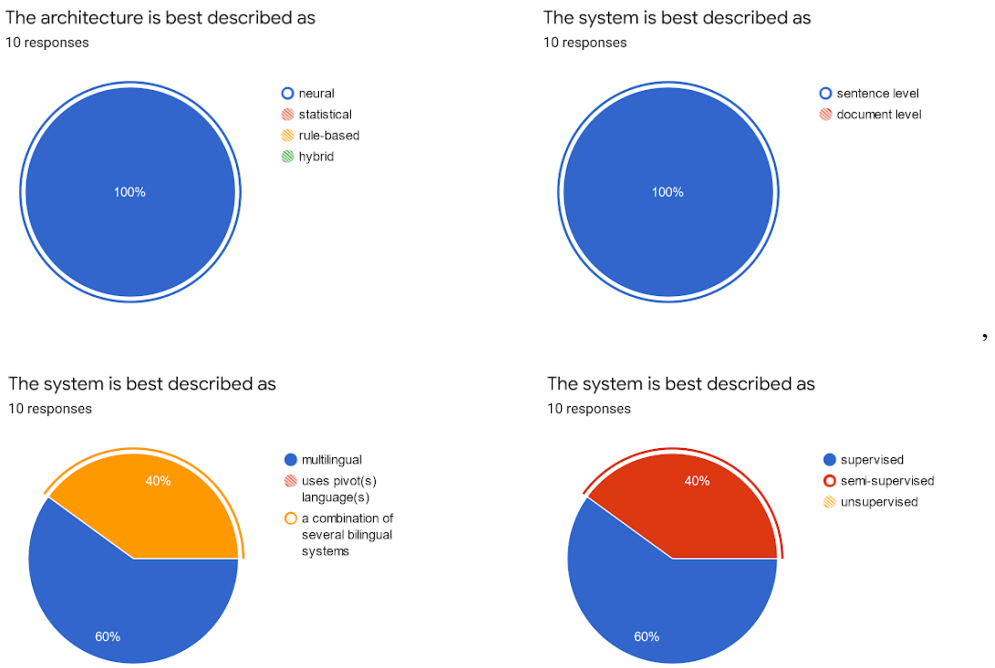


Figure 10: Main characteristics of the systems submitted to the multilingual low-resource translation task. Percentages are over the sample of 10 submissions.

through a pivot language. One team, CUNI, tried pivot through English for the Romance languages but translation quality was significantly better with direct systems. TenTrans used a pivot language for creating a synthetic corpus using backtranslation. Similarly to CUNI's, the approach worked well for ca2it and ca2ro but did not work at all for the lowest resourced language, Occitan, damaging the quality of the multilingual system as a whole. In both cases, multilingual systems trained with parallel data of the task languages plus additional corpora with the related rich languages as source gave the best performance.

Data augmentation via backtranslations and/or parallel data including high-resourced languages have been beneficial for all the systems. Two teams also got improvements by selecting data close to the domain of the validation set, but the in-domain adaptation was not decisive to win the shared task. TenTrans extracted in-domain sentences with a domain classifier trained on mBERT in Task 2 while EdinSaar used cross-entropy for the same purpose in Task 1.

In this shared task, we have evaluated systems per family, but differences among translation pairs are significant and determine the final ranking. The trends for the 2 families are similar. One of the languages has a relatively large amount of data (Swedish/Italian), the second language in terms of amount of data is the most distant one within the family (Icelandic/Romanian) and the lowest-resourced language is linguistically very similar to the richest language (Norwegian Bokmål/Occitan). Icelandic is the bottleneck for Task 1 and Romanian for Task 2 showing that in this case the distance between languages is more important than the amount of data.

It is interesting to see how the ranking depends on the language pair. The most extreme case is our baseline mT5-devFinetuned which performed the best when translating from Icelandic and the worst in the other cases (Task 1). Similarly but not so extreme, UBCNLP-Contrastive performed very well when translating from Swedish and significantly worse on the other cases. In Task 2, Romance languages, the two baselines specially M2M-100, are penalised by the bad performance on ca2oc showing that the amount of Occitan text might be too diluted in their multilingual training. M2M-100 is the best for ca2ro, and this is the only pair where the best system is not CUNI. For all the systems, ca2ro is the most difficult pair.

Finally, we want to emphasise the correlation between automatic and human evaluations among systems even though standard deviations are high and top performing systems are not significantly

different.

7 Automatic Post Editing

This section presents the results of the 7^{th} round of the WMT 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-revised machine-translated output. In continuity with last year, the challenge consisted of fixing the errors present in English Wikipedia pages translated – into German and Chinese – by state-of-the-art, not domain-adapted neural MT (NMT) systems unknown to participants. Despite a number of data downloads in line with the previous rounds, this year we observed an unexpected drop in participation: two teams participated in the English-German task, submitting two runs each, while the English-Chinese task had no participants. Most likely, this setback can be ascribed to the difficulty to handle the released test data, which are characterized by NMT output of very high quality. This is reflected by much higher baseline results compared to last year (18.05 TER / 71.07 BLEU for en-de, 22.73 TER / 69.2 BLEU for en-zh), which only one run was able to improve according to both the automatic metrics used (-0.77 for the primary TER metric and +0.48 for the secondary BLEU metric). Nevertheless, the outcomes of human evaluation still reveal the ability of APE systems to improve MT output quality: significant gains over the baseline are indeed observed for all the participating systems.

7.1 The Task

MT Automatic Post-Editing (APE) is the task of automatically correcting 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.

This 7^{th} round of the WMT APE shared task kept the same overall evaluation setting of the previous six rounds. Specifically, the participating systems had to automatically correct the output of an unknown "black box" (neural) MT system by learning from training data containing human revisions of translations produced by the same system. The selected language pairs (English-German and English-Chinese) and the data domain (Wikipedia articles) were the same of last year (Chatterjee et al., 2020), as well as the type of MT systems (generic NMT systems not adapted to the target domain).

7.2 Data, Metrics, Baseline

7.2.1 Data

In continuity with all previous rounds, participants were provided with **training** and **development** data consisting of (*source*, *target*, *human post-edit*) triplets (7,000 for the training and 1,000 for the development sets for both languages) where:

- The source (SRC) is a tokenized English sentence;
- The target (TGT) is a tokenized German/Chinese translation of the source, which was produced by a generic, black-box NMT system unknown to participants.[38]
- The human post-edit (PE) is a tokenized manually-revised version of the target, which was produced by professional translators.

For the English-German sub-task, two additional training resources were made available to participants. These are: *i)* the corpus of 4.5 million artificially-generated post-editing triplets described in (Junczys-Dowmunt and Grundkiewicz, 2016), and *ii)* the 14.5 million artificially-generated instances of the English-German section of the eSCAPE corpus (Negri et al., 2018).

[38]The NMT systems for both the languages are based on the standard Transformer architecture (Vaswani et al., 2017) and follow the implementation details described in (Ott et al., 2018). They were trained on publicly available MT datasets including Paracrawl (Bañón et al., 2020) and Europarl (Koehn, 2005), summing up to 23.7M parallel sentences for English-German and 22.6M for English-Chinese.

Test data consisted of newly-released (*source*, *target*) pairs (1,000 in total for each target language), similar in nature to the corresponding elements in the train/dev sets (i.e. same domain, same NMT architectures). The human post-edits of the target elements were left apart to measure APE systems' performance both with automatic metrics (TER, BLEU) and via manual assessments.

7.2.2 Metrics

Also this year, the participating systems were evaluated both by means of automatic metrics and manually (see Section 7.5). Automatic evaluation was carried out by computing the distance between the automatic post-edits produced by each system for the target elements of the test set, and the human corrections of the same test items. Case-sensitive TER (Snover et al., 2006) and BLEU (Papineni et al., 2002) were respectively used as primary and secondary evaluation metrics. The official systems' ranking is hence based on the average TER calculated on the test set by using the TERcom[39] software: lower average TER scores correspond to higher ranks. BLEU was computed using the multi-bleu.perl package[40] available in MOSES. Automatic evaluation results are presented in Section 7.5.1.

Manual evaluation was conducted via source-based direct human assessment (Graham et al., 2013). Complete details are provided in Section 7.5.3.

7.2.3 Baseline

Also this year, the official baseline results were the TER and BLEU scores calculated by comparing the raw MT output with human post-edits. This corresponds to the score achieved by a "*do-nothing*" APE system that leaves all the test targets unmodified. For each submitted run, the statistical significance of performance differences with respect to the baseline was calculated with the bootstrap test (Koehn, 2004).

7.3 Complexity indicators

To get an idea of the difficulty of the task, in previous rounds we have focused on three aspects of the released data, which provide us with information about the possibility of learning useful correction patterns during training and successfully applying them at test time. These are: *i)* repetition rate, *ii)* MT quality, and *iii)* TER distribution in the test set. For the sake of comparison across the seven rounds of the APE task (2015–2021), Table 44 reports, for each dataset, information about the first two aspects. The third one, instead, will be discussed by referring to Figure 11. Concerning this year's round, we only report information for the English-German sub-task, the only one for which we had participants; also the discussion henceforth will exclusively focus on this sub-task.

7.3.1 Repetition Rate

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 that may suggest a higher chance of learning from the training set correction patterns that are applicable also to the test set. However, over the years, the influence of repetition rate in the data on systems' performance was found to be marginal.[41] For the sake of completeness, we hence just observe that, being drawn from the same Wikipedia domain, this year's data feature very low repetitiveness values (i.e. 0.73, 0.78, and 0.76 respectively for the SRC, TGT and PE elements), which are comparable to those from last year (0.653, 0.823, and 0.656). In spite of this, while last year's gains over the baseline were the highest ever observed in the APE task history, this year's results are significantly lower. This suggests the higher importance of other complexity factors, on which repetition rate might have an additive effect that still has to be fully understood.

7.3.2 MT Quality

MT quality, that is the initial quality of the machine-translated (TGT) texts to be corrected, is indeed a much more reliable indicator of task difficulty. We measure it by computing, the TER (↓) and BLEU (↑) scores using the human post-edits as reference. As discussed in (Bojar et al., 2017; Chatterjee et al., 2018, 2019, 2020) higher quality of the original translations leaves to the APE systems a smaller room for improvement since they have, at the same time, less to learn during training and less to correct at test stage. On one

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

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

[41] The analyses carried out over the years produced mixed outcomes, with impressive final results obtained in spite of low repetition rates (Chatterjee et al., 2020) and vice-versa (Chatterjee et al., 2018, 2019).

	Lang.	Domain	MT type	RR_SRC	RR_TGT	RR_PE	Baseline BLEU	Baseline TER	δ TER
2015	en-es	News	PBSMT	2.9	3.31	3.08	n/a	23.84	+0.31
2016	en-de	IT	PBSMT	6.62	8.84	8.24	62.11	24.76	-3.24
2017	en-de	IT	PBSMT	7.22	9.53	8.95	62.49	24.48	-4.88
2017	de-en	Medical	PBSMT	5.22	6.84	6.29	79.54	15.55	-0.26
2018	en-de	IT	PBSMT	7.14	9.47	8.93	62.99	24.24	-6.24
2018	en-de	IT	NMT	7.11	9.44	8.94	74.73	16.84	-0.38
2019	en-de	IT	NMT	7.11	9.44	8.94	74.73	16.84	-0.78
2019	en-ru	IT	NMT	18.25	14.78	13.24	76.20	16.16	+0.43
2020	en-de	Wiki	NMT	0.65	0.82	0.66	50.21	31.56	-11.35
2020	en-zh	Wiki	NMT	0.81	1.27	1.2	23.12	59.49	-12.13
2021	en-de	Wiki	NMT	0.73	0.78	0.76	71.07	18.05	-0.77

Table 44: Basic information about the APE shared task data released since 2015: languages, domain, type of MT technology, repetition rate and initial translation quality (TER/BLEU of TGT). The last row (δ TER) indicates, for each evaluation round, the difference in TER between the baseline (i.e. the "*do-nothing*" system) and the top-ranked submission. For this year's round we report results for the only sub-task – English-German – for which we had participants.

side, 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 *i)* drastically reduce the number of corrections required and the applicability of the learned patterns, and *ii)* increase the chance to introduce errors, especially when post-editing near-perfect TGTs. The findings of all previous rounds of the task support this observation and, as discussed in Section 7.5, this year is no exception. For English-German, the quality of the initial translations (18.05 TER / 71.07 BLEU) is close the level of the "hardest" previous rounds (2017-2019), characterized by baseline scores in the 15.5-16.8 TER interval (and BLEU>70.0). Accordingly, this year's gains over the baseline amount to less than 1 TER/BLEU points. The strict correlation between the quality of the initial translations and the actual potential of APE is hence confirmed.

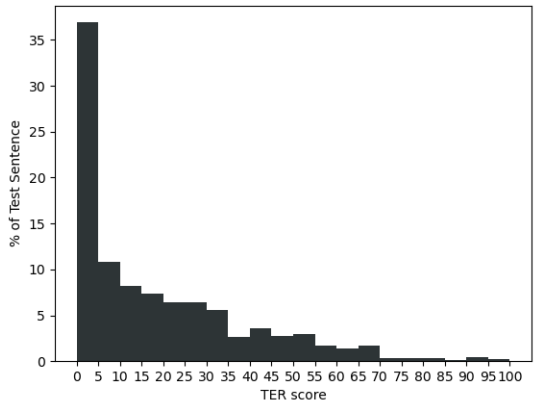


Figure 11: TER distribution in the **English-German** test set.

7.3.3 TER Distribution

A third reliable complexity indicator is the TER distribution (computed against human references) for the translations present in the test sets. Although TER distribution and MT quality can be seen as two sides of the same coin, it's worth remarking that, even at the same level of overall quality, more/less peaked distributions can result in very different testing conditions. Indeed, as shown by previous analyses, harder rounds of the tasks were typically characterized by TER distributions particularly skewed towards low values (i.e. a larger percentage of test items having a TER between 0 and 10). On one side, the higher the proportion of (near-)perfect test instances requiring few edits or no corrections at all, the higher the probability that APE systems will perform unnecessary corrections penalized by automatic evaluation metrics. On the other side, less skewed distributions can be expected to be easier to handle as they give to automatic systems a larger room for improvement (i.e. more test items requiring - at least minimal - revision). In the lack of more focused analyses on this aspect, we can hypothesize that, in ideal conditions from the APE standpoint, the peak of the distribution would be observed for "post-editable" translations containing enough errors that leave some margin for focused corrections, but not too many errors to be so unintelligible to require a whole re-translation from scratch.[42]

Also with respect to this complexity indicator, this year's test set looks particularly difficult to handle. As shown in Figure 11, more than 35%

[42]For instance, based on the empirical findings reported in (Turchi et al., 2013), TER=0.4 is the threshold that, for human post-editors, separates the "post-editable" translations from those that require complete rewriting from scratch.

ID	Participating team
PVIE	Amazon Prime Video, India (Sharma et al., 2021)
Netmarble	Netmarble AI Center, South Korea Korea (Oh et al., 2021)

Table 45: Participants in the WMT21 Automatic Post-Editing task.

of the test instances feature a TER between 0 and 5 and almost 50% of them have 0<TER<10. This distribution, which is very different from last year (where less than 7% of the test samples had 0<TER<5 and ∼55% of them had 15<TER<45), is similar to the one featured by the most challenging datasets from previous rounds.

All in all, the small gains over the baseline mentioned above also confirm the strict correlation between TER distribution and task difficulty. This goes hand in hand with the above considerations about MT quality and, together with the possible additive effect of very low repetition rate values in raising the difficulty bar, might have discouraged potential participants.

7.4 Submissions

As shown in Table 45, we received submissions from two teams, which is indeed a significant drop with respect to last year's round. Moreover, as anticipated, both teams participated only in the English-German sub-task by submitting 2 runs each.

Amazon Prime Video (*PVIE*). Amazon participated with a model leveraging a state-of-the-art MT system based on fairseq (Ott et al., 2019) and pre-trained on data from the WMT'19 News Translation task (Barrault et al., 2019). The basic model is first fine-tuned on the APE dataset, by creating (*source*, *target*) pairs where the *source* is a concatenation of the SRC and MT elements of the APE data and the *target* is the corresponding PE element. Then, to cope with the domain mismatch between the initial training data and the APE task ones, the model is fine-tuned on *i)* data drawn from WikiMatrix (Schwenk et al., 2019) (64k parallel sentences after cleaning), *ii)* additional APE samples (45k triplets) from previous rounds (2016-2018) of the shared task, and *iii)* this year's APE data. The primary submission is obtained by ensembling models built from different combinations of the available data.

Netmarble AI Center (*Netmarble*). Netmarble participated with a Transformer-based system (Vaswani et al., 2017) built using: *i)* the WMT21 News Translation data, *ii)* the additional artificial synthetic data provided to the APE task participants, and *iii)* data augmentation techniques that make use of an external MT component. These resources are processed through a curriculum training procedure aimed to step-wise learn from easier problems to more complex ones. Multi-task learning is also applied to alleviate data sparsity issues by sharing knowledge across related tasks (in this case part of speech recognition, named entity recognition, masked language modeling and keep/translate classification). All tasks are jointly trained and, to cope with imbalanced data from the selected tasks, task-specific losses – namely focal loss (Lin et al., 2017) and class-balanced loss (Cui et al., 2019) - are exploited in addition to standard cross-entropy. Moreover, dynamic weight average (Liu et al., 2019), which adapts the task weighting over time by considering the rate of change of the loss for each task, is applied to optimize the contribution of each task in the multi-task framework.

7.5 Results

7.5.1 Automatic evaluation

Participants' results are shown in Table 46. 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. We also report the BLEU score, computed using the same references, which represents our secondary evaluation metric.

As it can be seen from the table, the two rankings slightly differ: while the top submission (17.28 TER, 71.55 BLEU) is the same, the BLEU-based ranking presents few swaps, with the *do nothing* baseline reaching the 2nd position. One obvious observation is that these fluctuations are due to the fact that all systems substantially perform on par: except for one case (i.e. the 0.77 TER reduction achieved by the top submission), all the results' differences with respect to the baseline are indeed not statistically significant.

Quite surprisingly, we also observe that the best

		TER	BLEU
en-de	Netmarble_CURRICULUM-ENSEMBLE_CONTRASTIVE	**17.28**	71.55
	PVIE_single_CONTRASTIVE	17.74	70.54
	PVIE_ensemble_PRIMARY	17.85	70.5
	Netmarble_CURRICULUM-MTL_PRIMARY	17.97	70.53
	Baseline	18.05	71.07

Table 46: Results for the WMT21 APE English-German – average TER (↓), BLEU score (↑) Statistically significant improvements over the baseline are marked in **bold**.

submission for both participants is the contrastive one. This highlights the difficulty to select the best configuration during system development, and indirectly confirms the difficulty to handle APE data characterized by very high MT quality, TER distribution skewed towards perfect/near-perfect translations and very low repetition rate values.

7.5.2 Systems' behaviour

Modified, improved and deteriorated sentences. In light of the hard conditions posed by what seems to be the hardest APE dataset ever released, we now turn an eye toward the changes made by each system to the test instances. To this aim, Table 47 shows, for each submitted run, the number of modified, improved and deteriorated sentences, as well as the overall system's precision (i.e. the proportion of improved sentences out of the total number of modified instances for which improvement/deterioration is observed). It's worth noting that, as in the previous rounds, 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 any TER variations. This grey area, for which quality improvement/degradation can not be automatically assessed, contributes to motivate the human evaluation discussed in Section 7.5.3.

As it can be seen from the table, systems' behaviour reflects the difficulty to handle this year's test set. The quite low percentage of modified sentences (50.2 on average, 46.2 for the top submission) is in line with our previous observations about TER distribution (see Section 7.3.1). With ∼50% of the test instances having 0<TER<10, all systems seem to have properly managed the small room for intervention by not exceeding the number of expected corrections. Accordingly, different from last year,[43] systems' final scores are inversely proportional to their aggressiveness.

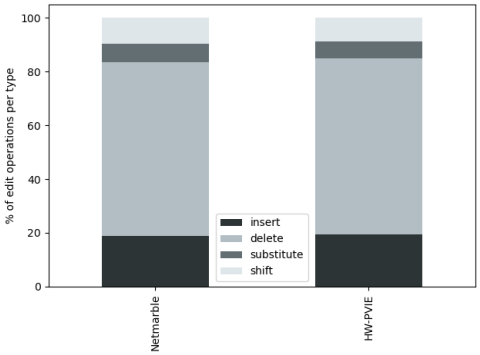


Figure 12: Distribution of edit operations (insertions, deletions, substitutions and shifts) performed by the two primary submissions to the English-German task.

Precision-wise, however, we are far from last year's values: despite lower aggressiveness, system's precision is 51.12 on average (in 2020 it was 58.0) with the best run peaking at 53.96 (vs 69.0 in 2020). This is due to significant variations in the percentage of improved (43.5 on average, 45.67 for the top submission) and deteriorated sentences (41.6 on average, 38.96 for the winning system), which are very different from last year where, on a simpler test set, the average values were respectively 58.2 and 23.6.

Edit operations. Similar to previous rounds, we analysed systems' behaviour also in terms of the distribution of edit operations (insertions, deletions, substitutions and shifts) done by each system. This fine-grained analysis of how systems corrected the test set instances is obtained by computing the TER between the original MT output and the output of each primary submission taken as reference. Similar to last year, and in line with the close TER/BLEU results obtained by the two systems, differences in their behaviour are barely visible. Both of them are characterised

[43] On the much simpler 2020 test set, featuring only ∼15.0% of instances with 0≤TER≤10, the modified sentences were 69.2% on average, with the more aggressive behaviour of the top systems peaking to more than 90.5%.

Systems	Modified	Improved	Deteriorated	Prec.
Netmarble_CURRICULUM-ENSEMBLE_CONTRASTIVE	462 (46.2%)	211 (45.67%)	180 (38.96%)	53.96
PVIE_single_CONTRASTIVE	504 (50.4%)	212 (42.06%)	212 (42.06%)	50.0
PVIE_ensemble_PRIMARY	508 (50.8%)	215 (42.32%)	218 (42.91%)	49.65
Netmarble_CURRICULUM-MTL_PRIMARY	533 (53.3%)	235 (44.09%)	227 (42.59%)	50.87
Average	50.2	43.5	41.6	51.12

Table 47: Number (raw and proportion) of test sentences modified, improved and deteriorated by each run submitted to the APE 2021 **English-German** sub-task. The "Prec." column shows systems' precision as the ratio between the number of improved sentences and the number of modified instances for which improvement/deterioration is observed (i.e. Improved + Deteriorated).

	Avg	Avg z
Netmarble_CURRICULUM-MTL_PRIMARY	79.82	0.144
Netmarble_CURRICULUM-ENSEMBLE_CONTRASTIVE	78.52	0.095
PVIE_ensemble_PRIMARY	76.85	0.02
PVIE_single_CONTRASTIVE	76.67	0.011
test.mt	69.68	-0.27

Table 48: Results for the WMT21 APE **English-German – human evaluation**. Systems ordered by DA score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$.

by a large number of deletions (65.0% on average), followed by insertions (19.2%), shifts (9.2%) and substitutions (6.5%). Although this year's test set turned out to be very different in terms of difficulty, this distribution is practically identical to last year. More thorough future investigations would be needed to find clear explanations for these observations. For the time being, to get further insights about systems' performance, we now complement our analysis by discussing the outcomes of human evaluation of the submitted runs.

7.5.3 Human evaluation

In order to complement the automatic evaluation of APE submissions, manual evaluation of the 4 submissions for English-German was conducted. In this section, we present the evaluation procedure, as well as the results obtained.

7.6 Evaluation procedure

We evaluated the overall quality of the MT and PE output using source-based direct assessment (Graham et al., 2013; Cettolo et al., 2017; Bojar et al., 2018b). We used the same instructions that are used in the News Translation track of WMT2021. Instead of using crowd-workers, we hired 2 professional translators for English-German that are native German speakers as suggested by Freitag et al. (2021a).

Human evaluation results for English-German are summarized in Table 48. Similar to last year's task (Chatterjee et al., 2020), all 4 submissions significantly improved the original MT output. Furthermore, the APE system of *Netmarble_CURRICULUM-MTL_PRIMARY* significantly outperforms all other submission and can be declared as the single winner of this years' APE task. Interestingly, the human evaluation results show no correlation with the automatic scores from Table 46 which confirms the findings from Freitag et al. (2019) that automatic evaluation is hard for post-edited systems.

7.7 Summary

We presented the results from the 7^{th} shared task on Automatic Post-Editing at WMT. This round of the challenge featured the same overall setting of last year. Specifically, the language directions were the same (English-German and English-Chinese), as well as the domain of the data (Wikipedia articles) and the neural MT systems used to produce the translations to be automatically post-edited. Also the evaluation process was carried out in continuity with the past, both with automatic metrics (TER and BLEU, respectively the primary and secondary metrics) and by means of human evaluation (via source-based direct assessment, similar to the News Translation track but involving professional translators). According to several complexity indicators (repetition rate, original MT quality and TER distribution), this year's data can be safely considered as the most difficult one ever released. On one side, this might have discouraged potential participants, which were only two for the English-German sub-task. On the other side, it contributes to explain the lower results compared to last year.

Indeed, only one submitted run was able to achieve statistically significant improvements over the *do-nothing* baseline in terms of our primary automatic metric. Nevertheless, all submissions were consistently ranked higher by human evaluators, indicating the effectiveness of APE technology even under such extremely challenging conditions.

8 Conclusion

The news translation task in 2021 covered 20 translation pairs, 14 of which had English on the source or target side and 6 were without English. Direct assessment (DA) was the main golden truth again, although the style varied across language pairs. Into-English translation was evaluated against human reference translation, preserving the order of sentences in a document but not presenting the whole document at once (SR+DC). Out-of-English and some of non-English pairs offered the full document context to the annotators and allowed them to revisit the scores assigned to individual segments (SR+FD), evaluating against the source. Four non-English pairs used a simpler evaluation without any document context (SR−DC). For English→Czech, English→German and Chinese→English, a contrastive DA scoring was also tested, presenting individual sentences in pairs of candidate translations (contr:SR-DC), aimed at a more discerning pairwise comparisons. And finally, an alternative scoring style called GENIE was additionally applied to German→English.

Document context was found to be extremely important for evaluation of high-quality MT systems. The ranking of participating systems differs considerably between SR+FD and contr:SR-DC. In particular, human reference is scored well if full document context is available throughout the annotation but tends to be surpassed by top systems when sentences are evaluated in isolation. Surprising effects were also observed when using these evaluation methods on different human translations.

The triangular machine translation task encouraged participants to use all the parallel data provided (involving direct and indirect sources) to build a better machine translation system for the particular language pair and direction (Russian-to-Chinese). The participants explored several modeling choices and data augmentation strategies that would help practitioners when building machine translation systems involving non-English language pairs.

The multilingual low-resource translation task dealt with two Indo-European language families: North Germanic and Romance. The best performing systems used multilingual supervised machine translation models enriched with backtranslated data and additional sentences from higher-resourced languages in the same family. Pivot translation via these high-resourced counter-parts and in-domain data selection was not beneficial for the final performance.

The results of the task on automatic post-editing were highly influenced by the difficulty of this year's data, which can also explain a drop in participation (two teams, only in the English-German sub-task). In light of the very high quality of the translation to be automatically corrected, the very skewed TER distribution towards near-perfect translations and the very low repetition rate in the data, it comes as no surprise that only one run was able to outperform the strong *do-nothing* baseline with statistically significant improvements. Nevertheless, human evaluation results reveal significant gains by all runs, attesting the difficulty to apply automatic evaluation procedures to APE and, on a positive note, the effectiveness of the proposed methods.

Acknowledgments

The organizers of the automatic post-editing task would like to thank Apple and Google Research for their support and sponsorship in organizing this round of the APE challenge. The organizers of the triangular machine translation task would like to thank DiDi Chuxing for providing data and research time to support this shared task.

The multilingual low-resource translation for Indo-European languages task has been funded by the European Language Resource Coordination ELRC (SMART 2019/1083) and LT-BRIDGE (H2020, 952194), and supported by the Directorate-General for Language Policy, Ministry of Culture. Government of Catalonia. We are thankful to Europeana for providing source texts in Icelandic, Norwegian and Swedish and to Antonio Toral for fruitful discussions on human evaluation.

For the news task, we are very grateful to the sponsors of our test sets. Translation of the tests sets received funding from the Eu-

ropean Union's Horizon 2020 Research and Innovation Programme under Grant Agreement Nos. 825460 and 825303 (Elitr and Bergamot, for Czech↔English) and No. 825299 (GoURMET, for Hausa↔English). The translation of the German↔English and Chinese↔English test sets was funded by Microsoft, the Russian↔English test sets were funded by Yandex, the Japanese↔English test sets by University of Tokyo and NTT and the French↔German test sets by LinguaCustodia. The Icelandic↔English task was sponsored by the Language Technology Programme for Icelandic 2019–2023. The programme, which is managed and coordinated by Almannarómur, is funded by the Icelandic Ministry of Education, Science and Culture. The Bengali↔Hindi and Xhosa↔Zulu test sets were sponsored by Facebook. The human evaluation was co-funded by Microsoft, Toloka AI, and Facebook. The effort that goes into the manual evaluation campaign each year is impressive, and we are grateful to all participating individuals and teams for their work. We are also grateful to the many workers who contributed to the human evaluation via Mechanical Turk.

The organizers of the Similar Languages Task would like to thank Pangeanic for the Spanish, Catalan, Portuguese, and Romanian data and the Directorate-General for Language Policy at the Ministry of Culture, Government of Catalonia for the Catalan translations. They further thank the AI Journal - Funding Opportunities for Promoting AI Research for supporting the French - Maninka data collection. The French - Bambara dataset is partially funded by a grant awarded by the Lacuna Fund within the scope of the program "Datasets for Languages in Sub-Saharan Africa". We also thank Andrij Rovenchak for the support on data collection. Marta R. Costa-jussà would like to acknowledge the support of the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 947657).

Ondřej Bojar would like to acknowledge the grant no. 19-26934X (NEUREM3) of the Czech Science Foundation for his time as well as co-funding manual annotation.

Support was provided by Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Trinity College Dublin funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

References

Md Mahfuz Ibn Alam, Ivana Kvapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu, Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn, and Vassilina Nikoulina. 2021. Findings of the wmt shared task on machine translation using terminologies. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, Aljoscha Burchardt, and Sebastian Möller. 2020. Fine-grained linguistic evaluation for state-of-the-art machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic Evaluation of German-English Machine Translation Using a Test Suite. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

George Awad, Asad Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Alan Smeaton, Yvette Graham, Gareth Jones, Wessel Kraaij, and Georges Quenot. 2021. Trecvid 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains.

George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Luca Diduch, Alan F. Smeaton, Yvette Graham, and Wessel Kraaij. 2019. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID*, volume 2019.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Starkaður Barkarson and Steinþór Steingrímsson. 2019. Compiling and filtering ParIce: An English-Icelandic parallel corpus. In *Proceedings of the*

22nd Nordic Conference on Computational Linguistics, pages 140–145, Turku, Finland. Linköping University Electronic Press.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Cristina España-Bonet, Josu Boldoba, and Lluís Màrquez. 2015. A Factory of Comparable Corpora from Wikipedia. In *Proceedings of the Eighth Workshop on Building and Using Comparable Corpora*, pages 3–13.

Chao Bei and Hao Zong. 2021. Gtcom neural machine translation systems for wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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, 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, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018a. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. 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. 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, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019. *Proceedings of the Fourth Conference on Machine Translation*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Jiří Mírovský, Kateřina Rysová, and Magdaléna Rysová. 2018b. EvalD Reference-Less Discourse Evaluation for WMT18. 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.

Sheila Castilho. 2020. On the same page? comparing inter-annotator agreement in sentence and document level human machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1150–1159, Online. Association for Computational Linguistics.

Sheila Castilho, Maja Popović, and Andy Way. 2020. On context span needed for machine translation evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3735–3742, Marseille, France. European Language Resources Association.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the iwslt 2017 evaluation campaign. In *Proc. of IWSLT*, Tokyo, Japan.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. 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: Shared Task Papers*, pages 710–725, Belgium, Brussels. 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.

Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. The University of Edinburgh's English-German and English-Hausa submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Wei-Rui Chen and Muhammad Abdul-Mageed. 2021. Machine translation of low-resource indo-european languages. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Marta R. Costa-jussà, Marcos Zampieri, and Santanu Pal. 2018. A Neural Approach to Language Variety Translation. In *Proceedings of VarDial*.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269.

Rohit Dholakia and Anoop Sarkar. 2014. Pivot-based triangulation for low-resource languages. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas (AMTA)*, volume 1, pages 315–328.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Grant Erdmann, Jeremy Gwinnup, and Tim Anderson. 2021. Tune in: The afrl wmt21 news-translation systems. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Carlos Escolano, Ioannis Tsiamas, Christine Basta, Javier Ferrando, Marta R. Costa-jussà, and José A. R. Fonollosa. 2021. The talp-upc participation in wmt21 news translation task: an mbart-based nmt

approach. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Cristina España-Bonet, Alberto Barrón-Cedeño, and Lluís Màrquez. 2020. Tailoring and Evaluating the Wikipedia for in-Domain Comparable Corpora Extraction.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Christian Federmann. 2012. Appraise: an open-source toolkit for manual evaluation of mt output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, George F. Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *CoRR*, abs/2104.14478.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Petr Gebauer, Ondřej Bojar, Vojtěch Švandelík, and Martin Popel. 2021. Cuni systems in wmt21: Revisiting backtranslation techniques for english-czech nmt. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *arXiv:2106.03193 [cs]*. ArXiv: 2106.03193.

Yvette Graham, George Awad, and Alan Smeaton. 2018. Evaluation of automatic video captioning using direct assessment. *PLOS ONE*, 13(9):1–20.

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.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in Machine Translation Evaluation. *arXiv e-prints*, page arXiv:1906.09833.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, Christian Federmann, and Tom Kocmi. 2021. On user interfaces for large-scale document-level human evaluation of machine translation outputs. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 97–106, Online. Association for Computational Linguistics.

Hangcheng Guo, Wenbin Liu, Yanqing He, Tian Lan, Hongjiao Xu, Zhenfeng Wu, and You Pan. 2021. Istic's triangular machine translation system for wmt2021. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Barry Haddow and Faheem Kirefu. 2020. Pmindia–a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

Hossein Hassani. 2017. Kurdish Interdialect Machine Translation. *Proceedings of VarDial*.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the wmt 2021 shared task on efficient translation. In *Proceedings of the*

Sixth Conference on Machine Translation, Online. Association for Computational Linguistics.

Amr Hendy, Esraa A. Gad, Mohamed Abdelghaffar, Jailan S. ElMosalami, Mohamed Afify, Ahmed Y. Tawfik, and Hany Hassan Awadalla. 2021. Ensembling of distilled models from multi-task teachers for constrained resource language pairs. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Josef Jon, Michal Novák, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. Cuni systems for wmt21: Multilingual low-resource translation for indo-european languages shared task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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. Association for Computational Linguistics.

Haukur Jónsson, Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Pétur Orri Ragnarson, and Vilhjálmur Þorsteinsson. 2021. Miðeind's wmt 2021 submission. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ksenia Kharitonova, Ona de Gibert Bonet, Jordi Armengol-Estapé, Mar Rodriguez i Alvarez, and Maite Melero. 2021. Transfer learning with shallow decoders: Bsc at wmt2021's multilingual low-resource translation for indo-european languages shared task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel S. Weld. 2021. GENIE: A leaderboard for human-in-the-loop evaluation of text generation.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. *CoRR*, abs/1909.09524.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation. *arXiv e-prints*, page arXiv:2107.10821.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

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.

Mikołaj Koszowski, Karol Grzegorczyk, and Tsimur Hadeliya. 2021. Allegro.eu submission to wmt21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Samuel Laubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research (JAIR)*, 67.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018a. 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*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018b. Has Neural Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In *EMNLP 2018*, Brussels, Belgium. Association for Computational Linguistics.

Giang Le, Shinka Mori, and Lane Schwartz. 2021. Illinois Japanese ↔ English News Translation for WMT 2021. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Zongyao Li, Daimeng Wei, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Zhengzhe Yu, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021a. Hw-tsc's participation in the wmt 2021 triangular mt shared task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Zuchao Li, Masao Utiyama, Eiichiro Sumita, and Hai Zhao. 2021b. Miss@wmt21: Contrastive learning-reinforced domain adaptation in neural machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the wmt 2021 shared tasks in unsupervised mt and very low resource supervised mt. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as Parallel Corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 176–186, Sofia, Bulgaria. Association for Computational Linguistics.

Huan Liu, Junpeng Liu, Kaiyu Huang, and Degen Huang. 2021a. Dutnlp machine translation system for wmt21 triangular translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021b. Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nikola Ljubešić and Antonio Toral. 2014. caWaC – a web corpus of Catalan and its application to language modeling and machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1728–1732, Reykjavik, Iceland. European Language Resources Association (ELRA).

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. Fine-grained evaluation of German-English Machine Translation based on a Test Suite. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova, and Sebastian Möller. 2021. Linguistic evaluation for the 2021 state-of-the-art machine translation systems for german to english and english to german. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ander Martinez. 2021. The fujitsu dmath submissions for wmt21 news translation and biomedical translation tasks. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shivam Mhaskar and Pushpak Bhattacharyya. 2021. Pivot based transfer learning for neural machine translation: Cfilt iitb @ wmt 2021 triangular mt. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China. Association for Computational Linguistics.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3603–3609.

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).

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. 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 72–78, New Orleans, Louisiana. Association for Computational Linguistics.

Artur Nowakowski and Tomasz Dwojak. 2021. Adam mickiewicz university's english-hausa submissions to the wmt 2021 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI Center's WMT21 Automatic Post-Editing Shared Task Submission. In *Proceedings of the Sixth Conference on Machine Translation*, Online.

Csaba Oravecz, Katina Bontcheva, David Kolovratník, Bhavani Bhaskar, Michael Jellinghaus, and Andreas Eisele. 2021. etranslation's submissions to the wmt 2021 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*.

Proyag Pal, Alham Fikri Aji, Pinzhen Chen, and Sukanta Sen. 2021. The University of Edinburgh's Bengali-Hindi submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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.

Jeonghyeok Park, Hyunjoong Kim, and Hyunchang Cho. 2021. Papago's submissions to the wmt21 triangular translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Nikita Pavlichenko, Ivan Stelmakh, and Dmitry Ustalov. 2021. Crowdspeech and vox DIY: Benchmark dataset for crowdsourced audio transcription. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Maja Popović, Alberto Poncelas, Marija Brkic, and Andy Way. 2020. Neural machine translation for translating into Croatian and Serbian. In *Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2020. Glancing transformer for non-autoregressive neural machine translation. *arXiv preprint arXiv:2008.07905*.

Lihua Qian, Yi Zhou, Zaixiang Zheng, Yaoming ZHU, Zehui Lin, Jiangtao Feng, Shanbo Cheng, Lei Li, Mingxuan Wang, and Hao Zhou. 2021. The volctrans glat system: Non-autoregressive translation meets wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia. *arXiv e-prints*, page arXiv:1907.05791.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting Neural Machine Translation for Automatic Post-Editing. In *Proceedings of the Sixth Conference on Machine Translation*, Online.

Shashank Siripragada, Jerin Philip, Vinay P. Namboodiri, and C V Jawahar. 2020. A multilingual parallel corpora collection effort for Indian languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3743–3751, Marseille, France. European Language Resources Association.

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, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A very large Icelandic text corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams, and Oleksii Kuchaiev. 2021. Nvidia nemo's neural machine translation systems for english-german and english-russian news and biomedical tasks at wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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.

Allahsera Auguste Tapo, Bakary Coulibaly, Sébastien Diarra, Christopher Homan, Julia Kreutzer, Sarah Luger, Arthur Nagashima, Marcos Zampieri, and Michael Leventhal. 2020. Neural machine translation for extremely low-resource african languages: A case study on bambara. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 23–32.

Svetlana Tchistiakova, Jesujoba Alabi, Koel Dutta Chowdhury, Sourav Dutta, and Dana Ruiter. 2021. Edinsaar@wmt21: North-germanic low-resource multilingual nmt. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2009. News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, pages 237–248. John Benjamins.

Jörg Tiedemann and Lars Nygaard. 2004. The opus corpus-parallel and free: http://logos.uio.no/opus. In *Proceedings of LREC*.

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).

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018a. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018b. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Belgium, Brussels. Association for Computational Linguistics.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai's wmt21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Marco Turchi, Matteo Negri, and Marcello Federico. 2013. Coping with the subjectivity of human judgements in MT quality estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 240–251, Sofia, Bulgaria. Association for Computational Linguistics.

Masao Utiyama and Hitoshi Isahara. 2007. A Comparison of Pivot Methods for Phrase-Based Statistical Machine Translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, Rochester, New York. 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.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021. Tencent translation system for the wmt21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. Hw-tsc's participation in the wmt 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings on the wmt 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Hua Wu and Haifeng Wang. 2009. Revisiting Pivot Language Approach for Machine Translation. 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, pages 154–162, Suntec, Singapore. Association for Computational Linguistics.

Jitao Xu, Minh Quang Pham, Sadaf Abdul Rauf, and François Yvon. 2021. LISN @ WMT 2021. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Han Yang, Bojie Hu, Wanying Xie, ambyera han, Pan Liu, Jinan Xu, and Qi Ju. 2021. Tentrans multilingual low-resource translation system for wmt21 indo-european languages task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Lana Yeganova, Dina Wiemann, Mariana Neves, Federica Vezzani, Amy Siu, Inigo Jauregi Unanue, Maite Oronoz, Nancy Mah, Aurélie Névéol, David Martinez, Rachel Bawden, Giorgio Maria Di Nunzio, Roland Roller, Philippe Thomas, Cristian Grozea, Olatz Perez-de Viñaspre, Maika Vicente Navarro, and Antonio Jimeno Yepes. 2021. Findings of the wmt 2021 biomedical translation shared task: Summaries of animal experiments as new test set. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Hui Zeng. 2021. Small model and in-domain data are all you need. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. Wechat neural machine translation systems for wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8545–8554. Association for Computational Linguistics.

Shiyu Zhao, Xiaopu Li, Minghui Wu, and Jie Hao. 2021. The mininglamp machine translation system for wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao, and Jingbo Zhu. 2021. The niutrans machine translation systems for wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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.

A Differences in Human Scores

Tables 49–59 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.

Tables 49-68 provide automatic metric scores (COMET, BLEU, chrF) for all competing systems.

	FACEBOOK-AI	ONLINE-A	CUNI-DOCTRANSFORMER	ONLINE-B	CUNI-TRANSFORMER2018	ONLINE-W	ONLINE-G	ONLINE-Y	HUMAN
FACEBOOK-AI	-	0.03	0.10⋆	0.12⋆	0.12‡	0.14‡	0.15‡	0.20‡	0.20‡
ONLINE-A	-0.03	-	0.07†	0.09†	0.09‡	0.11‡	0.12‡	0.17‡	0.17‡
CUNI-DOCTRANSFORMER	-0.10	-0.07	-	0.01	0.02	0.04	0.05†	0.09†	0.09⋆
ONLINE-B	-0.12	-0.09	-0.01	-	0.00	0.03	0.03⋆	0.08†	0.08†
CUNI-TRANSFORMER2018	-0.12	-0.09	-0.02	0.00	-	0.02	0.03	0.08⋆	0.08
ONLINE-W	-0.14	-0.11	-0.04	-0.03	-0.02	-	0.01	0.05⋆	0.05
ONLINE-G	-0.15	-0.12	-0.05	-0.03	-0.03	-0.01	-	0.05	0.05
ONLINE-Y	-0.20	-0.17	-0.09	-0.08	-0.08	-0.05	-0.05	-	0.00
HUMAN	-0.20	-0.17	-0.09	-0.08	-0.08	-0.05	-0.05	0.00	-
score	0.11	0.08	0.01	-0.01	-0.01	-0.03	-0.04	-0.08	-0.09
rank	1–2	1–2	3–6	3–6	3–8	3–8	5–9	7–9	5–9
bleu-A	31.1	28.3	30.2	**31.7**	26.2	28.9	28.6	24.6	-
chrF-A	**.599**	.569	.585	.593	.551	.576	.575	.549	-
comet-A	**.628**	.534	.592	.557	.510	.595	.517	.459	.358
bleu-B	**26.4**	23.5	24.7	24.8	21.7	24.8	22.8	20.3	-
chr-B	**.549**	.520	.532	.531	.504	.534	.520	.502	-
comet-B	**.513**	.411	.466	.431	.391	.486	.383	.322	.414

Table 49: Head to head comparison for Czech→English systems

	NiuTrans	KwaiNLP	DIDI-NLP	HUMAN	HappyNewYear	P3AI	Borderline	ICL	MiSS	IIE-MT	Machine-Translation	SMU	YYDS	Facebook-AI	Online-B	ZengHuiMT	HuaweiTSC	XMU	NJUSC-TSC	Online-G	Online-W	UF	Online-A	Online-Y
NiuTrans	-	0.00	0.01	0.02	0.03	0.04⋆	0.07†	0.07†	0.07‡	0.07†	0.07‡	0.08‡	0.08‡	0.08‡	0.08‡	0.09‡	0.10‡	0.11‡	0.11‡	0.12‡	0.13‡	0.15‡	0.15‡	0.22‡
KwaiNLP	0.00	-	0.01	0.02	0.02	0.04	0.06⋆	0.06⋆	0.07†	0.07	0.07‡	0.07†	0.07†	0.08⋆	0.08†	0.09†	0.10‡	0.11‡	0.11‡	0.12‡	0.13‡	0.14‡	0.14‡	0.21‡
DIDI-NLP	-0.01	-0.01	-	0.01	0.02	0.03	0.05†	0.06†	0.06†	0.06⋆	0.06‡	0.07‡	0.07‡	0.07†	0.07‡	0.08‡	0.09‡	0.10‡	0.10‡	0.11‡	0.12‡	0.13‡	0.14‡	0.21‡
HUMAN	-0.02	-0.02	-0.01	-	0.00	0.02	0.04	0.04	0.05⋆	0.05	0.05†	0.05⋆	0.05⋆	0.06⋆	0.06⋆	0.07⋆	0.08†	0.09†	0.09‡	0.10‡	0.11†	0.12‡	0.13‡	0.19‡
HappyNewYear	-0.03	-0.02	-0.02	0.00	-	0.02	0.04	0.04	0.04⋆	0.05	0.05†	0.05⋆	0.05⋆	0.05⋆	0.06⋆	0.07⋆	0.08†	0.08†	0.09‡	0.10‡	0.10†	0.12‡	0.12‡	0.19‡
P3AI	-0.04	-0.04	-0.03	-0.02	-0.02	-	0.02	0.03	0.03	0.03	0.03†	0.03⋆	0.04⋆	0.04	0.04⋆	0.05⋆	0.06⋆	0.07⋆	0.07‡	0.08‡	0.09†	0.10‡	0.11‡	0.17‡
Borderline	-0.07	-0.06	-0.05	-0.04	-0.04	-0.02	-	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.05	0.05⋆	0.06⋆	0.06	0.08†	0.08‡	0.15‡
ICL	-0.07	-0.06	-0.06	-0.04	-0.04	-0.03	0.00	-	0.00	0.01	0.01⋆	0.01	0.01	0.01	0.02	0.02	0.04	0.04	0.05†	0.06†	0.06	0.08‡	0.08‡	0.15‡
MiSS	-0.07	-0.07	-0.06	-0.05	-0.04	-0.03	-0.01	0.00	-	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.03	0.04	0.04⋆	0.05⋆	0.06	0.07†	0.08‡	0.14†
IIE-MT	-0.07	-0.07	-0.06	-0.05	-0.05	-0.03	-0.01	-0.01	0.00	-	0.00⋆	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04†	0.05†	0.06⋆	0.07†	0.08‡	0.14‡
Machine-Translation	-0.07	-0.07	-0.06	-0.05	-0.05	-0.03	-0.01	-0.01	0.00	0.00	-	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.05	0.05	0.07	0.07⋆	0.14⋆
SMU	-0.08	-0.07	-0.07	-0.05	-0.05	-0.03	-0.01	-0.01	-0.01	0.00	0.00	-	0.00	0.00	0.01	0.02	0.03	0.03	0.04	0.05	0.05	0.07⋆	0.07†	0.14†
YYDS	-0.08	-0.07	-0.07	-0.05	-0.05	-0.04	-0.01	-0.01	-0.01	0.00	0.00	0.00	-	0.00	0.01	0.01	0.03	0.03	0.04	0.05	0.05	0.07⋆	0.07†	0.14⋆
Facebook-AI	-0.08	-0.08	-0.07	-0.06	-0.05	-0.04	-0.01	-0.01	-0.01	-0.01	-0.01	0.00	0.00	-	0.00	0.01	0.03	0.03	0.04⋆	0.04	0.05	0.07†	0.07†	0.14†
Online-B	-0.08	-0.08	-0.07	-0.06	-0.06	-0.04	-0.02	-0.02	-0.01	-0.01	-0.01	-0.01	-0.01	0.00	-	0.01	0.02	0.03	0.03	0.04	0.04	0.06⋆	0.06†	0.13†
ZengHuiMT	-0.09	-0.09	-0.08	-0.07	-0.07	-0.05	-0.03	-0.02	-0.02	-0.02	-0.02	-0.02	-0.01	-0.01	-0.01	-	0.01	0.02	0.02	0.03	0.04	0.05⋆	0.06†	0.12†
HuaweiTSC	-0.10	-0.10	-0.09	-0.08	-0.08	-0.06	-0.04	-0.04	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.02	-0.01	-	0.01	0.01	0.02	0.02	0.04⋆	0.04†	0.11⋆
XMU	-0.11	-0.11	-0.10	-0.09	-0.08	-0.07	-0.05	-0.04	-0.04	-0.04	-0.04	-0.03	-0.03	-0.03	-0.03	-0.02	-0.01	-	0.00	0.01	0.02	0.03	0.04†	0.11⋆
NJUSC-TSC	-0.11	-0.11	-0.10	-0.09	-0.09	-0.07	-0.05	-0.05	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.03	-0.02	-0.01	0.00	-	0.01	0.01	0.03	0.03	0.10
Online-G	-0.12	-0.12	-0.11	-0.10	-0.10	-0.08	-0.06	-0.06	-0.05	-0.05	-0.05	-0.05	-0.05	-0.04	-0.04	-0.03	-0.02	-0.01	-0.01	-	0.01	0.02	0.02	0.09
Online-W	-0.13	-0.13	-0.12	-0.11	-0.10	-0.09	-0.06	-0.06	-0.06	-0.06	-0.05	-0.05	-0.05	-0.05	-0.04	-0.04	-0.02	-0.02	-0.01	-0.01	-	0.02	0.02⋆	0.09⋆
UF	-0.15	-0.14	-0.13	-0.12	-0.12	-0.10	-0.08	-0.08	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.06	-0.05	-0.04	-0.03	-0.03	-0.02	-0.02	-	0.00	0.07
Online-A	-0.15	-0.14	-0.14	-0.13	-0.12	-0.11	-0.08	-0.08	-0.08	-0.08	-0.07	-0.07	-0.07	-0.07	-0.06	-0.06	-0.04	-0.04	-0.03	-0.02	-0.02	0.00	-	0.07
Online-Y	-0.22	-0.21	-0.21	-0.19	-0.19	-0.17	-0.15	-0.15	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.13	-0.12	-0.11	-0.11	-0.10	-0.09	-0.09	-0.07	-0.07	-
score	0.04	0.04	0.03	0.02	0.02	-0.00	-0.02	-0.03	-0.03	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.07	-0.08	-0.09	-0.10	-0.11	-0.17
rank	1–5	1–6	1–6	1–9	1–9	2–19	4–18	4–19	6–17	3–24	9–22	7–21	7–21	6–20	7–21	7–21	7–21	7–22	12–24	11–24	8–22	17–24	20–24	20–24
bleu	31.9	32.2	32.5	-	29.3	29.6	**33.4**	30.1	32.6	31.7	29.8	32.2	32.7	32.1	32.4	32.2	28.9	31.9	28.7	27.2	28.5	30.2	24.6	24.1
chrF	.611	.611	.617	-	.584	.593	**.617**	.594	.613	.609	.588	.609	.612	.604	.613	.601	.593	.594	.590	.564	.575	.595	.545	.543
comet	**.487**	.476	.462	-	.417	.459	.483	.433	.483	.462	.433	.467	.478	.477	.438	.413	.434	.421	.427	.323	.358	.447	.253	.248

Table 50: Head to head comparison for Chinese→English systems

	Borderline	Online-A	Online-W	UF	VolcTrans-AT	Facebook-AI	ICL	Online-G	Online-B	Online-Y	VolcTrans-GLAT	P3AI	SMU	UEdin	Nemo	Manifold	Watermelon	happypoet	HUMAN	HuaweiTSC
Borderline	-	0.00	0.00	0.01	0.02	0.03⋆	0.06⋆	0.08⋆	0.11‡	0.11‡	0.12†	0.12†	0.13‡	0.13‡	0.14‡	0.16‡	0.17‡	0.19‡	0.21‡	0.25‡
Online-A	0.00	-	0.00	0.01	0.02	0.02	0.06⋆	0.08	0.11‡	0.11‡	0.11†	0.12†	0.13‡	0.13‡	0.13‡	0.16‡	0.17‡	0.19‡	0.21‡	0.24‡
Online-W	0.00	0.00	-	0.01	0.02⋆	0.02⋆	0.05†	0.07†	0.11‡	0.11‡	0.11‡	0.11‡	0.13‡	0.13‡	0.13‡	0.16‡	0.16‡	0.18‡	0.20‡	0.24‡
UF	-0.01	-0.01	-0.01	-	0.01⋆	0.01⋆	0.05†	0.06⋆	0.10‡	0.10‡	0.10‡	0.11‡	0.12‡	0.12‡	0.12‡	0.15‡	0.16‡	0.17‡	0.19‡	0.23‡
VolcTrans-AT	-0.02	-0.02	-0.02	-0.01	-	0.01	0.04	0.06	0.09⋆	0.09⋆	0.10⋆	0.10	0.11†	0.11†	0.12†	0.14‡	0.15‡	0.17†	0.19‡	0.23‡
Facebook-AI	-0.03	-0.02	-0.02	-0.01	-0.01	-	0.03	0.05	0.08†	0.08†	0.09⋆	0.09	0.11†	0.11†	0.11†	0.13‡	0.14‡	0.16⋆	0.18‡	0.22‡
ICL	-0.06	-0.06	-0.05	-0.05	-0.04	-0.03	-	0.02	0.05⋆	0.05	0.06	0.06	0.08⋆	0.08⋆	0.08⋆	0.10†	0.11‡	0.13	0.15‡	0.19‡
Online-G	-0.08	-0.08	-0.07	-0.06	-0.06	-0.05	-0.02	-	0.03	0.03	0.04	0.04	0.06⋆	0.06⋆	0.06⋆	0.08†	0.09‡	0.11⋆	0.13‡	0.17‡
Online-B	-0.11	-0.11	-0.11	-0.10	-0.09	-0.08	-0.05	-0.03	-	0.00	0.01	0.01	0.02	0.02	0.03	0.05	0.06⋆	0.08	0.10⋆	0.14‡
Online-Y	-0.11	-0.11	-0.11	-0.10	-0.09	-0.08	-0.05	-0.03	0.00	-	0.01	0.01	0.02	0.02	0.03	0.05	0.06⋆	0.08	0.10†	0.14‡
VolcTrans-GLAT	-0.12	-0.11	-0.11	-0.10	-0.10	-0.09	-0.06	-0.04	-0.01	-0.01	-	0.00	0.02	0.02	0.02	0.04	0.05†	0.07	0.09†	0.13‡
P3AI	-0.12	-0.12	-0.11	-0.11	-0.10	-0.09	-0.06	-0.04	-0.01	-0.01	0.00	-	0.01	0.02	0.02	0.04⋆	0.05†	0.07	0.09†	0.13‡
SMU	-0.13	-0.13	-0.13	-0.12	-0.11	-0.11	-0.08	-0.06	-0.02	-0.02	-0.02	-0.01	-	0.00	0.00	0.03	0.03	0.05	0.07	0.11†
UEdin	-0.13	-0.13	-0.13	-0.12	-0.11	-0.11	-0.08	-0.06	-0.02	-0.02	-0.02	-0.02	0.00	-	0.00	0.03	0.03⋆	0.05	0.07⋆	0.11†
Nemo	-0.14	-0.13	-0.13	-0.12	-0.12	-0.11	-0.08	-0.06	-0.03	-0.03	-0.02	-0.02	0.00	0.00	-	0.02	0.03⋆	0.05	0.07⋆	0.11‡
Manifold	-0.16	-0.16	-0.16	-0.15	-0.14	-0.13	-0.10	-0.08	-0.05	-0.05	-0.04	-0.04	-0.03	-0.03	-0.02	-	0.01	0.03	0.05	0.08⋆
Watermelon	-0.17	-0.17	-0.16	-0.16	-0.15	-0.14	-0.11	-0.09	-0.06	-0.06	-0.05	-0.05	-0.03	-0.03	-0.03	-0.01	-	0.02	0.04	0.08
happypoet	-0.19	-0.19	-0.18	-0.17	-0.17	-0.16	-0.13	-0.11	-0.08	-0.08	-0.07	-0.07	-0.05	-0.05	-0.05	-0.03	-0.02⋆	-	0.02⋆	0.06‡
HUMAN	-0.21	-0.21	-0.20	-0.19	-0.19	-0.18	-0.15	-0.13	-0.10	-0.10	-0.09	-0.09	-0.07	-0.07	-0.07	-0.05	-0.04	-0.02	-	0.04
HuaweiTSC	-0.25	-0.24	-0.24	-0.23	-0.23	-0.22	-0.19	-0.17	-0.14	-0.14	-0.13	-0.13	-0.11	-0.11	-0.11	-0.08	-0.08	-0.06	-0.04	-
score	0.13	0.12	0.12	0.11	0.11	0.10	0.07	0.05	0.02	0.02	0.01	0.01	-0.01	-0.01	-0.01	-0.04	-0.04	-0.06	-0.08	-0.12
rank	1–5	1–6	1–4	4	3–8	4–9	5–12	4–12	8–17	7–17	7–17	5–16	9–19	9–17	9–17	10–19	15–20	7–17	16–20	18–20
bleu-A	34.9	34.0	32.9	33.8	34.4	34.0	32.5	33.4	33.8	32.1	**35.0**	33.2	33.8	33.7	33.3	33.0	34.5	31.2	-	34.6
chrF-A	**.629**	.625	.619	.619	.623	.625	.616	.619	.619	.605	.625	.614	.622	.620	.617	.614	.623	.592	-	.625
comet-A	.612	.616	.622	.618	.607	.638	.618	.593	.617	.556	.612	.600	.589	.608	.621	.590	.608	.450	.531	**.628**
bleu-B	39.0	39.6	39.7	40.1	39.7	**41.9**	39.6	38.8	40.2	37.5	39.5	37.9	38.7	40.2	40.6	37.8	39.8	34.6	-	40.8
chrF-B	.654	.657	.661	.658	.652	**.671**	.660	.649	.657	.634	.652	.643	.651	.656	.656	.642	.652	.610	-	.659
comet-B	.620	.628	.650	.638	.626	**.670**	.642	.609	.638	.568	.624	.612	.599	.638	.641	.605	.626	.446	.564	.644

Table 51: Head to head comparison for German→English systems

	NEMO	ONLINE-W	ONLINE-B	HUMAN	MANIFOLD	FACEBOOK-AI	NIUTRANS	ONLINE-G	AFRL	ONLINE-A	ONLINE-Y
NEMO	-	0.01	0.03⋆	0.05	0.08	0.08	0.09⋆	0.12‡	0.15‡	0.16‡	0.26‡
ONLINE-W	-0.01	-	0.02⋆	0.04	0.07⋆	0.07	0.09†	0.11‡	0.14‡	0.15‡	0.25‡
ONLINE-B	-0.03	-0.02	-	0.02	0.05	0.05	0.06	0.09†	0.12‡	0.13‡	0.23‡
HUMAN	-0.05	-0.04	-0.02	-	0.03	0.03	0.04	0.07†	0.10‡	0.11‡	0.21‡
MANIFOLD	-0.08	-0.07	-0.05	-0.03	-	0.00	0.02	0.04⋆	0.07†	0.08†	0.18‡
FACEBOOK-AI	-0.08	-0.07	-0.05	-0.03	0.00	-	0.01	0.04†	0.07‡	0.08‡	0.18‡
NIUTRANS	-0.09	-0.09	-0.06	-0.04	-0.02	-0.01	-	0.03	0.06⋆	0.07⋆	0.17‡
ONLINE-G	-0.12	-0.11	-0.09	-0.07	-0.04	-0.04	-0.03	-	0.03	0.04	0.14⋆
AFRL	-0.15	-0.14	-0.12	-0.10	-0.07	-0.07	-0.06	-0.03	-	0.01	0.11
ONLINE-A	-0.16	-0.15	-0.13	-0.11	-0.08	-0.08	-0.07	-0.04	-0.01	-	0.10
ONLINE-Y	-0.26	-0.25	-0.23	-0.21	-0.18	-0.18	-0.17	-0.14	-0.11	-0.10	-
score	0.14	0.13	0.11	0.09	0.06	0.06	0.04	0.02	-0.01	-0.02	-0.12
rank	1–5	1–4	3–7	1–7	2–7	1–7	3–8	7-10	8–11	8–11	9–11
bleu-A	40.2	37.0	40.6	-	41.1	**42.3**	41.8	41.2	38.8	38.7	32.8
chrF-A	.660	.631	.661	-	.659	.661	.658	**.668**	.635	.652	.600
comet-A	.625	.610	.624	.619	.619	**.656**	.632	.635	.595	.595	.524
bleu-B	40.1	37.2	40.0	-	40.5	**41.6**	41.2	40.7	39.6	38.8	33.2
chrF-B	.663	.635	.663	-	.661	.663	.661	**.671**	.640	.657	.602
comet-B	.619	.606	.621	.619	.614	**.647**	.623	.629	.589	.591	.523

Table 52: Head to head comparison for Russian→English systems

	HUAWEITSC	IIE-MT	NIUTRANS	KWAINLP	FACEBOOK-AI	XMU	CAPITALMARVEL	ONLINE-B	MISS	ONLINE-W	WECHAT-AI	ONLINE-A	ONLINE-G	MOVELIKEAJAGUAR	ONLINE-Y	ILLINI
HUAWEITSC	-	0.06⋆	0.09⋆	0.11‡	0.11⋆	0.12‡	0.13‡	0.14‡	0.17‡	0.18‡	0.20‡	0.22‡	0.28‡	0.30‡	0.33‡	0.33‡
IIE-MT	-0.06	-	0.04	0.05	0.05	0.06†	0.07⋆	0.08†	0.11†	0.12†	0.14‡	0.16‡	0.22‡	0.24‡	0.27‡	0.27‡
NIUTRANS	-0.09	-0.04	-	0.01	0.01	0.02⋆	0.03	0.04⋆	0.08†	0.08⋆	0.11‡	0.13‡	0.19‡	0.20‡	0.23‡	0.24‡
KWAINLP	-0.11	-0.05	-0.01	-	0.00	0.01	0.02	0.03	0.06⋆	0.07	0.10†	0.11‡	0.17‡	0.19‡	0.22‡	0.23‡
FACEBOOK-AI	-0.11	-0.05	-0.01	0.00	-	0.01⋆	0.02	0.03⋆	0.06⋆	0.07⋆	0.09‡	0.11‡	0.17‡	0.19‡	0.22‡	0.22‡
XMU	-0.12	-0.06	-0.02	-0.01	-0.01	-	0.01	0.02	0.06	0.06	0.09	0.11⋆	0.17‡	0.18‡	0.21‡	0.22‡
CAPITALMARVEL	-0.13	-0.07	-0.03	-0.02	-0.02	-0.01	-	0.01	0.04	0.05	0.07†	0.09‡	0.15‡	0.17‡	0.20‡	0.20‡
ONLINE-B	-0.14	-0.08	-0.04	-0.03	-0.03	-0.02	-0.01	-	0.03	0.04	0.06	0.08⋆	0.14‡	0.16‡	0.19‡	0.19‡
MISS	-0.17	-0.11	-0.08	-0.06	-0.06	-0.06	-0.04	-0.03	-	0.01	0.03	0.05⋆	0.11‡	0.13†	0.16‡	0.16‡
ONLINE-W	-0.18	-0.12	-0.08	-0.07	-0.07	-0.06	-0.05	-0.04	-0.01	-	0.02	0.04†	0.10‡	0.12‡	0.15‡	0.15‡
WECHAT-AI	-0.20	-0.14	-0.11	-0.10	-0.09	-0.09	-0.07	-0.06	-0.03	-0.02	-	0.02	0.08⋆	0.09⋆	0.13†	0.13†
ONLINE-A	-0.22	-0.16	-0.13	-0.11	-0.11	-0.11	-0.09	-0.08	-0.05	-0.04	-0.02	-	0.06	0.08	0.11⋆	0.11⋆
ONLINE-G	-0.28	-0.22	-0.19	-0.17	-0.17	-0.17	-0.15	-0.14	-0.11	-0.10	-0.08	-0.06	-	0.02	0.05	0.05
MOVELIKEAJAGUAR	-0.30	-0.24	-0.20	-0.19	-0.19	-0.18	-0.17	-0.16	-0.13	-0.12	-0.09	-0.08	-0.02	-	0.03	0.04
ONLINE-Y	-0.33	-0.27	-0.23	-0.22	-0.22	-0.21	-0.20	-0.19	-0.16	-0.15	-0.13	-0.11	-0.05	-0.03	-	0.00
ILLINI	-0.33	-0.27	-0.24	-0.23	-0.22	-0.22	-0.20	-0.19	-0.16	-0.15	-0.13	-0.11	-0.05	-0.04	0.00	-
score	0.14	0.08	0.05	0.03	0.03	0.03	0.01	0.00	-0.03	-0.04	-0.06	-0.08	-0.14	-0.16	-0.19	-0.19
rank	1	2–5	2–6	2–9	2–6	5–11	3–10	5–11	6–11	5–11	7–12	11–14	12–16	12–16	13–16	13–16
bleu	26.5	25.4	27.2	25.8	27.7	25.8	23.7	27.2	27.0	22.8	**27.8**	21.0	20.6	21.2	17.3	18.6
chrf	.528	.521	.532	.524	**.536**	.524	.496	.526	.529	.489	.535	.455	.476	.476	.482	.453
comet	.348	.314	.371	.307	**.392**	.307	.236	.270	.294	.270	.361	.167	.145	.182	.061	.073

Table 53: Head to head comparison for Japanese→English systems

	FACEBOOK-AI	MANIFOLD	NIUTRANS	ONLINE-B	HUAWEITSC	MIDEIND	ONLINE-A	ALLEGRO	ONLINE-Y	ONLINE-G
FACEBOOK-AI	-	0.18‡	0.25‡	0.26‡	0.28‡	0.28‡	0.29‡	0.33‡	0.37‡	0.55‡
MANIFOLD	-0.18	-	0.07⋆	0.08‡	0.10†	0.10†	0.11‡	0.15‡	0.19‡	0.37‡
NIUTRANS	-0.25	-0.07	-	0.02	0.03	0.04	0.04	0.08†	0.12†	0.30‡
ONLINE-B	-0.26	-0.08	-0.02	-	0.02	0.02	0.03	0.07	0.11⋆	0.28‡
HUAWEITSC	-0.28	-0.10	-0.03	-0.02	-	0.00	0.01	0.05⋆	0.09†	0.27‡
MIDEIND	-0.28	-0.10	-0.04	-0.02	0.00	-	0.01	0.05⋆	0.09⋆	0.26‡
ONLINE-A	-0.29	-0.11	-0.04	-0.03	-0.01	-0.01	-	0.04	0.08	0.26‡
ALLEGRO	-0.33	-0.15	-0.08	-0.07	-0.05	-0.05	-0.04	-	0.04	0.22‡
ONLINE-Y	-0.37	-0.19	-0.12	-0.11	-0.09	-0.09	-0.08	-0.04	-	0.18‡
ONLINE-G	-0.55	-0.37	-0.30	-0.28	-0.27	-0.26	-0.26	-0.22	-0.18	-
score	0.29	0.11	0.04	0.03	0.01	0.01	0.00	-0.04	-0.08	-0.26
rank	1	2	3–7	3–8	3–7	3–7	3–9	6–9	7–9	10
bleu	**41.7**	39.8	39.2	40.6	38.4	33.5	33.6	33.3	30.1	23.7
chrF	.623	.621	.610	**.624**	.611	.578	.574	.574	.559	.492
comet	**.683**	.629	.619	.645	.604	.552	.512	.467	.422	-.071

Table 54: Head to head comparison for Icelandic→English systems

	FACEBOOK-AI	ONLINE-B	TRANSSION	ZMT	GTCOM	HUAWEITSC	MS-EGDC	P3AI	NIUTRANS	ONLINE-Y	MANIFOLD	AMU	UEDIN	TWB
FACEBOOK-AI	-	0.13‡	0.19‡	0.19‡	0.19‡	0.22‡	0.25‡	0.28‡	0.28‡	0.34‡	0.36‡	0.42‡	0.45‡	0.51‡
ONLINE-B	-0.13	-	0.06⋆	0.06	0.06	0.09†	0.12‡	0.15‡	0.15‡	0.21‡	0.23‡	0.29‡	0.32‡	0.39‡
TRANSSION	-0.19	-0.06	-	0.00	0.00	0.03	0.06	0.09†	0.09⋆	0.15‡	0.17‡	0.24‡	0.27‡	0.33‡
ZMT	-0.19	-0.06	0.00	-	0.00	0.03	0.06⋆	0.09†	0.09⋆	0.15‡	0.17‡	0.23‡	0.26‡	0.33‡
GTCOM	-0.19	-0.06	0.00	0.00	-	0.03	0.06⋆	0.09†	0.09⋆	0.15‡	0.17‡	0.23‡	0.26‡	0.33‡
HUAWEITSC	-0.22	-0.09	-0.03	-0.03	-0.03	-	0.03	0.06	0.06	0.12†	0.14‡	0.20‡	0.23‡	0.30‡
MS-EGDC	-0.25	-0.12	-0.06	-0.06	-0.06	-0.03	-	0.03	0.03	0.09⋆	0.11†	0.18‡	0.21‡	0.27‡
P3AI	-0.28	-0.15	-0.09	-0.09	-0.09	-0.06	-0.03	-	0.00	0.06	0.08⋆	0.14†	0.17‡	0.24‡
NIUTRANS	-0.28	-0.15	-0.09	-0.09	-0.09	-0.06	-0.03	0.00	-	0.06	0.08⋆	0.14‡	0.17‡	0.24‡
ONLINE-Y	-0.34	-0.21	-0.15	-0.15	-0.15	-0.12	-0.09	-0.06	-0.06	-	0.02	0.08⋆	0.12†	0.18‡
MANIFOLD	-0.36	-0.23	-0.17	-0.17	-0.17	-0.14	-0.11	-0.08	-0.08	-0.02	-	0.06	0.09⋆	0.16‡
AMU	-0.42	-0.29	-0.24	-0.23	-0.23	-0.20	-0.18	-0.14	-0.14	-0.08	-0.06	-	0.03	0.09†
UEDIN	-0.45	-0.32	-0.27	-0.26	-0.26	-0.23	-0.21	-0.17	-0.17	-0.12	-0.09	-0.03	-	0.06⋆
TWB	-0.51	-0.39	-0.33	-0.33	-0.33	-0.30	-0.27	-0.24	-0.24	-0.18	-0.16	-0.09	-0.06	-
score	0.25	0.12	0.06	0.06	0.06	0.03	0.00	-0.03	-0.03	-0.09	-0.11	-0.17	-0.20	-0.27
rank	1	2–4	3–7	2–6	3–6	3–9	5–19	6–10	6–10	8–11	10–12	11–13	12–13	14
bleu	**21.0**	18.7	18.8	18.8	17.8	17.5	17.1	17.8	16.5	13.9	16.9	14.1	14.9	12.3
chrF	**.487**	.467	.472	.472	.467	.468	.453	.463	.447	.448	.456	.413	.422	.403
comet	**.422**	.335	.345	.344	.345	.253	.148	.245	.174	.124	.127	.070	.076	-0.046

Table 55: Head to head comparison for Hausa→English systems

	GTCOM	ONLINE-B	TRANSSION	MS-EGDC	UEDIN	ONLINE-Y	HUAWEITSC	ONLINE-A	ONLINE-G
GTCOM	-	0.04	0.12‡	0.13‡	0.15‡	0.22‡	0.28‡	0.31‡	0.58‡
ONLINE-B	-0.04	-	0.08⋆	0.09⋆	0.11†	0.18‡	0.24‡	0.27‡	0.54‡
TRANSSION	-0.12	-0.08	-	0.00	0.03	0.09⋆	0.16†	0.19†	0.45‡
MS-EGDC	-0.13	-0.09	0.00	-	0.02	0.09	0.16†	0.18†	0.45‡
UEDIN	-0.15	-0.11	-0.03	-0.02	-	0.07	0.13⋆	0.16†	0.43‡
ONLINE-Y	-0.22	-0.18	-0.09	-0.09	-0.07	-	0.07	0.09	0.36‡
HUAWEITSC	-0.28	-0.24	-0.16	-0.16	-0.13	-0.07	-	0.03	0.29‡
ONLINE-A	-0.31	-0.27	-0.19	-0.18	-0.16	-0.09	-0.03	-	0.27‡
ONLINE-G	-0.58	-0.54	-0.45	-0.45	-0.43	-0.36	-0.29	-0.27	-
score	0.20	0.16	0.08	0.08	0.05	-0.01	-0.08	-0.11	-0.37
rank	1–2	1–2	3–5	3–5	3–6	4–8	6–8	6–8	9
bleu	24.2	24.1	**24.5**	21.1	21.7	21.5	21.9	21.1	16.7
chrF	**.517**	.512	.512	.486	.489	.488	.488	.483	.433
comet	**.692**	.670	.637	.532	.584	.501	.528	.494	.116

Table 56: Head to head comparison for Bengali→Hindi systems

	HUAWEITSC	ONLINE-A	GTCOM	UEDIN	ONLINE-Y	TRANSSION	ONLINE-B	MS-EGDC	ONLINE-G
HUAWEITSC	-	0.01	0.01	0.03	0.17⋆	0.20‡	0.22⋆	0.25‡	1.35‡
ONLINE-A	-0.01	-	0.00	0.02	0.16⋆	0.19‡	0.21⋆	0.24‡	1.34‡
GTCOM	-0.01	0.00	-	0.02	0.15†	0.19‡	0.20†	0.24‡	1.33‡
UEDIN	-0.03	-0.02	-0.02	-	0.13⋆	0.17‡	0.19⋆	0.22‡	1.31‡
ONLINE-Y	-0.17	-0.16	-0.15	-0.13	-	0.04⋆	0.05	0.09‡	1.18‡
TRANSSION	-0.20	-0.19	-0.19	-0.17	-0.04	-	0.02	0.05⋆	1.14‡
ONLINE-B	-0.22	-0.21	-0.20	-0.19	-0.05	-0.02⋆	-	0.04†	1.13‡
MS-EGDC	-0.25	-0.24	-0.24	-0.22	-0.09	-0.05	-0.04	-	1.09‡
ONLINE-G	-1.35	-1.34	-1.33	-1.31	-1.18	-1.14	-1.13	-1.09	-
score	0.24	0.24	0.23	0.21	0.08	0.04	0.03	-0.01	-1.10
rank	1–4	1–4	1–4	1–4	5–6	7	6–7	8	9
bleu	13.0	13.4	13.9	12.5	10.6	15.0	**15.3**	10.9	5.9
chrF	.457	.465	.471	.454	.432	.478	**.480**	.434	.364
comet	.523	.552	**.575**	.545	.386	.537	.535	.411	-0.215

Table 57: Head to head comparison for Hindi→Bengali systems

	TRANSSION	HUAWEITSC	MS-EGDC	GTCOM	ONLINE-G
TRANSSION	-	0.19‡	0.24‡	0.34‡	1.75‡
HUAWEITSC	-0.19	-	0.05	0.15†	1.56‡
MS-EGDC	-0.24	-0.05	-	0.10	1.51‡
GTCOM	-0.34	-0.15	-0.10	-	1.41‡
ONLINE-G	-1.75	-1.56	-1.51	-1.41	-
score	0.50	0.31	0.26	0.16	-1.25
rank	1	2–3	2–4	3–4	5
bleu	**14.5**	9.9	9.2	11.9	3.6
chrF	**.503**	.486	.476	.475	.361
comet	.290	**.315**	.299	.199	-.606

Table 58: Head to head comparison for Zulu→Xhosa systems

	HUAWEITSC	TRANSSION	GTCOM	MS-EGDC	FJDMATH	ONLINE-G
HUAWEITSC	-	0.04	0.09	0.19‡	0.22‡	1.47‡
TRANSSION	-0.04	-	0.05	0.14†	0.18‡	1.42‡
GTCOM	-0.09	-0.05	-	0.10⋆	0.13†	1.38‡
MS-EGDC	-0.19	-0.14	-0.10	-	0.04	1.28‡
FJDMATH	-0.22	-0.18	-0.13	-0.04	-	1.24‡
ONLINE-G	-1.47	-1.42	-1.38	-1.28	-1.24	-
score	0.33	0.29	0.24	0.14	0.11	-1.14
rank	1–3	1–3	1–3	4–5	4–5	6
bleu	**11.8**	**11.8**	11.5	9.9	9.8	3.9
chrF	**.504**	.497	.493	.477	.479	.370
comet	**.233**	.206	.192	.180	.197	-.582

Table 59: Head to head comparison for Xhosa→Zulu systems

Rank	Ave.	Ave. z	System	Comet$_A$	BLEU$_{A,B}$	BLEU$_A$	BLEU$_B$	chrF$_A$	chrF$_B$
1	90.2	0.397	HUMAN-A	–	–	–	–	–	–
2-4	87.9	0.284	HUMAN-B	–	–	–	–	–	–
2-4	87.6	0.263	Facebook-AI	**0.775**	**36.1**	**24.8**	**22.7**	**0.536**	**0.506**
2-4	86.1	0.214	Online-W	0.751	33.6	23.0	21.6	0.528	0.500
5-7	83.0	0.122	eTranslation	0.625	30.8	21.0	19.4	0.506	0.478
5-6	82.1	0.047	CUNI-Transformer2018	0.671	31.5	21.6	19.7	0.509	0.482
6-8	79.2	-0.120	CUNI-DocTransformer	0.680	32.1	22.2	19.8	0.517	0.485
7-9	79.3	-0.154	CUNI-Marian-Baselines	0.621	28.9	20.1	18.3	0.499	0.472
8-10	77.8	-0.183	Online-B	0.586	28.9	20.0	17.9	0.496	0.466
9-10	74.6	-0.308	Online-A	0.585	29.0	20.2	18.2	0.499	0.468
11	76.2	-0.373	Online-Y	0.456	26.2	18.1	16.1	0.481	0.451
12	65.6	-0.674	Online-G	0.293	22.0	15.3	13.9	0.457	0.431

Table 60: Automatic metric scores for English→Czech systems

Rank	Ave.	Ave. z	System	Comet$_A$	Comet$_C$	BLEU$_{A,C}$	BLEU$_A$	BLEU$_C$	chrF$_A$	chrF$_C$
1-17	83.3	0.266	Online-B	0.502	0.568	47.3	28.4	37.2	0.588	0.650
1-5	84.7	0.243	Online-W	0.546	0.616	51.0	29.7	41.3	0.602	0.678
1-14	86.6	0.217	WeChat-AI	0.548	0.610	**51.2**	**31.3**	40.0	**0.607**	0.668
1-6	87.6	0.145	Facebook-AI	**0.567**	**0.630**	52.5	31.3	**42.0**	0.606	**0.676**
1-10	89.4	0.116	UF	0.507	0.573	47.3	28.5	37.2	0.589	0.650
2-17	85.2	0.089	HW-TSC	0.516	0.576	48.9	29.8	38.6	0.597	0.658
3-17	86.8	0.072	UEdin	0.517	0.574	48.4	29.9	38.0	0.595	0.650
3-18	86.5	0.041	P3AI	0.498	0.560	46.3	28.3	36.5	0.584	0.639
3-18	86.4	0.030	HUMAN-A	–	0.554	–	–	–	–	–
5-19	83.3	0.013	happypoet	0.452	0.511	44.6	27.6	35.4	0.582	0.634
4-19	86.1	0.010	eTranslation	0.506	0.568	48.7	29.6	38.5	0.594	0.653
4-19	84.4	0.001	Online-A	0.511	0.573	47.6	29.0	37.9	0.594	0.653
3-18	84.5	0.001	HUMAN-C	0.540	–	–	–	–	–	–
5-19	78.8	-0.053	VolcTrans-AT	0.518	0.580	47.8	29.3	38.0	0.595	0.653
5-19	86.7	-0.055	NVIDIA-NeMo	0.531	0.592	49.8	30.0	39.2	0.598	0.660
8-21	83.1	-0.058	Manifold	0.497	0.557	47.5	29.4	37.2	0.592	0.644
4-20	84.3	-0.062	Online-G	0.439	0.497	43.4	27.1	33.5	0.577	0.627
12-20	84.5	-0.072	Online-Y	0.465	0.522	45.2	27.9	35.3	0.582	0.636
18-21	73.9	-0.130	ICL	0.196	0.246	39.0	24.5	30.4	0.552	0.595
4-20	85.0	-0.140	VolcTrans-GLAT	0.542	0.616	53.6	31.3	43.2	0.608	0.683
16-21	78.3	-0.179	nuclear_trans	0.386	0.445	44.3	27.7	34.5	0.578	0.626
22	80.0	-0.415	BUPT_rush	0.371	0.428	42.0	26.4	32.6	0.571	0.618

Table 61: Automatic metric scores for English→German systems

Rank	Ave.	Ave. z	System	Comet$_A$	BLEU$_A$	chrF$_A$
1-2	84.1	0.362	HUMAN-A	–	–	–
1-4	82.7	0.264	Facebook-AI	**0.329**	20.1	0.511
2-5	80.8	0.263	NiuTrans	0.304	19.7	**0.532**
3-6	81.2	0.175	Online-B	0.224	18.9	0.504
4-6	80.1	0.128	TRANSSION	0.228	18.9	0.504
2-6	79.2	0.124	ZMT	0.230	18.8	0.504
7-10	78.0	0.018	P3AI	0.273	**20.4**	0.517
7-10	78.7	0.006	HW-TSC	0.307	20.3	0.512
8-12	75.2	-0.026	AMU	0.092	16.2	0.465
7-10	78.8	-0.036	GTCOM	0.197	17.9	0.499
9-12	75.0	-0.128	MS-EgDC	0.086	16.1	0.465
12-15	70.2	-0.227	UEdin	-0.061	14.8	0.453
11-15	73.4	-0.243	Manifold	0.175	18.0	0.495
12-15	70.5	-0.340	TWB	0.000	17.1	0.483
11-15	67.7	-0.448	Online-Y	0.083	15.0	0.469

Table 62: Automatic metric scores for English→Hausa systems

Rank	Ave.	Ave. z	System	Comet$_A$	BLEU$_A$	chrF$_A$
1	88.1	0.872	HUMAN-A	–	–	–
2	84.5	0.594	Facebook-AI	**0.776**	**33.3**	**0.596**
3-4	68.2	0.277	NiuTrans	0.694	30.6	0.575
3-4	72.7	0.240	Manifold	0.648	28.6	0.562
5-9	75.2	0.200	Online-A	0.550	25.5	0.545
5-7	65.6	0.130	Lan-Bridge-MT	0.589	24.9	0.538
5-9	62.6	0.063	Mideind	0.542	24.3	0.531
6-9	73.9	0.026	Online-B	0.583	25.7	0.543
6-9	75.6	-0.034	HW-TSC	0.560	27.5	0.554
10	62.0	-0.236	Online-Y	0.351	22.4	0.513
11	48.7	-0.470	Allegro.eu	0.323	22.7	0.510
12	33.9	-1.082	Online-G	-0.327	12.2	0.421

Table 63: Automatic metric scores for English→Icelandic systems

Rank	Ave.	Ave. z	System	Comet$_A$	BLEU$_A$	chrF$_A$
1-2	86.4	0.430	Facebook-AI	**0.652**	46.8	0.407
1-2	85.3	0.314	HUMAN-A	–	–	–
3-5	84.2	0.266	Online-W	0.602	42.1	0.366
3-5	81.3	0.168	WeChat-AI	0.615	**46.9**	**0.404**
3-5	82.6	0.148	NiuTrans	0.619	46.2	0.399
6-8	77.8	0.017	HW-TSC	0.614	45.4	0.392
6-8	71.8	-0.042	MiSS	0.517	42.6	0.370
8-13	78.5	-0.051	Online-Y	0.386	39.5	0.341
6-10	77.8	-0.067	BUPT_rush	0.549	42.9	0.372
8-13	70.9	-0.129	Online-A	0.421	40.8	0.350
9-13	67.4	-0.184	Online-B	0.488	41.6	0.360
9-14	74.2	-0.284	ephemeraler	0.414	39.6	0.343
9-14	72.5	-0.339	capitalmarvel	0.460	41.0	0.355
12-14	70.1	-0.373	movelikeajaguar	0.379	38.5	0.334
15-16	63.5	-0.440	Illini	0.189	34.3	0.294
15-16	65.7	-0.541	Online-G	0.143	33.5	0.287

Table 64: Automatic metric scores for English→Japanese systems

Rank	Ave.	Ave. z	System	Comet$_A$	Comet$_B$	BLEU$_{A,B}$	BLEU$_A$	BLEU$_B$	chrF$_A$	chrF$_B$
1-3	86.0	0.317	HUMAN-B	0.600	–	–	–	–	–	–
1-3	83.3	0.277	Online-W	**0.664**	**0.660**	45.0	31.8	29.9	0.576	**0.571**
1-3	82.5	0.093	HUMAN-A	–	0.599	–	–	–	–	–
4-6	79.4	0.056	Online-B	0.604	0.601	43.5	29.8	29.2	0.568	0.567
4-7	75.3	0.032	Online-A	0.576	0.559	41.2	28.8	27.2	0.561	0.556
4-7	80.1	-0.001	Facebook-AI	0.650	0.644	**46.0**	**32.2**	**30.4**	**0.576**	0.571
7-10	74.5	-0.123	NiuTrans	0.512	0.510	40.5	28.4	27.1	0.546	0.543
7-10	72.3	-0.153	Manifold	0.566	0.566	41.5	29.2	27.6	0.554	0.551
7-10	75.4	-0.161	NVIDIA-NeMo	0.582	0.578	41.6	29.3	27.6	0.562	0.558
5-10	76.0	-0.180	Online-G	0.600	0.595	42.8	30.1	28.6	0.570	0.564
11	62.7	-0.541	Online-Y	0.474	0.470	37.7	25.8	25.3	0.538	0.538

Table 65: Automatic metric scores for English→Russian systems

Rank	Ave.	Ave. z	System	Comet$_A$	Comet$_B$	BLEU$_{A,B}$	BLEU$_A$	BLEU$_B$	chrF$_A$	chrF$_B$
1-3	82.5	0.325	HUMAN-B	0.427	–	–	–	–	–	–
2-14	74.9	0.284	HappyNewYear	0.468	0.403	48.0	35.7	32.1	0.300	0.278
1-7	81.2	0.250	Facebook-AI	0.499	0.425	49.9	35.9	35.3	0.343	0.331
1-8	80.0	0.216	HUMAN-A	–	0.421	–	–	–	–	–
4-19	75.3	0.164	Borderline	0.473	0.403	49.2	36.5	33.2	0.313	0.289
2-19	81.0	0.161	bjtu_nmt	0.474	0.409	46.9	34.8	32.5	0.295	0.274
3-14	75.5	0.151	Lan-Bridge-MT	0.463	0.406	44.6	32.6	31.3	0.320	0.300
4-21	79.3	0.124	BUPT_rush	0.425	0.368	44.7	33.1	31.1	0.296	0.278
2-18	79.2	0.098	NiuTrans	0.483	0.411	48.1	35.8	32.9	0.305	0.282
4-18	75.7	0.091	Machine_Translation	0.467	0.403	47.7	35.5	32.3	0.294	0.275
2-15	80.9	0.078	SMU	0.474	0.402	47.9	35.8	32.5	0.306	0.280
6-22	81.4	0.064	capitalmarvel	0.378	0.299	43.9	32.2	30.5	0.268	0.261
4-19	79.5	0.056	WeChat-AI	0.501	0.437	49.2	36.9	33.4	0.337	0.305
6-22	78.1	0.026	Online-W	0.468	0.391	44.8	33.4	30.9	0.303	0.277
7-22	75.2	0.004	ICL	0.463	0.396	47.5	34.8	33.3	0.317	0.300
9-23	75.9	-0.008	HW-TSC	0.447	0.380	47.4	35.1	32.3	0.298	0.279
5-23	78.2	-0.025	ZengHuiMT	0.448	0.386	48.5	35.9	32.6	0.304	0.282
11-22	81.2	-0.026	yyds	0.474	0.407	48.1	35.9	32.4	0.302	0.278
10-26	79.7	-0.050	P3AI	0.436	0.375	47.0	34.0	33.3	0.318	0.308
17-27	77.1	-0.061	windfall	0.395	0.313	44.2	32.6	30.3	0.282	0.269
6-24	78.9	-0.075	Online-B	0.458	0.381	48.5	36.0	33.1	0.321	0.299
13-26	76.8	-0.080	NJUSC_TSC	0.439	0.381	46.3	34.2	31.9	0.312	0.291
9-24	77.7	-0.100	MiSS	0.468	0.404	49.0	36.2	33.2	0.304	0.286
19-27	77.0	-0.101	UF	0.413	0.361	45.3	33.1	31.4	0.288	0.277
22-28	72.7	-0.123	Online-A	0.340	0.292	43.3	31.6	30.1	0.264	0.261
22-28	79.3	-0.160	happypoet	0.364	0.307	43.5	32.5	29.7	0.277	0.259
20-28	76.9	-0.185	nuclear_trans	0.428	0.361	44.7	33.4	30.5	0.284	0.261
25-29	76.4	-0.247	ephemeraler	0.382	0.311	44.0	32.6	30.2	0.287	0.273
28-31	67.5	-0.257	Online-G	0.301	0.238	43.2	31.1	29.7	0.304	0.288
29-31	67.1	-0.463	Online-Y	0.317	0.254	43.9	32.0	30.9	0.281	0.271
29-31	68.3	-0.613	movelikeajaguar	0.371	0.309	43.7	32.7	29.7	0.280	0.260

Table 66: Automatic metric scores for English→Chinese systems

Rank	Ave.	Ave. z	System	Comet$_A$	BLEU$_A$	chrF$_A$
1–5	87.7	0.088	Online-W	**0.714**	**60.4**	**0.788**
1–7	89.2	0.052	Online-A	0.566	40.6	0.670
1–4	89.5	0.035	HUMAN-A	– –	–	
2–8	85.7	0.002	LISN	0.505	37.3	0.644
1–8	86.9	-0.014	Online-B	0.576	43.8	0.689
4–10	85.0	-0.021	talp_upc	0.481	36.3	0.641
3–8	85.0	-0.064	eTranslation	0.595	40.6	0.666
7–10	84.1	-0.154	Online-G	0.454	36.9	0.653
3–10	86.6	-0.210	Online-Y	0.503	39.5	0.659
7–10	86.4	-0.229	P3AI	0.583	39.3	0.654

Table 67: Automatic metric scores for French→German systems

Rank	Ave.	Ave. z	System	Comet$_A$	BLEU$_A$	chrF$_A$
1–3	87.9	0.160	Online-B	0.544	29.7	0.584
1–3	86.5	0.126	HUMAN-A	–	–	–
3–6	83.4	0.018	Manifold	0.586	32.5	0.606
1–6	84.8	0.006	Online-W	**0.622**	29.9	0.591
3–6	84.5	0.004	Online-A	0.561	**35.7**	0.613
6–10	83.0	-0.084	Online-G	0.449	28.6	0.577
3–10	83.5	-0.148	P3AI	0.512	31.7	**0.626**
6–10	81.3	-0.149	LISN	0.426	28.1	0.563
6–10	83.7	-0.177	Online-Y	0.463	28.3	0.568
6–10	81.0	-0.190	talp_upc	0.466	27.5	0.565

Table 68: Automatic metric scores for German→French systems

B Translator Brief: Sentence-Split News Test Sets

Translator Brief

In this project we wish to translate online news articles for use in evaluation of Machine Translation (MT). The translations produced by you will be compared against the translations produced by a variety of different MT systems. They will be released to the research community to provide a benchmark, or "gold-standard" measure for translation quality. The translation therefore needs to be a high-quality rendering of the source text into the target language, as if it was news written directly in the target language. However there are some constraints imposed by the intended usage:

- All translations should be **"from scratch", without post-editing from MT**. Using post-editing would bias the evaluation, so we need to avoid it. We can detect post-editing so will reject translations that are post-edited.
- Translation should **preserve the sentence boundaries.** The source texts are provided with exactly one sentence per line, and the translations should be the same, one sentence per line.
- Translators should **avoid inserting parenthetical explanations** into the translated text and obviously **avoid losing any pieces of information** from the source text.

We will check a sample of the translations for quality, and we will check the entire set for evidence of post-editing.

The source files will be delivered as text files (sometimes known as "notepad" files), with one sentence per line. We need the translations to be returned in the same format. If you prefer to receive the text in a different format, then please let us know as we may be able to accommodate it.

C News Task System Submission Summaries

This appendix lists self-reported details on MT systems participating in the News Translation Task.

C.1 AFRL (Erdmann et al., 2021)

No brief description provided.

C.2 ALLEGRO.EU (Koszowski et al., 2021)

Allegro news translation system is based on the transformer-big architecture, it makes use of corpora filtering and backtranslation both applied to parallel and monolingual data alike.

ALLEGRO.EU	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece) Vocabulary Size: 32000 Toolkit Used: OpenNMT-py Batch size: 8192 tokens Features of your model structure: Dropout, Tied source and target word embeddings Document-level training: No document-level: Our system processes each segment independently. Number of GPUs Used Concurrently: 1x A100 Wallclock training time: 13h Number of contrastive configurations used: 4 Other comments: fp16 was used
ALLEGRO.EU	en-is	True Parallel Training Data Size in Sentence Pairs: 3935903 parallel.en-is True Parallel Training Data Size in Words: 60185218 parallel.en 55419088 parallel.is Synthetic Parallel Training Data Size in Sentence Pairs: 2953528 synt.en-is Synthetic Parallel Training Data Size in Words: 47082741 synt.en 44441374 synt.is Monolingual Training Data in Sentences: 4044137 mono.en-is Monolingual Training Data in Words: 81559107 mono.en 72315845 mono.is Processing Tools Used: Language detection (e.g. for data cleanup) Features of your model development: Data filtering, Data selection, Iterative back-translation, Oversampling Number of Systems Ensembled/Averaged: 1
ALLEGRO.EU	is-en	True Parallel Training Data Size in Sentence Pairs: 3935903 parallel.is-en True Parallel Training Data Size in Words: 55419088 parallel.is 60185218 parallel.en Synthetic Parallel Training Data Size in Sentence Pairs: 2907611 synt.is-en Synthetic Parallel Training Data Size in Words: 43642048 synt.is 47392565 synt.en Monolingual Training Data in Sentences: 3991420 mono.is-en Monolingual Training Data in Words: 78481284 mono.is 81693347 mono.en Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Features of your model development: Data filtering, Data selection, Iterative back-translation, Oversampling, Ensembling Number of Systems Ensembled/Averaged: 2

C.3 AMU (Nowakowski and Dwojak, 2021)

AMU submission for the low-resource English-Hausa language pair involved data filtering and cleaning, transfer learning from the pretrained unrelated high-resource language pair (German-English) and iterative backtranslation. The initial iteration of backtranslation was performed with a PB-SMT model, while the subsequent iterations were performed with NMT Transformer models.

C.4 BJTU-NMT (no associated paper)

No brief description provided.

C.5 BORDERLINE (Wang et al., 2021)

No brief description provided.

C.6 BUPT-RUSH **(no associated paper)**

No brief description provided.

C.7 CAPITALMARVEL **(no associated paper)**

No brief description provided.

C.8 CFILT

We train our DE-DSB system using transfer learning from DE-HSB model. Our DE-HSB model is using monolingual data of HSB and DE and train an unsupervised system first using MASS objective, then finetune it with iterative back-translation and then finetune it for translation using parallel data of DE-HSB. This system is then trained using monolingual data of DE and DSB with iterative back-translation. We use shared encoder and decoder with 6 layers in both encoder and decoder.

CFILT	common	Multilingual MT System: No.
CFILT	de-dsb	Basic System Classification: Masked sequence to sequence pretraining (Song et al 2019)+ Transfer learning Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer Vocabulary Size: 33678 True Parallel Training Data Size in Sentence Pairs: de-hsb 147521 de-dsb 0 Processing Tools Used: Tokenizer Other Processing Tools Used: fastBPE Toolkit Used: Moses, fastBPE, MASS Features of your model development: Iterative back-translation, Unsupervised (i.e. not involving parallel data), Language model pretraining with MASS objective Pre-trained parts of models: Masked Sequence to Sequence Pre-training (MASS) Document-level training: No document-level: Our system processes each segment independently. Other Features of Your Training: Transfer learning
CFILT	de-hsb	Basic System Classification: MASS pretraining (song et al) Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece), Moses Tokenizer Toolkit Used: Moses, fastBPE, MASS Pre-trained parts of models: Masked Sequence to Sequence Pre-training (MASS) Document-level training: No document-level: Our system processes each segment independently.
CFILT	dsb-de	Basic System Classification: MASS pretraining, Transfer learning Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer
CFILT	hsb-de	Basic System Classification: MASS pretraining (song et al 2019), Transfer learning Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt) Pre-trained parts of models: Masked Sequence to Sequence Pre-training (MASS)

C.9 CUNI **(Gebauer et al., 2021)**

CUNI-DOCTRANSFORMER CUNI-DocTransformer is similar to the sentence-level version called CUBBITT (Popel et al., 2020), but trained on sequences with multiple sentences of up to 3000 characters. This year, a better sentence detection and number/unit conversion post-processing have been applied.

CUNI-TRANSFORMER2018 CUNI-Transformer2018, also called CUBBITT, is exactly the same system as in WMT2018. It is the Transformer model trained according to Popel and Bojar (2018) plus a Block Back-translation (Popel et al., 2020).

CUNI	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: SubwordTextEncoder of Tensor2Tensor (as in https://github.com/tensorflow/tensor2tensor) Vocabulary Size: 32k Monolingual Training Data in Sentences: see synthetic Monolingual Training Data in Words: see synthetic Processing Tools Used: Tokenizer Toolkit Used: Tensor2Tensor Features of your model development: Data filtering, Data selection, Block-backtranslation as in Martin Popel, Marketa Tomkova, Jakub Tomek et al. (2020), Iterative back-translation, Oversampling, Averaging Features of your model structure: Dropout, Tied source and target word embeddings, Weight tying (other than word embeddings) Number of Systems Ensembled/Averaged: 8 checkpoints Wallclock training time: 8 days (without iterated backtranslation)
CUNI-DocTransformer	cs-en, en-cs	True Parallel Training Data Size in Sentence Pairs: 61000000 True Parallel Training Data Size in Words: en=617000000, cs=702000000 Synthetic Parallel Training Data Size in Sentence Pairs: en=76000000, cs=51000000 Synthetic Parallel Training Data Size in Words: en=1296000000, cs=833000000 Batch size: 1800*10 subwords Document-level training: Overlapping windows: A window is moved over segments, receiving multiple translations of each of them, with some voting or combination afterwards. Number of GPUs Used Concurrently: 10 GTX 1080 Ti Number of contrastive configurations used: 4
CUNI-Transformer2018	cs-en, en-cs	True Parallel Training Data Size in Sentence Pairs: 58000000 True Parallel Training Data Size in Words: en=642000000, cs=563000000 Synthetic Parallel Training Data Size in Sentence Pairs: en=47000000, cs=65000000 Synthetic Parallel Training Data Size in Words: en=935000000, cs=927000000 Batch size: 2900*8 subwords Document-level training: No document-level: Our system processes each segment independently. Number of GPUs Used Concurrently: 8 GTX 1080 Ti Number of contrastive configurations used: Now only one. In 2018, I trained hundreds of models on smaller data or less GPUs, as described in Training Tips for the Transformer Model (Popel and Bojar, 2018).

C.10 DIDI-NLP **(no associated paper)**

No brief description provided.

C.11 EPHEMERALER

We use Transformer big model and ensembling.

EPHEMERALER	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...)
EPHEMERALER	en-ja	Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt)
EPHEMERALER	en-zh	—

C.12 eTranslation **(Oravecz et al., 2021)**

eTranslations's En-De system is an ensemble of 4 big transformers, trained from all available parallel data (cleaned up and filtered with heuristic rules and with a language model built from the German NewsCrawl data) and with additional tagged, back-translated data generated from the monolingual news corpora. The original parallel data is upsampled to a 1:1 ratio. Each transformer model is then tuned on a 10M top subset of original parallel data scored and ranked by the monolingual news language model and then fine-tuned further on previous year's test sets. The models use a 36k SentencePiece vocabulary. The SentencePiece module as built in the Marian toolkit is used for end-to-end text processing, without the standard pre- and postprocessing steps of truecasing, or (de)tokenization.

The Fr-De system is an ensemble of 4 big transformers. Three of them are trained on original parallel (OP) data and back-translated (BT) data in a 1:1 ratio. The 4th big transformer was additionally fine-

tuned for 7 epochs on 2M of the OP data scored by a domain language model. BT data and data for the domain language model were selected using topic modelling techniques to tune the model towards the domain defined in the task.

The En-Cs system is an ensemble of two big transformer models from last year's submission, trained on the WMT 2020 data, both original parallel and back-translated. Training on the 2021 data had not finished until the submission deadline and intermediate models scored worse than the 2020 models.

ETRANSLATION	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece) Toolkit Used: Marian Document-level training: No document-level: Our system processes each segment independently.
ETRANSLATION	en-de	Vocabulary Size: 36000 True Parallel Training Data Size in Sentence Pairs: 32077088 True Parallel Training Data Size in Words: 637753194; 603406453 Synthetic Parallel Training Data Size in Sentence Pairs: 226375233 Synthetic Parallel Training Data Size in Words: 3514437534; 3007895939 Monolingual Training Data in Sentences: BT: 226375233; En LM: 133385694; De LM: 167110102; Monolingual Training Data in Words: BT: 3514437534; 3007895939 En LM: 2891767899; De LM: 3012152905 Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Batch size: 1500-5000 Features of your model development: Data filtering, Data selection, Back-translation with greedy decoding, Oversampling, Ensembling, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Other Features of Your Training: continued training on LM scored subset of OP data Number of Systems Ensembled/Averaged: 4 Number of GPUs Used Concurrently: 4-8 V100 Wallclock training time: 10 days Number of contrastive configurations used: 16 Other comments: described in the system paper
ETRANSLATION	fr-de	Vocabulary Size: 30000 True Parallel Training Data Size in Sentence Pairs: 13640043 True Parallel Training Data Size in Words: 257966051; 228953683 Synthetic Parallel Training Data Size in Sentence Pairs: 14980793 Synthetic Parallel Training Data Size in Words: 241457887; 209714902 Monolingual Training Data in Sentences: de: 11475958 Monolingual Training Data in Words: de: 160803597 Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Batch size: 1500 Features of your model development: Data filtering, Data selection, Back-translation with greedy decoding, Oversampling, Ensembling, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Number of Systems Ensembled/Averaged: 4 Number of GPUs Used Concurrently: 4 Wallclock training time: 5 days Number of contrastive configurations used: 11
ETRANSLATION	en-cs	Vocabulary Size: 36000 True Parallel Training Data Size in Sentence Pairs: 45104433 True Parallel Training Data Size in Words: cs: 559485115 en: 637004843 Synthetic Parallel Training Data Size in Sentence Pairs: 88164502 Synthetic Parallel Training Data Size in Words: cs: 1206604906 en: 1450464754 Monolingual Training Data in Sentences: 0 Monolingual Training Data in Words: 0 Processing Tools Used: Language detection (e.g. for data cleanup) Batch size: 1000 Features of your model development: Data filtering, Back-translation with sampling, Ensembling Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 2 Number of GPUs Used Concurrently: 4 Wallclock training time: 12 days Number of contrastive configurations used: 4

C.13 FACEBOOK-AI (Tran et al., 2021)

Facebook AI participated in the unconstrained track for all 14 English-centric directions. To explore the limit of scaling multilingual translation, we trained two multilingual systems: Any-to-English, and English-to-Any, and submitted them to all directions. In addition to well-known techniques such as large scale backtranslation, in-domain finetuning, ensembling, and noisy channel re-ranking, we also experimented with scaling dense transformer (up to 4.7B parameters), and sparse mixture of experts (up to 52B parameters)

FACEBOOK-AI	common	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt)
FACEBOOK-AI	cs-en, de-en, ha-en, is-en, ja-en, ru-en, zh-en	Vocabulary Size: 128000 True Parallel Training Data Size in Sentence Pairs: (This includes mined data from CCMatrix and CCAligned) cs-en 163,005,937 de-en 544,549,887 ha-en 1,176,367 is-en 20,632,971 ja-en 141,399,044 ru-en 276,805,988 zh-en 163,188,501 Total 1,310,758,695 True Parallel Training Data Size in Words: (This includes mined data from CCMatrix and CCAligned) 2725979073 train.cs_en.cs 2661179726 train.cs_en.en 10546303763 train.de_en.de 9692849751 train.de_en.en 20466571 train.ha_en.ha 18786730 train.ha_en.en 342802801 train.is_en.is 301337746 train.is_en.en 640041697 train.ja_en.ja 1907474016 train.ja_en.en 4896618898 train.ru_en.ru 4887514242 train.ru_en.en 714086693 train.zh_en.zh 2853757236 train.zh_en.en Synthetic Parallel Training Data Size in Sentence Pairs: (Backtranslation data) cs-en 428,914,158 de-en 394,678,147 ha-en 378,439,788 is-en 428,581,678 ja-en 428,227,231 ru-en 381,863,501 zh-en 432,017,983 Total 2,872,722,486 Monolingual Training Data in Sentences: Similar to backtranslation data (430M English sentences) Processing Tools Used: Language detection (e.g. for data cleanup) Toolkit Used: fairseq(-py) Batch size: 1M tokens Features of your model development: Data filtering, Iterative back-translation, Ensembling, Averaging, Right-to-left reranking, Target-to-source reranking, Fine-tuning for domain adaptation, Mixture of Experts Features of your model structure: Dropout, Tied source and target word embeddings Document-level training: No document-level: Our system processes each segment independently. Other Features of Your Training: In-domain parallel data mining Number of Systems Ensembled/Averaged: 3 Number of GPUs Used Concurrently: 128 Wallclock training time: 1 week Number of contrastive configurations used: 5 different architectures, 3-4 training iterations each
FACEBOOK-AI	en-cs, en-de, en-ha, en-is, en-ja, en-ru, en-zh	Vocabulary Size: 128000 True Parallel Training Data Size in Sentence Pairs: (Includes mined data from CCMatrix, CCAligned) en-cs 163,758,080 en-de 546,657,024 en-ha 995,860 en-is 27,228,288 en-ja 142,843,968 en-ru 277,540,224 en-zh 163,774,144 Total 1,322,797,588 Synthetic Parallel Training Data Size in Sentence Pairs: en-cs 140,172,928 en-de 237,235,904 en-ha 6,719,488 en-is 101,139,008 en-ja 218,456,960 en-ru 163,223,744 en-zh 123,211,776 Total 990,159,808 Monolingual Training Data in Sentences: Same as backtranslation Processing Tools Used: Language detection (e.g. for data cleanup) Toolkit Used: fairseq(-py) Batch size: 1M tokens per batch Features of your model development: Data filtering, Data selection, Iterative back-translation, Oversampling, Ensembling, Averaging, Right-to-left reranking, Target-to-source reranking, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Document-level training: No document-level: Our system processes each segment independently. Number of Systems Ensembled/Averaged: 2-3 Number of GPUs Used Concurrently: 128 Wallclock training time: 1 week Number of contrastive configurations used: 20

C.14 FJDMATH (Martinez, 2021)

No brief description provided.

C.15 GTCOM (Bei and Zong, 2021)

No brief description provided.

C.16 HAPPYNEWYEAR (no associated paper)

No brief description provided.

C.17 HAPPYPOET (no associated paper)

No brief description provided.

C.18 HW-TSC (Wei et al., 2021)

We participate in 7 language pairs including Zh/En, De/En, Ja/En, Ha/En, Is/En, Hi/Bn, and Xh/Zu and in both directions under the constrained condition. We use the standard Transformer-Big model as the baseline and obtain the best performance via two variants with larger parameter sizes. We perform detailed pre-processing and filtering on the provided large-scale bilingual and monolingual datasets. Several commonly used strategies are used to train our models such as Back Translation, Ensemble Knowledge Distillation, etc. We also conduct experiments regarding similar language augmentation, which lead to positive results, although not used in our submission. Our submission obtains competitive results in the final evaluation.

HW-TSC	common	Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Document-level training: No document-level: Our system processes each segment independently. Number of GPUs Used Concurrently: 8
HW-TSC	en-zh	Multilingual MT System: No. Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer, jieba Vocabulary Size: 32k True Parallel Training Data Size in Sentence Pairs: 16.5M Synthetic Parallel Training Data Size in Sentence Pairs: 316.5M Monolingual Training Data in Sentences: 300M Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup), Jieba word segmentation for Chinese Toolkit Used: Marian, fairseq(-py), Moses Batch size: 4096 Features of your model development: Data filtering, Data selection, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 2Ensembled
HW-TSC	zh-en	Multilingual MT System: No. Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer, jieba Vocabulary Size: 32k True Parallel Training Data Size in Sentence Pairs: 16.5M Synthetic Parallel Training Data Size in Sentence Pairs: 316.5M Monolingual Training Data in Sentences: 300M Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py), Moses Batch size: 4096 Features of your model development: Data filtering, Data selection, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 2ensemble

HW-TSC	en-ha	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece) Vocabulary Size: 32K True Parallel Training Data Size in Sentence Pairs: 0.6M Synthetic Parallel Training Data Size in Sentence Pairs: 14.9M Monolingual Training Data in Sentences: 14.3M Processing Tools Used: Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Features of your model development: Data filtering, Data selection, Back-translation with greedy decoding, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 4ensemble
HW-TSC	ha-en	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Vocabulary Size: 32K True Parallel Training Data Size in Sentence Pairs: 0.6M Synthetic Parallel Training Data Size in Sentence Pairs: 14.9M Monolingual Training Data in Sentences: 14.3M Processing Tools Used: Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Features of your model development: Data filtering, Data selection, Back-translation with greedy decoding, Iterative back-translation, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 4
HW-TSC	en-is	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece) Vocabulary Size: 32K True Parallel Training Data Size in Sentence Pairs: 4M Synthetic Parallel Training Data Size in Sentence Pairs: 42M Monolingual Training Data in Sentences: 38M Processing Tools Used: Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: 4096 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with greedy decoding, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 3
HW-TSC	is-en	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece) Vocabulary Size: 32K True Parallel Training Data Size in Sentence Pairs: 4M Synthetic Parallel Training Data Size in Sentence Pairs: 42M Monolingual Training Data in Sentences: 38M Processing Tools Used: Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Features of your model development: Data filtering, Data selection, Back-translation with greedy decoding, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 3

HW-TSC	bn-hi	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: sentencepiece Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 3400000 Synthetic Parallel Training Data Size in Sentence Pairs: 46500000 Monolingual Training Data in Sentences: 46500000 Monolingual Training Data in Words: 1899414973 Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: 1500 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling Number of Systems Ensembled/Averaged: 4
HW-TSC	hi-bn	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: sentencepiece Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 3400000 Synthetic Parallel Training Data Size in Sentence Pairs: 50000000 Monolingual Training Data in Sentences: 50000000 Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: 1500 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling, Ensembling, Averaging Number of Systems Ensembled/Averaged: 4
HW-TSC	xh-zu	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: sentencepiece Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 67000 Synthetic Parallel Training Data Size in Sentence Pairs: 12000000 Monolingual Training Data in Sentences: 12000000 Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: 1500 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling, Ensembling, Averaging, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 4
HW-TSC	zu-xh	Multilingual MT System: Yes, the system was trained and used jointly for all the language pairs. Token Unit Type Used: sentencepiece Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 67000 Synthetic Parallel Training Data Size in Sentence Pairs: 12000000 Synthetic Parallel Training Data Size in Words: 50000000 Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++) Toolkit Used: Marian, fairseq(-py) Batch size: 1500 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling, Ensembling, Averaging Number of Systems Ensembled/Averaged: 4

HW-TSC	en-ja	Multilingual MT System: No. Token Unit Type Used: sentencepiece Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 14000000 Synthetic Parallel Training Data Size in Sentence Pairs: 80000000 Monolingual Training Data in Sentences: 150000000 Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: 1500 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling, Ensembling, Averaging, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 4
HW-TSC	ja-en	Multilingual MT System: No. Token Unit Type Used: sentencepiece Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 12000000 Synthetic Parallel Training Data Size in Sentence Pairs: 80000000 Monolingual Training Data in Sentences: 150000000 Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: 1500 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling, Ensembling, Averaging, Right-to-left reranking, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 4
HW-TSC	en-de	Multilingual MT System: No. Token Unit Type Used: Moses Tokenizer, spm Vocabulary Size: 32k True Parallel Training Data Size in Sentence Pairs: 79M Synthetic Parallel Training Data Size in Sentence Pairs: 300M Monolingual Training Data in Sentences: en 300M,de 300M Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py), Moses Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 4 ensembled, 3 averaged. Wallclock training time: max_token=500000, max_step=50000
HW-TSC	de-en	Multilingual MT System: No. Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece), Moses Tokenizer Vocabulary Size: 32K True Parallel Training Data Size in Sentence Pairs: 79M Synthetic Parallel Training Data Size in Sentence Pairs: 300M Monolingual Training Data in Sentences: en 300M, de 300M+ Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Toolkit Used: Marian, fairseq(-py) Batch size: max_token=500000 Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of Systems Ensembled/Averaged: ensembled: 4, average: 3 Wallclock training time: step 50000

C.19 ICL **(no associated paper)**

No brief description provided.

C.20 IICT-YVERDON

IICT-Yverdon presents the systems submitted by our team from the Institute of ICT (HEIG-VD / HES-SO) to the Unsupervised MT and Very Low Resource Supervised MT task. We first study a baseline system using a Transformer architecture, using the Upper Sorbian (HSB) / German data from the 2020 edition of the task. We quantify the improvements brought by additional techniques such as back-translation of large German corpora and parent-language initialization using Czech-German data, and show that each of these is beneficial, and helps to reach scores that are comparable to more sophisticated systems from the 2020 task. We then present the application of this system to the 2021 task for low-resource supervised HSB-DE translation, in both directions. Finally, we present a contrastive system for HSB-DE in both directions, and for unsupervised German to Lower Sorbian (DSB) translation, which uses multi-task training with various training schedules to improve over the baseline. More specifically, we present a baseline system using a Transformer architecture, which uses back-translation of large German corpora and parent-language initialization using Czech-German data. We submit translations from this system for low-resource supervised HSB-DE, in both directions. We also present a contrastive system that makes use as well of back-translation and Czech-German initialization, and also multi-task training, in which we first train Czech-German systems by giving them different denoising tasks, together with translation, in increasing order of complexity. Afterwards, we first present the child systems with denoising tasks, and later introduce translation. Finally, we train different models with some changes in their training setups that we use for ensembling, in order to maximize diversity among the models.

C.21 IIE-MT **(no associated paper)**

No brief description provided.

C.22 ILLINI **(Le et al., 2021)**

Illini team presents an end-to-end NMT pipeline for the Japanese ↔ English news translation task using Transformer models and techniques such as politeness and formality tagging, back-translation, model ensembling, and n-best reranking to improve our translation systems.

C.23 KWAINLP **(no associated paper)**

No brief description provided.

C.24 LAN-BRIDGE-MT **(no associated paper)**

No brief description provided.

C.25 LISN **(Xu et al., 2021)**

LISN's systems for DE↔FR use Transformer-big model with the "priming" based on a prior retrieval step, which looks for similar sentences (in source and target) to prime a similar translation. These techniques aim to perform some unsupervised domain transfer, which is one of the challenge of this task. Our system only uses the data provided for the task (bilingual and backtranslated monolingual data) and are thus constrained submissions. They are built using the fairseq toolkit.

LISN	de-fr, fr-de	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Toolkit Used: fairseq(-py) Batch size: 4096

C.26 MACHINE-TRANSLATION (no associated paper)
No brief description provided.

C.27 MANIFOLD (no associated paper)
No brief description provided.

C.28 MIDEIND (Jónsson et al., 2021)
We fine-tuned a sentence-level mBART25 model on the en-is/is-en translation task using a filtered version of the ParIce parallel corpus and a back-translated corpus of roughly 30 million sentence pairs per translation direction. The back-translated corpus was generated via iterative back-translation using a Transformer-base model and a final iteration using the mBART25 translation model. Miðeind is an Icelandic startup company focusing on NLP and AI applications for the Icelandic language.

C.29 MISS (Li et al., 2021b)
No brief description provided.

C.30 MOVELIKEAJAGUAR (no associated paper)
No brief description provided.

C.31 MS-EGDC (Hendy et al., 2021)
We develop NMT for low resource language pairs Bengali to/from Hindi, English to/from Hausa and Xhosa to/from Zulu. We use constrained resources provided by the organizers. The main idea is to train a multi-lingual model with a multi-task objective using both parallel and monolingual data. This model is then used to forward and backward translate monolingual and parallel data (the latter is known as knowledge distillation). The resulting synthetic data is then used to train bilingual MT models for each language pair. The best multi-lingual and multi-task models are then combined with the best bilingual model for each pair using a novel transformer-based method.

C.32 NIUTRANS (Zhou et al., 2021)
No brief description provided.

C.33 NJUSC-TSC (no associated paper)
No brief description provided.

C.34 NUCLEAR-TRANS (no associated paper)
No brief description provided.

C.35 NVIDIA-NEMO (Subramanian et al., 2021)
No brief description provided.

C.36 P3AI (Zhao et al., 2021)
No brief description provided.

C.37 SMU **(no associated paper)**

No brief description provided.

C.38 TALP-UPC **(Escolano et al., 2021)**

No brief description provided.

C.39 TRANSSION

This paper describes the submission systems of TRANSSION for WMT21 . We participated in 6 translation directions including Hindi ↔ Bengali, Zulu ↔ Xhosa and English ↔ Hausa in both directions. 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 Multi-Lingual Training, Back Translation, In-domain Finetuning, Transfer Learning, ensemble and Reranking.

TRANSSION	common	Multilingual MT System: No. Token Unit Type Used: Custom Tokenizer, BPE (as in https://github.com/rsennrich/subword-nmt) Vocabulary Size: 50000 Processing Tools Used: Tokenizer, Shallow Dependency Parser (UD), Shallow Consituency Parser, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Batch size: 6144 Document-level training: No document-level: Our system processes each segment independently. Number of Systems Ensembled/Averaged: 5 Number of GPUs Used Concurrently: 1
TRANSSION	bn-hi	Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...), Hybrid Monolingual Training Data in Sentences: 44,035,924 Monolingual Training Data in Words: 329,604,211,372,512,000 Toolkit Used: Custom in Tensorflow, Custom in Keras (whatever is below it) Features of your model development: Data filtering, Data selection, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Extra languages used beyond those listed above (e.g. some form of pivoting or multi-lingual training), Ensembling, Averaging, Right-to-left reranking, Target-to-source reranking, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings, Residual adapters Pre-trained parts of models: Pre-trained word embeddings Wallclock training time: 12hours Number of contrastive configurations used: 15
TRANSSION	xh-zu, zu-xh, bn-hi, hi-bn, ha-en, en-ha	Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Toolkit Used: Custom in Tensorflow Features of your model development: Data filtering, Data selection, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Oversampling, Extra languages used beyond those listed above (e.g. some form of pivoting or multi-lingual training), Ensembling, Averaging, Right-to-left reranking, Target-to-source reranking, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Wallclock training time: 12 hours

C.40 TWB

We developed a bidirectional transformer-based system for Hausa-English news translation task. In our paper we give an overview of the data available including the 15,000 hand-crafted parallel dataset which was created internally. Our best systems achieved 17.1 and 12.3 BLEU on EN-HA and HA-EN directions on the task test sets, respectively.

TWB	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt) Vocabulary Size: 50,000 True Parallel Training Data Size in Sentence Pairs: 806345 True Parallel Training Data Size in Words: 10697192(en), 11405851(ha) Toolkit Used: OpenNMT-py Batch size: 4096 tokens Features of your model development: Data filtering, Data selection, Back-translation with sampling, Ensembling, Averaging, Fine-tuning for domain adaptation Features of your model structure: Dropout Document-level training: No document-level: Our system processes each segment independently. Number of Systems Ensembled/Averaged: Averaged up to 8 models Number of GPUs Used Concurrently: 2 Number of contrastive configurations used: 1
TWB	en-ha	Synthetic Parallel Training Data Size in Sentence Pairs: 567231 Synthetic Parallel Training Data Size in Words: 25495541(ha), 23815542(en) Monolingual Training Data in Sentences: Only the 567231 sentence dataset that were machine translated to make synthetic data Monolingual Training Data in Words: 25495541 Wallclock training time: 24 hours
TWB	ha-en	Synthetic Parallel Training Data Size in Sentence Pairs: 1,000,000 Synthetic Parallel Training Data Size in Words: 11442297(en), 13188160(ha) Monolingual Training Data in Sentences: Only the 1,000,000 sentence dataset that were machine translated to make synthetic data Monolingual Training Data in Words: 11442297 Wallclock training time: 36 to 48 hours

C.41 UEDIN (Chen et al., 2021; Pal et al., 2021)

UEdin's bn-hi and hi-bn systems use models trained on constrained parallel data to back-translate all of the provided monolingual data. New transformer models are then pre-trained on back-translated data, and fine-tuned on parallel data. A second stage of fine-tuning is done on training data that is in-domain, which is extracted in a number of ways, including n-gram matching, TF-IDF similarity, and language model scoring with the validation set. Finally, multiple models fine-tuned in different ways are ensembled to generate the final translations.

UEdin's approach to de↔en started with rule-based and dual conditional cross-entropy filtering of the provided corpora. All models were trained on a mix of parallel and back-translated data, and further trained on parallel sentences only. Specifically for en→de, we trained the model on additional title-cased sentences. The models were then fine-tuned on previous WMT test sets. We ensembled 5 models for en→de and 6 for de→en. During inference, each test instance was split at sentence-level, translated, and then concatenated.

UEDIN	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece) Toolkit Used: Marian Document-level training: No document-level: Our system processes each segment independently. Number of GPUs Used Concurrently: 4

UEDIN	bn-hi	Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 2036669 True Parallel Training Data Size in Words: 24797974 Synthetic Parallel Training Data Size in Sentence Pairs: 248828890 Synthetic Parallel Training Data Size in Words: hi (monolingual, target side): 4368794315 bn (back-translated, source side): 3287105444 Monolingual Training Data in Sentences: 248828890 Monolingual Training Data in Words: 4368794315 Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Other Processing Tools Used: Sentence splitter Batch size: Dynamic Features of your model development: Data filtering, Data selection, Ensembling, Fine-tuning for domain adaptation, Back-translation with beam search Number of Systems Ensembled/Averaged: 5 Wallclock training time: 40 (6 * 4 for model ensemble for back-translation + the rest for the final model) Number of contrastive configurations used: 30
UEDIN	hi-bn	Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 2036669 True Parallel Training Data Size in Words: 24797974 Synthetic Parallel Training Data Size in Sentence Pairs: 59736357 Synthetic Parallel Training Data Size in Words: bn (monolingual, target side): 873200873 hi (back-translated, source side): 1044281945 Monolingual Training Data in Sentences: 59736357 Monolingual Training Data in Words: 873200873 Processing Tools Used: Tokenizer, Language detection (e.g. for data cleanup) Other Processing Tools Used: Sentence splitter Batch size: Dynamic Features of your model development: Data filtering, Data selection, Forward translation for synthetic data, Ensembling, Fine-tuning for domain adaptation, Back-translation with beam search Number of Systems Ensembled/Averaged: 8 Wallclock training time: 50 (8 * 4 for model ensemble for back-translation + the rest for the final model) Number of contrastive configurations used: 30
UEDIN	de-en	Vocabulary Size: 32k True Parallel Training Data Size in Sentence Pairs: 66530788 Synthetic Parallel Training Data Size in Sentence Pairs: 91033109 Processing Tools Used: Language detection (e.g. for data cleanup) Other Processing Tools Used: fastText for language identification Features of your model development: Data filtering, Back-translation with greedy decoding, Back-translation with sampling, Ensembling, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Pre-trained parts of models: Did not use Number of Systems Ensembled/Averaged: 6 Number of contrastive configurations used: N/A
UEDIN	en-de	Vocabulary Size: 32k True Parallel Training Data Size in Sentence Pairs: 66530788 Synthetic Parallel Training Data Size in Sentence Pairs: 146216106 Processing Tools Used: Language detection (e.g. for data cleanup) Other Processing Tools Used: fastText for language identification Features of your model development: Data filtering, Back-translation with greedy decoding, Back-translation with sampling, Ensembling, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Pre-trained parts of models: did not use Number of Systems Ensembled/Averaged: 5 Wallclock training time: 274 hours Number of contrastive configurations used: N/A

C.42 UF **(no associated paper)**

No brief description provided.

C.43 VOLCTRANS **(Qian et al., 2021)**

VOLCTRANS-AT	VolcTrans-AT's submission is described in the respective paper (Qian et al., 2021).

VOLCTRANS-GLAT VolcTrans-GLAT's submission is a non-autoregressive model equipped with our recent technique of "glancing transformer" (Qian et al., 2020, to appear in ACL 2021).

VOLCTRANS	common	Multilingual MT System: No. True Parallel Training Data Size in Sentence Pairs: 75M Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Document-level training: No document-level: Our system processes each segment independently.
VOLCTRANS-AT	de-en	Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer Vocabulary Size: 12000 Synthetic Parallel Training Data Size in Sentence Pairs: 110M Monolingual Training Data in Sentences: 0 Other Processing Tools Used: n/a Toolkit Used: fairseq(-py), Custom in Pytorch, Custom in Keras (whatever is below it), Moses Batch size: 125k-256k Features of your model development: Data filtering, Data selection, Knowledge distillation, Iterative back-translation, Forward translation for synthetic data, Ensembling, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Number of Systems Ensembled/Averaged: 9 Number of GPUs Used Concurrently: 16 Wallclock training time: 2 days Other comments: 3
VOLCTRANS-GLAT	de-en	Basic System Classification: Non-Autoregressive Transformer Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece), Moses Tokenizer Vocabulary Size: 32000 Synthetic Parallel Training Data Size in Sentence Pairs: 100M Monolingual Training Data in Sentences: 0 Toolkit Used: fairseq(-py), Custom in Pytorch, Moses Batch size: 256k Features of your model development: Data filtering, Data selection, Knowledge distillation, Iterative back-translation, Forward translation for synthetic data, Ensembling, Right-to-left reranking Features of your model structure: Dropout Number of Systems Ensembled/Averaged: 3 Number of GPUs Used Concurrently: 32 Wallclock training time: 3 days Number of contrastive configurations used: 6
VOLCTRANS-AT	en-de	Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt), Moses Tokenizer Vocabulary Size: 12000 Synthetic Parallel Training Data Size in Sentence Pairs: 110M Monolingual Training Data in Words: 0 Toolkit Used: fairseq(-py), Custom in Pytorch, Custom in Keras (whatever is below it), Moses Batch size: 125k-256k Features of your model development: Data filtering, Data selection, Knowledge distillation, Iterative back-translation, Forward translation for synthetic data, Ensembling Features of your model structure: Dropout, Tied source and target word embeddings Number of Systems Ensembled/Averaged: 3 Number of GPUs Used Concurrently: 16 Wallclock training time: 3 days Number of contrastive configurations used: 3
VOLCTRANS-GLAT	en-de	Basic System Classification: Non-Autoregressive Transformer Token Unit Type Used: Unigram (as in https://github.com/google/sentencepiece), Moses Tokenizer Vocabulary Size: 32000 Synthetic Parallel Training Data Size in Sentence Pairs: 100M Monolingual Training Data in Sentences: 0 Toolkit Used: fairseq(-py), Custom in Pytorch Batch size: 256k Features of your model development: Data filtering, Data selection, Knowledge distillation, Iterative back-translation, Fine-tuning for domain adaptation Features of your model structure: Dropout Number of GPUs Used Concurrently: 32 Wallclock training time: 3 days Number of contrastive configurations used: 6

C.44 WATERMELON

We only truly participated de-en direction using constraint settings. For other directions, we submit results from online translators (mainly from DeepL) just in order to see the performance.

WATERMELON	de-en	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt) Vocabulary Size: 32000 True Parallel Training Data Size in Sentence Pairs: 45M Synthetic Parallel Training Data Size in Sentence Pairs: 65M Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Other Processing Tools Used: Truecaser Toolkit Used: fairseq(-py) Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with greedy decoding, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Ensembling, Averaging, Right-to-left reranking, Target-to-source reranking, Fine-tuning for domain adaptation Features of your model structure: Dropout, Tied source and target word embeddings Number of Systems Ensembled/Averaged: 15

C.45 WECHAT-AI (Zeng et al., 2021)

We have participated in the WMT 2021 shared news translation task on English-to-Chinese, English-to-Japanese, Japanese-to-English and English-to-German. Our systems are based on the Transformer (Vaswani et al., 2017) with some effective variants, such as mixed-aan model, dual-attention model, weighted-aan model, talking-heads attention model, etc. In our experiments, we employ data selection, several synthetic data generation approaches, advanced finetuning approaches and self-bleu based model ensemble. Our constrained systems achieve 36.9, 46.9, 27.8 and 31.3 case-sensitive BLEU scores on English-to-Chinese, English-to-Japanese, Japanese-to-English and English-to-German, respectively. The BLEU scores of English-to-Chinese, English-to-Japanese and Japanese-to-English are the highest among all submissions, and that of English-to-German ranks the second. Additionally, one of our submissions on English-to-Chinese also achieves the highest chrF score 0.344.

WECHAT-AI	common	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: BPE (as in https://github.com/rsennrich/subword-nmt) Processing Tools Used: Tokenizer, Word Aligner (e.g. fast_align or GIZA++), Language detection (e.g. for data cleanup) Batch size: 65536 tokens Features of your model structure: Dropout Document-level training: No document-level: Our system processes each segment independently.
WECHAT-AI	en-de	Toolkit Used: fairseq(-py) Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Forward translation for synthetic data, Ensembling, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 6
WECHAT-AI	en-ja	Vocabulary Size: en: 34981, ja: 48519 True Parallel Training Data Size in Sentence Pairs: 12339352 True Parallel Training Data Size in Words: en: 310739662, ja: 379286579 Toolkit Used: OpenNMT-py Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Ensembling, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 8

WECHAT-AI	ja-en	Vocabulary Size: en: 34981, ja: 48519 True Parallel Training Data Size in Sentence Pairs: 12339352 True Parallel Training Data Size in Words: en: 310739662, ja: 310739662 Toolkit Used: OpenNMT-py Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Forward translation for synthetic data, Ensembling, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 15
WECHAT-AI	en-zh	Vocabulary Size: en: 38038, zh: 47038 True Parallel Training Data Size in Sentence Pairs: 31076375 True Parallel Training Data Size in Words: en: 784141085, zh: 749465141 Toolkit Used: fairseq(-py) Features of your model development: Data filtering, Data selection, Knowledge distillation, Back-translation with sampling, Iterative back-translation, Forward translation for synthetic data, Ensembling, Fine-tuning for domain adaptation Number of Systems Ensembled/Averaged: 4

C.46 WINDFALL (no associated paper)

No brief description provided.

C.47 XMU (no associated paper)

No brief description provided.

C.48 YYDS (no associated paper)

No brief description provided.

C.49 ZENGHUIMT (Zeng, 2021)

No brief description provided.

ZENGHUIMT	en-zh, zh-en	Multilingual MT System: No. Basic System Classification: Seq2seq Transformer Style [Vaswani+2017] (self-attention, ...) Token Unit Type Used: Custom Tokenizer, BPE (as in https://github.com/rsennrich/subword-nmt) Vocabulary Size: 45467 True Parallel Training Data Size in Sentence Pairs: 5600583 True Parallel Training Data Size in Words: 88573016 Synthetic Parallel Training Data Size in Sentence Pairs: 23428568 Monolingual Training Data in Sentences: 23428568 Toolkit Used: THUMT Batch size: 15000 Features of your model development: Data filtering, Data selection, Iterative back-translation, Ensembling Features of your model structure: Dropout, Tied source and target word embeddings Document-level training: No document-level: Our system processes each segment independently. Number of Systems Ensembled/Averaged: 4 Number of GPUs Used Concurrently: 1 Wallclock training time: three days

C.50 ZMT (no associated paper)

No brief description provided.

Findings of the WMT 2021 Shared Task on Large-Scale Multilingual Machine Translation

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, Francisco Guzmán

Facebook AI

{guw,vishrav,angelafan,sahir}@fb.com

{namangoyal,somyaj,dkiela,tthrush,fguzman}@fb.com

Abstract

We present the results of the first task on Large-Scale Multilingual Machine Translation. The task consists on the many-to-many evaluation of a single model across a variety of source and target languages. This year, the task consisted on three different settings: (*i*) SMALL-TASK1 (Central/South-Eastern European Languages), (*ii*) the SMALL-TASK2 (South East Asian Languages), and (*iii*) FULL-TASK (all 101 x 100 language pairs). All the tasks used the FLORES-101 datasetas the evaluation benchmark. To ensure the longevity of the dataset, the test sets were not publicly released and the models were evaluated in a controlled environment on Dynabench.There were a total of 10 participating teams for the tasks, with a total of 151 intermediate model submissions and 13 final models. This year's result show a significant improvement over the known baselines with +17.8 BLEU for SMALL-TASK2, +10.6 for FULL-TASK and +3.6 for SMALL-TASK1.

1 Introduction

Despite recent advances in translation quality for a handful of languages and domains, MT systems still perform poorly on *low-resource languages*. Yet, most of the world's population speak low-resource languages and would benefit from improvements in translation quality on their native languages. As a result, the field has been shifting focus towards the evaluation of MT in low-resource situations (Thu et al., 2016; Guzmán et al., 2019; Barrault et al., 2020; ∀ et al., 2020; Ebrahimi et al., 2021; Kuwanto et al., 2021). However, these efforts have had poor coverage of low-resource languages which limits our understanding on generalization.More importantly, there has been little focus on the evaluation of true many-to-many multilingual models, which hampers the progress of the field despite all the recent excitement on this research direction (Fan et al., 2020).

The recent release of the FLORES-101 (Goyal et al., 2021) benchmark made possible to evaluate massively multilingual systems in a consistent way. The benchmark consists of 3001 sentences sampled from English Wikipedia and professionally translated in 101 languages. This poses a unique opportunity to understand translation across many languages with varied typology, resources, etc.

In this first multilingual large-scale shared task, we use the FLORES-101 benchmark to evaluate the progress on massively multilingual translation, where the evaluation is performed in a non-English-centric way. We propose 3 different tasks: two small tasks involving translation between 6 languages each (30 pairs), and a large task involving the translation across 101 languages (10K pairs). In the remainder of this paper, we describe the task setup, the participants, and the official results for the task. We also analyze the results to understand better the languages for which progress has been attained, and those where a gap in quality is still observed. Finally, we propose future directions for other tasks in the future.

2 Shared tasks

In this section, we briefly describe each of the tasks, the data, the baselines and metric used for evaluation.

2.1 Languages

The languages and statistics for the languages in the small tasks can be observed in Table 1, while the statistics for the complete set of languages in the full task can be obtained in Goyal et al. (2021).

SMALL-TASK1 - This task consisted of English and Central and South-Eastern European Languages: Croatian, Estonian, Hungarian, Macedonian, Serbian. These languages were chosen by their low availability of resources, geographical

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 89–99

ISO 639-3	Language	Family	Subgrouping	Script	Bitext w/ En	Mono Data
SMALL-TASK1						
hrv	**Croatian**	Indo-European	Balto-Slavic	Latin	42.2K	144M
est	**Estonian**	Uralic	Uralic	Latin	4.82M	46M
hun	**Hungarian**	Uralic	Uralic	Latin	16.3M	385M
mkd	**Macedonian**	Indo-European	Balto-Slavic	Cyrillic	1.13M	28.8M
srp	**Serbian**	Indo-European	Balto-Slavic	Cyrillic	7.01M	35.7M
SMALL-TASK2						
ind	**Indonesian**	Austronesian	Austronesian	Latin	39.1M	1.05B
jav	**Javanese**	Austronesian	Austronesian	Latin	1.49M	24.4M
msa	**Malay**	Austronesian	Austronesian	Latin	968K	77.5M
tam	**Tamil**	Dravidian	Dravidian	Tamil	992K	68.2M
tgl	**Filipino** (Tagalog)	Austronesian	Austronesian	Latin	70.6K	107M

Table 1: **Languages in each of the small tasks.** We include the ISO 639-3 code, the language family, and script. We also include the amount of resources available in OPUS as reported by Goyal et al. (2021)

proximity, language family diversity (Balto-Slavic, Uralic and Germanic), and different scripts.

SMALL-TASK2 This task consisted of English and South-Eastern Asian languages: Javanese, Indonesian, Malay, Filipino (Tagalog) and Tamil. These were chosen by their low-resource nature, geographical proximity and relatedness to a high-resource language (Indonesian).

FULL-TASK This task consisted of all 101 languages in the FLORES-101 benchmark, including English.

2.2 The evaluation data

The original sentences in FLORES-101 were sourced in English, from a broad group of topics that could be of general interest regardless of the native language of the reader. The sentences were sampled equally from *Wikinews*, *Wikijunior* and *WikiVoyage* by selecting an article randomly from each domain, and then selecting 3 to 5 contiguous sentences (not considering segments with very short or malformed sentences).

All source sentences were sent to a Language Service Provider (LSP) for translation into 101 languages. After that, the data was sent to different translators within the LSP for editing and quality assessment which then moved on to an automated quality control setup to ensure that the translation quality score was at least 90 on a scale of 0-100.

2.3 The baselines

Fan et al. (2021) worked on creating a Many-to-Many translation model, but it did not have the full coverage of languages in FLORES-101. Hence, we used the extended model trained in Goyal et al. (2021) which was supplemented with OPUS data and extended to 124 total languages. We trained two different sizes of models with 615M and 175M parameters.

2.4 Evaluation Metric

Automatically evaluating translation quality using BLEU is suboptimal as it relies on n-gram overlap which is heavily dependent on the particular tokenization used. The challenge of making BLEU comparable by using equivalent tokenization schemes has been partially addressed by `sacrebleu` (Post, 2018). Ideally, the automatic evaluation process should be robust, simple and can be applied to any language without the need to specify any particular tokenizer, as this will make it easier for researchers to compare against each other.

Towards this goal, we trained a SentencePiece (SPM) tokenizer (Kudo and Richardson, 2018) with 256K tokens using the CC100 monolingual data[1] (Conneau et al., 2020; Wenzek et al., 2020) from all the FLORES-101 languages. SPM is a system that learns subword units based on untokenized training data, providing a *universal* tokenizer that can operate on any language. One challenge is that the amount of monolingual data available for different languages is not the same — an effect that is extreme when considering low-resource languages. Languages with small quantities of data may not have the same level of coverage in subword units, or an insufficient quantity of sentences to represent a diverse enough set of content. To address

[1] http://data.statmt.org/cc-100/

this, we train our SPM model with temperature upsampling similar to Conneau et al. (2020), so that low-resource languages are represented. Finally, to compute BLEU, we apply SPM tokenization to the system output and the reference, and then calculate BLEU in the space of sentence-pieces. Due to the difference in tokenization, spBLEU scores are not strictly comparable across different target languages. However, to compare different models, here we use averages across the same set of target languages assuming that difference in tokenizations do not favor any specific model. In Goyal et al. (2021) this metric is described as spBLEU, but in this paper we use BLEU and spBLEU interchangeably.

3 Participants

In this section, we list each of the task participants and briefly describe each of their submissions. For reproducibility, we link to each of the model submitted, available in the Dynabench platform.

eBay (Liao et al., 2021) This submissions compares different kind of back-translation settings to improve the baseline model. They compare different generation algorithms: top-5 beam search; regular decoding without beam search; regular decoding with sampling from top-10 words. Contrary to Edunov et al. (2018), they find that top-10 decoding works best. They also consider how much English data should be used for the back translation (since it's more abundant than for the other languages). The models are trained from scratch using iterative back translation. **Models:** model 440 (SMALL-TASK1), model 441 (SMALL-TASK2), model 425 (FULL-TASK)

Huawei-TSC (Yu et al., 2021) The Huawei-TSC's team use a deep transformer encoder-decoder architecture (Sun et al., 2019), and focus their efforts on a combination of heuristics for data preprocessing, synthetic data generation, fine-tuning language-specific layers, and ensemble knowledge distillation. Compared to their baseline transformer on devtest, they get +2.8 BLEU from the synthetic data generation, +0.5 BLEU from layer fine tuning, and +0.8 BLEU from the ensemble knowledge distillation. **Models:** model 439 (SMALL-TASK2)

LMU (Lai et al., 2021) The LMU team's submission was based on a multilingual model, which were improved based on two techniques: (*i*) Tagged back-translation originating from bilingual models (+1.6 above back-translation coming from a multilingual)[2]; (*ii*) data selection w.r.t to the dev/devtest corpora following (Axelrod et al., 2011). **Models:** model 444 (SMALL-TASK1)

Maastricht University (Liu and Niehues, 2021) This submission trained a single multilingual Machine translation system by training on all 30 directions of track 2 languages. They mainly adapted the released pretrained M2M-100 model. They also did some data filtering to create a cleaner version of training corpus. Also they created synthetic pairs by taking parallel source to pivot language translation dataset and automatically translating pivot language sentences into target language, which gives 0.5 BLEU score improvement. They also tried similarity regularizer and language specific adapter weight which give 0.2 BLEU score gains overall. **Models:** model 445 (SMALL-TASK2)

Microsoft (Yang et al., 2021) The Microsoft team participated in all three tasks. The submission is based on the newly-released pretrained model DeltaLM (Ma et al., 2021a). The final submission to the shared task uses a mixture of direct and pivoted translation to improve the performance of individual directions, depending on whether the direct or pivoted models perform best. The mixture results in an improvement of +3.63 BLEU for the FULL-TASK, over their baseline architecture (24/12), but smaller improvements for the SMALL-TASK2. In addition, the models use progressive learning, which starts with a smaller architecture, noisier training data, and later changes to improve performance. The model also uses a combination of parallel, back-translated and noisy-parallel data (obtained for langs. X and Y from back-translating into X and Y) **Models:** model 483 (FULL-TASK) model 448 (SMALL-TASK1) model 457 (SMALL-TASK2)

MMTAfrica (Emezue and Dossou, 2021) This submission creates a non-English-centric multilingual translation system focusing on six African languages (Igbo, Kinyarwanda, Fon, Swahili, Xhosa, Yoruba) and English and French. The system starts from mT5 (Xue et al., 2021) and finetunes it on parallel data with additional monolingual data used

[2] Authors hypothesized that the difference in performance could be due to the implicit *self-training* coming from a multilingual model, as opposed to the diversity introduced by a bilingual model.

for online backtranslation (Sennrich et al., 2016). To cover Fon and Kinyarwanda, which are not included in FLORES-101, a small new test set was created. Compared to the small baseline models provided in Goyal et al. (2021), significant improvements were obtained.

Samsung RPH - Konvergen AI (Sutawika and Cruz, 2021) The submission of the Samsung Research Philippines/Kovergen AI's team focuses on the languages in SMALL-TASK2, in particular on data preprocessing. For large-scale multilingual models, the importance of preprocessing has risen as researchers focus on using web crawls or noisily aligned data to train translation models. In this submission, various different preprocessing techniques are applied while holding the model and architecture fixed. The authors have gains of more than 1 BLEU point from improving preprocessing. **Models:** model 443 (SMALL-TASK2)

TenTrans (Xie et al., 2021) The submission explores several techniques to improve performance. It focuses on two systems: TenTrans and FLORES101, although the second one is favored in later experimentation. The authors achieve large improvements in performance by using a the pretrained M2M124 FLORES101 model. Main benefit comes from in-domain knowledge adaptation and fine-tuning. The authors use a domain classifier based on BERT. Then they use gradual fine-tuning to gradually removing the least-likely in-domain sentence pairs at the later stages of training. They also explore other techniques, including model averaging that help to improve the performance of their system. **Models:** model 460 (SMALL-TASK2)

TelU-KU (Budiwati et al., 2021) The team from TelU-KU participated in SMALL-TASK2. Their approach explores an interesting alternative of improving NMT performance via hyper-parameter optimization (most promising for low resource languages). Although simple, this approach effectively provides improvements by +1.08 BLEU on top of the small baseline and opens up a promising direction for hyper-parameter optimization. **Models:** model 465 (SMALL-TASK2)

UMD (Bandyopadhyay et al., 2021) This system build upon the baseline M2M-124 model (Fan et al., 2020). It includes two improvements: (*i*) fine-tuning over MultiCCAligned; (*ii*) it uses ReLUs, which improve +0.8 BLEU over GELUs. In addition, the final system is the result of an extensive hyper-parameter optimization. Interestingly, the authors find that using the bible for finetuning improves performance over the baseline model despite its small size (only about 0.5 BLEU behind MultiCCAligned). **Models:** model 304 (SMALL-TASK2)

4 Evaluation Environment

All models were evaluated within the Dynaboard evaluation-as-a-service framework (Ma et al., 2021b) that is a part of the Dynabench platform (Kiela et al., 2021). This was done to ensure that the FLORES test set remains hidden while we evaluate many-to-many translation. Moreover, the testing conditions were constrained to a p2.xlarge AWS instance, which has one NVIDIA K80 GPU.

All model submissions had to be wrapped in a torchserve[3] handler and were required to follow a fixed input/output specification using Dynalab[4]. Submitting a system to the task required writing some wrapper code, and often testing different configurations (e.g. batch size), to ensure that the model was able to run under the constraints.

Given the additional work needed to run the evaluation, participants were encouraged to test the platform and to submit models early on. To avoid fine-tuning on the devtest set, we established a submission cap of one model per day.

In total, we had 81 distinct model submissions for the small task2 (South-East Asian Languages), 57 distinct submissions to the small task1 (Central / South-East European Languages), and 13 model submissions to the full task. During the evaluation period, participants were requested to mark a model as their final submission. In the end, we had 10 final submissions to the small task2, 4 to the small task 1 and 3 to the full task.

In Figure 1 we observe the total number of submissions per day. We can see that the total number of submissions per day remained low (less than 5) until August, where the number of submissions reached 16 per day.

5 Results

Present the analysis of the results for each of the tasks. Furthermore, we analyze the progress made for each task, that is, how much improvement has

[3]https://pytorch.org/serve

[4]https://github.com/facebookresearch/dynalab

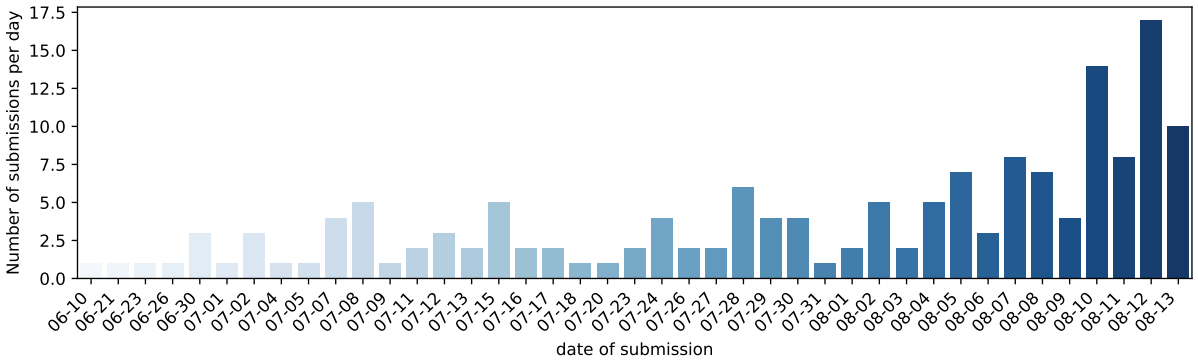


Figure 1: Submissions to the shared task through Dynabench per day. As expected, we see a rise in the number of submissions towards the end of the evaluation period.

there been between the baselines and the best models. Lastly, we analyze the difference between the models for the full task and each of the smaller tasks.

5.1 Main Results

In Table 2 we observe the final results for each of the shared tasks. From the results we observe that the DeltaLM model from the Microsoft team performs best by a large margin on the SMALL-TASK1 (+2.6 BLEU) and FULL-TASK (+9.1 BLEU), but the margin is smaller for the SMALL-TASK2 (0.6 BLEU). Below, analyze each task's results independently.

	BLEU
SMALL-TASK1 ***(CSE European langs)***	
Microsoft	37.59
eBay	34.96
LMU	31.86
baseline M2M-615	28.23
baseline M2M-175	21.33
SMALL-TASK2 ***(SE Asian langs)***	
Microsoft	33.89
eBay	33.34
TenTrans	28.89
Maastricht University	28.64
Huawei-TSC	28.40
Samsung RPH/ Konvergen AI	22.97
baseline M2M-615	16.11
UMD	15.72
TelU-KU	13.19
baseline M2M-175	12.30
FULL-TASK ***(all langs)***	
Microsoft	16.63
eBay	7.55
baseline M2M-175	6.05

Table 2: Official results for the three shared tasks in the large-scale multilingual machine translation task

SMALL-TASK1 In the Central/South-East European languages we observed that the model pre-trained with DeltaLM performed best, followed by eBay's model by a margin of 2.6 BLEU. In this task we observe that the progress between the M2M-615 baseline and the next best system of 3.6 BLEU.

SMALL-TASK2 In the South-East Asian languages task, there were many more submissions than in the other tasks. We see a smaller gap between the first and second models. These two models are very different, one using a large pre-trained language model, while the other one trains from scratch and uses iterative back translation. There is also a second cluster formed by the submission of the next three models, with a gap less than 0.5 BLEU among them. In this cluster, two models are based on the pre-trained M2M model while the third one is trained from scratch. Six out of eight participants perform better than the M2M-615 baseline, while all participants perform better than the M2M-175 baseline. The gap between the best system and the M2M-615 baseline is of 17.8 BLEU.

FULL-TASK In the full task we had fewer submissions, possibly due to the difficulty and resources to train an evaluate such models. Here the gap between the best and second-best models is significant, around 9 BLEU. However, note that the gap between the best systems and the baseline is much smaller (∼10.6 BLEU), denoting how much harder is translating more languages with similarly sized models.

5.2 Analysis of the progress on quality

One interesting aspect that we can analyze is how much progress has there been since the release of M2M-100 (Fan et al., 2020), and its subsequent adaptation for FLORES101, M2M-124. Here, we break down the improvements by language pairs to understand better the changes in performance.

Note that looking at spm-BLEU numbers across target languages can be deceiving. This is due to the different spm vocabularies that are used for each target language. However, for the sake of simplicity in the following analyses we assume that: (*i*) relative improvements (deltas) are comparable across language pairs, (*ii*) averages of relative improvements from two different source languages (say English and Hausa) into the remaining 101 languages are roughly comparable, even though the average for Hausa on the source doesn't contain on the target Hausa and contains English, and the average for English on the source doesn't contain English on the target but contains Hausa.

5.2.1 Progress on SMALL-TASK1

SMALL-TASK1 is constrained and encompasses Central and South-East European Languages. In Table 3 we see that the top performing pairs (most progress) are into and out of English, while the worst performing ones include Croatian and Macedonian. The gap between the best and the worst performing pairs is of 13 BLEU, yet on average, translation across language pairs improved 11.3 BLEU.

Source	Target	ΔBLEU
Best 5		
English	Serbian	19.08
Serbian	English	15.58
Macedonian	English	14.81
Estonian	English	14.17
Hungarian	English	13.37
Worst 5		
Hungarian	Croatian	9.05
Macedonian	Croatian	8.09
Croatian	Macedonian	6.96
Serbian	Macedonian	6.49
Serbian	Croatian	6.13
Average:		11.32

Table 3: Progress in quality for the best and worst language pairs in SMALL-TASK1

In Table 4 we present the average progress for languages in the source or target, and we observe the following: there was more progress in translating from English than any other language. However, the gap between the best and worst is less than 5 BLEU. When looking at the performance when translating into each of the task languages, we see a very similar tendency: English tops the list, Croatian is at the bottom, and the gap between best performing and worst performing languages is less than 5 BLEU.

Source	ΔBLEU	Target	ΔBLEU
English	14.20	English	13.97
Macedonian	11.65	Serbian	13.58
Estonian	11.43	Hungarian	10.96
Hungarian	11.22	Estonian	10.91
Serbian	9.84	Macedonian	9.47
Croatian	9.58	Croatian	9.02

Table 4: Average progress for each of the languages in SMALL-TASK1

5.3 Progress on SMALL-TASK2

For SMALL-TASK2, there was a significant progress on languages like Tamil (tam) and Tagalog (tgl). In Table 5 we see a progress of 30+ BLEU for translation between Tamil <> English. This is encouraging, as the baseline model had issues translating from/into Tamil. It is also encouraging to see that even for the translation between Malay <> Indonesian (which was strong to begin with), we see more than 10+ BLEU improvement. On average, we see an improvement of 21.8 across all directions. It's important to note the fact that all submissions for this task were constrained, so these improvements come from better modeling and training techniques.

Another aspect to note comes from Table 6, where we see that the language with most progress is Tamil, followed by English and Tagalog. On the other hand, in this case we see more disparity on the progress between the languages with most and least progress. For instance, it is harder to translate into Javanese, which only improves 14.7 BLEU on average.

5.3.1 Progres on FULL-TASK

In Table 7 we present the deltas between the best scores in the competition for each language pair, and the baseline. We observe that there are significant improvements for certain languages, particularly: Welsh (cym), Irish (gle), Maltese (mlt) and their pairings with English. These are languages for which the original M2M model was doing poorly,

Source	Target	ΔBLEU
Best 5		
English	Tamil	32.63
English	Tagalog	31.04
Tagalog	English	30.16
Tamil	English	30.00
Indonesian	Tamil	28.45
Worst 5		
Tagalog	Javanese	14.67
Malay	Javanese	12.40
Indonesian	Malay	11.59
Indonesian	Javanese	11.05
Malay	Indonesian	10.45
Average:		21.75

Table 5: Progress in quality for the best and worst language pairs in SMALL-TASK2

Source	ΔBLEU	Target	ΔBLEU
Tamil	24.35	Tamil	27.29
English	24.30	Tagalog	25.29
Tagalog	23.19	English	24.68
Javanese	20.68	Malay	19.72
Indonesian	19.13	Indonesian	18.81
Malay	18.88	Malay	14.74

Table 6: Average progress for each of the languages in SMALL-TASK2

yet the DeltaLM model is doing much better [5]. In fact, as seen in Fig. 2, these language pairs are an exception, and most language pairs fall around the 11 BLEU improvement range. The average improvement across language pairs is 10.6 BLEU. However, there are several language pairs for which there was no progress at all. In Fig. 2, close to 10% (∼1K pairs) have less than 5 BLEU improvement.

[5]Since this is an unconstrained submission, it is hard to know what data went into the models. However, we hypothesize that the improvement is likely due to the amount of training data available for DeltaLM. As pointed out in Yang et al. (2021) their model contains about 300K sentences for Maltese (mt), 1.5M sentences for Irish (ga), and 3M sentences for Welsh (cy)

Source	Target	ΔBLEU
Best 5		
English	Welsh	46.41
Irish	English	43.55
English	Irish	43.10
Maltese	Welsh	42.88
Irish	Maltese	41.83
Average:		10.60

Table 7: Progress in quality for the best and worst language pairs in FULL-TASK. Note that we exclude the worst performing pairs, which made no progress at all.

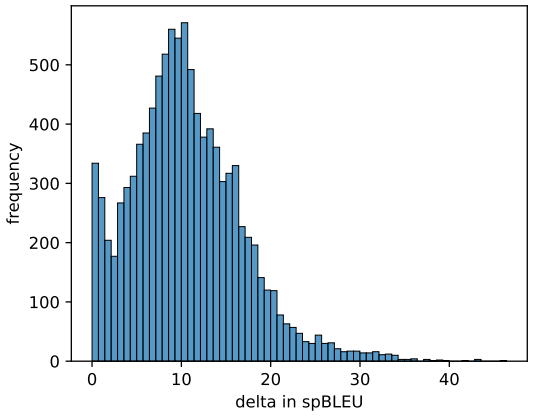


Figure 2: Distribution of improvements in BLEU for different language pairs in the full task

To facilitate the analysis of the progress across languages, in Fig. 3 we present the improvements by language groupings. We see big improvements coming from Other Indo-European (influenced by Irish, Welsh), Dravidian (influenced by Tamil, Telugu), Austronesian (influenced by Tagalog). However we note that there is very little progress for African Languages as represented by the Bantu and Nilotic subgroups. Another interesting finding is that progress trends to be lower when translating into harder languages.

In summary, there is large progress for a few languages, but sadly, there is little progress made for very low-resource languages, particularly those unrelated to other major languages.

5.4 Moderately Multilingual vs. Massively Multilingual

A natural question that arises is: what is the gap that remains between what we're calling moderately multilingual models (MoM), i.e models handle just a few languages and a couple dozen pairs; vs. the massively multilingual models (M2M) that handle hundreds of languages and tens of thousand pairs?

To analyze this aspect, we compare the best models for the full task, and each of the small tasks.

5.4.1 SMALL-TASK1 vs. FULL-TASK

In Figure 4 we present the scores of the best system for task1 (MoM) vs. the best system for the full task (M2M). Here we observe that there is a consistent gap of about 4.7 BLEU between the MoM and the M2M models when averaging across source languages. We can observe on the distribution of deltas of performance that drops in performance are similarly distributed across languages. This

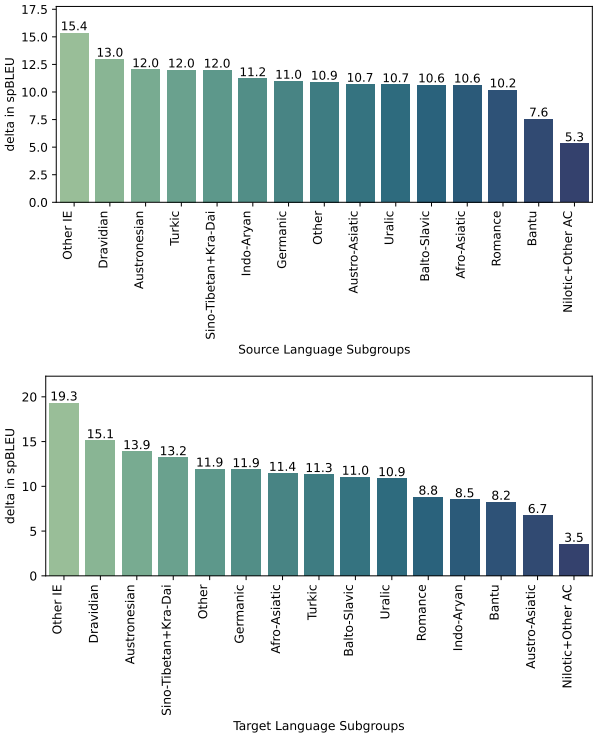


Figure 3: Average BLEU improvements per languages in the source and target language families

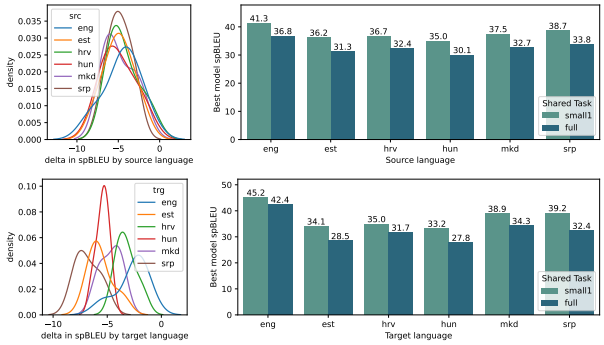


Figure 4: Comparison of average performances of the best systems in the FULL-TASK and SMALL-TASK1 by source and target languages

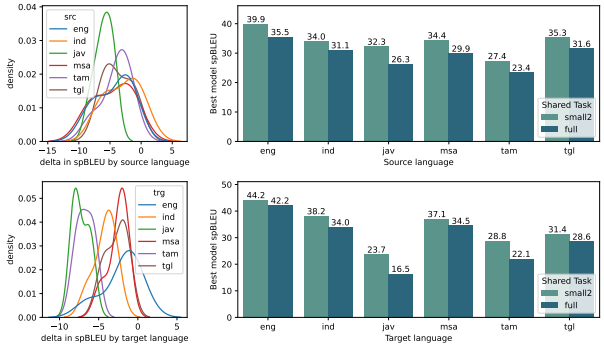


Figure 5: Comparison of average performances of the best systems in the FULL-TASK and SMALL-TASK2 by source and target languages

suggests that the *curse of multilinguality* (Conneau et al., 2020), i.e. the loss in performance by adding more languages into to a model with fixed capacity, affects equally the encoding of different languages to a rate of about 0.05 BLEU per language added to the model. This is encouraging, as it suggests that encoding is robust to the addition of new languages.

On the other hand, when we look at the target side the picture is quite different. Particularly, we observe more variation in performance, ranging from -2.7 BLEU for English to -6.8 BLEU for Serbian. We hypothesize that these differences could be due to a combination of factors: (*i*) amount of supervision (which would explain why English performance doesn't drop as much), (*ii*) additional supervision from similar languages, (*iii*) morphological richness (which would explain why Hungarian and Estonian are more affected), and (*iv*) script usage (which would explain why Serbian is more affected than Croatian). However, proving these hypotheses is beyond the scope of this paper.

5.4.2 SMALL-TASK2 vs. FULL-TASK

In Figure 5 we present the scores of the best system for task2 (MoM) vs. the best system for the full task (M2M). Here we see again that the model with more parameters per language is still ahead by about 4.26 BLEU. We also observe more variability in the distribution of drops in performance, notably, Javanese, the lowest resource language, being the most different to the others.

On the target side, we observe that English is ahead of the curve, showing the least regression. On the other hand Javanese and Tamil further reinforce our observations that additional supervision and morphology play an important role in decoding performance.

5.5 African Languages

While not officially a track on this year's competition, Emezue and Dossou (2021) focused on the task of multilingual machine translation for African languages that are in FLORES-101. They introduced MMTAfrica, the first many-to-many multilingual translation system for six African languages: Fon (fon), Igbo (ibo), Kinyarwanda (kin), Swahili/Kiswahili (swa), Xhosa (xho), and Yoruba (yor) and two non-African languages: English (eng) and French (fra). For multilingual translation concerning African languages, a novel backtranslation

and reconstruction objective, BT&REC, was introduced which is inspired by the random online back translation and T5 modelling framework respectively, to effectively leverage monolingual data. Additionally, MMTAfrica improves over the FLORES 101 benchmarks (BLEU gains ranging from +0.58 in Swahili to French to +19.46 in French to Xhosa).

6 Conclusion and Future Work

In this paper we presented the first iteration of the large-scale multilingual translation task. This task attracted several teams from across the globe and many models submissions. We kept the test set blind and used a platform to evaluate model submissions under a controlled environment. In this task, we observed significant progress in translation quality across tasks, but particularly in the *small* tasks. We observed that pre-trained language models and large amounts of back-translation (either at one go, or in iterative fashion) were important tools used by many participants.

We observed that models that have to translate fewer languages trend to do better on average, and that lower resources and morphology complicate translation, particularly for decoding. We also observed that languages in certain groups, like the African languages in the Bantu and Nilotic families, experience little to no improvement.

In the future, we want to organize shared tasks with languages for which little or no progress was achieved this time around. Additionally, we want to open up the FLORES evaluation setup to other organizers interested groups of languages within the FLORES-101 set.

Acknowledgements

We would like to thank Geeta Chouhan and Hamid Shojanazeri for their support setting up the GPU inference and batch decoding with torchserve. We would like to thank Carlos Escapa for his support in getting compute credits for this competition, and Microsoft Azure, Google Cloud and Amazon AWS for donating credits for participants.

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, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Saptarashmi Bandyopadhyay, Tasnim Kabir, Zizhen Lian, and Marine Carpuat. 2021. The University of Maryland, College Park Submission to Large-Scale Multilingual Shared Task at WMT 2021. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Sari Dewi Budiwati, Tirana Fatyanosa, Mahendra Data, Dedy Rahman Wijaya, Patrick Adolf Telnoni, Arie Ardiyanti Suryani, Agus Pratondo, and Masayoshi Aritsugi. 2021. To Optimize, or Not to Optimize, That Is the Question: TelU-KU Models for WMT21 Large-Scale Multilingual Machine Translation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir, et al. 2021. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. *arXiv preprint arXiv:2104.08726*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Conference of the Association for Computational Linguistics (ACL)*.

Chris Chinenye Emezue and Bonaventure F. P. Dossou. 2021. MMTAfrica: Multilingual Machine Translation for African Languages. In *Proceedings of the*

Sixth Conference on Machine Translation, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

∀, Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Tajudeen Kolawole, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddee Hassan Muhammad, Salomon Kabongo, Salomey Osei, et al. 2020. Participatory research for low-resourced machine translation: A case study in african languages. *arXiv preprint arXiv:2010.02353*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in nlp.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Garry Kuwanto, Afra Feyza Akyürek, Isidora Chara Tourni, Siyang Li, and Derry Wijaya. 2021. Low-resource machine translation for low-resource languages: Leveraging comparable data, code-switching and compute resources. *arXiv preprint arXiv:2103.13272*.

Wen Lai, Jindřich Libovický, and Alexander Fraser. 2021. The LMU munich system for the wmt 2021 large-scale multilingual machine translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Baohao Liao, Shahram Khadivi, and Sanjika Hewavitharana. 2021. Back-translation for Large-Scale Multilingual Machine Translation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Danni Liu and Jan Niehues. 2021. Maastricht University's Large-Scale Multilingual Machine Translation System for WMT 2021. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021a. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021b. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *Conference of the Association for Computational Linguistics (ACL)*.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Lintang Sutawika and Jan Christian Blaise Cruz. 2021. Data Processing Matters: SRPH-Konvergen AI's Machine Translation System for WMT'21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ye Kyaw Thu, Win Pa Pa, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Introducing the Asian language treebank (ALT). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1574–1578, Portorož, Slovenia. European Language Resources Association (ELRA).

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Wanying Xie, Bojie Hu, Han Yang, Dong Yu, and Qi Ju. 2021. TenTrans Large-Scale Multilingual Machine Translation System for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual Machine Translation Systems from Microsoft for WMT21 Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Zhengzhe Yu, Daimeng Wei, Zongyao Li, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. HW-TSC's Participation in the WMT 2021 Large-Scale Multilingual Translation Task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

GTCOM Neural Machine Translation Systems for WMT21

Chao Bei, Hao Zong, Qinming Liu and Conghu Yuan
Global Tone Communication Technology Co., Ltd.
{beichao, zonghao, liuqingmin and yuanconghu}@gtcom.com.cn

Abstract

This paper describes the Global Tone Communication Co., Ltd.'s submission of the WMT21 shared news translation task. We participate in six directions: English to/from Hausa, Hindi to/from Bengali and Zulu to/from Xhosa. Our submitted systems are unconstrained and focus on multilingual translation model, back-translation and forward-translation. We also apply rules and language model to filter monolingual, parallel sentences and synthetic sentences.

1 Introduction

We applied fairseq(Ott et al., 2019) as our develop tool and use transformer(Vaswani et al., 2017) as the main architecture. The primary ranking index for submitted systems is BLEU (Papineni et al., 2002), therefore we apply BLEU as the evaluation matrix for our translation system.

For data preprocessing, punctuation normalization, tokenization and BPE(byte pair encoding) (Sennrich et al., 2015) are applied for all language. Further, we apply truecase model for English, Hausa, Zulu and Xhosa according to the character of each language. Regarding to the tokenization, we use polyglot [1] as the tokenizer for Hausa, Hindi, Bengali, Zulu and Xhosa. Besides, knowledge based rules and language model are also involved to clean parallel data, monolingual data and synthetic data.

Due to the quantity limitation of parallel corpus in low-resource language pair, we use forward-translation with monolingual data to generate more synthetic data instead of knowledge distillation (Kim and Rush, 2016). Here forward-translation refers to translate the source language sentences to the target language, and then clean this synthetic data with the above described method. In order to enrich the low-resource language corpus, we add English to X corpus to construct a multilingual translation model. This multilingual model is expected to obtain the inner deep information among all languages and give us a better translation.

This paper is arranged as follows. We firstly describe the task and show the data information, then introduce our multilingual translation model. After that, we describe the techniques on low-resource condition and show the conducted experiments in detail of all directions, including data preprocessing, model architecture, back-translation, forward-translation and multilingual translation model. At last, we analyze 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. For the directions between Hindi and Bengali, the parallel data is mainly from CC-Aligned, as well as the directions between Zulu and Xhosa. For the directions between English and Hausa, the parallel data is mainly from English-Hausa Opus corpus, Khamenei corpus, ParaCrawl v8. The monolingual data we used includes: News Crawl in English, Hindi and Bengali; extended Common Crawl in Hausa, Xhosa and Zulu; Common Crawl in Hausa. All language directions we participated are new tasks in this year, therefore we only use the provided newsdev2021 as our development set for the directions of English to/from Hausa, flores-dev for the directions of Hindi to/from Bengali and Zulu to/from Xhosa.

3 Multilingual Translation Model

In low-resource condition, data augmentation and pretrained model are the most effective approaches to improve translation quality. According Google's Multilingual Neural Machine Translation System(Johnson et al., 2017), we use other language

[1] https://github.com/aboSamoor/polyglot

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 100–103

language	number of sentences
bn-hi parallel data	3.3M
en-bn parallel data	2.2M
en-hi parallel data	2.2M
en-ha parallel data	750K
xh-zu parallel data	60K
en-xh parallel data	41K
en-zu parallel data	45K
en monolingual data	93.4M
bn monolingual data	59.7M
hi monolingual data	46.1M
ha monolingual data	46.1M
xh monolingual data	1.6M
zu monolingual data	2M
en-ha development set	2000
bn-hi development set	997
xh-zu development set	997

Table 1: Task Description

pairs parallel data along with the provided bilingual data to training a multilingual translation model, the low-resource language pair is expected to get the benefits from other language pair's parallel data, especially in similar language. For the multilingual model preprocessing, we add a language tag at beginning of each source sentence, and use joint BPE for all languages in one multilingual translation model.

4 Experiment

4.1 Model architecture

- **Baseline** Table 2 shows the baseline model architecture.

- **Big transformer** We use fairseq to train our model with transformer big architecture. The model configuration and training parameters is almost same as GTCOM2020(Bei et al., 2020).

4.2 Training Step

This section introduces all the experiments we set step by step and Figure1 shows the full improvement status.

- **Date Filtering** The methods of data filtering are mainly the same as GTCOM2020, including knowledge based rules, language model and repeat cleaning.

configuration	value
architecture	transformer
word embedding	512
Encoder depth	5
Decoder depth	5
transformer heads	2
size of FFN	2048
attention dropout	0.2
dropout	0.4
relu dropout	0.2

Table 2: The FLoRes model architecture.

- **Baseline** We use FLoRes (Guzmán et al., 2019) architecture to construct our baseline in low-resource condition.

- **Multilingual translation model.** Due to the language distinction, We construct two multilingual translation models with the corpus organized as: 1. English-Bengali parallel data, English-Hindi parallel data and Bengali-Hindi parallel data; 2. English-Hausa parallel data, English-Xhosa parallel data, English-Zulu parallel data and Xhosa-Zulu parallel data. Each multilingual translation model has a shared vocabulary.

- **Back-translation** We use multilingual translation model to translate the target language sentence to source language, and clean synthetic data with language model. Here, we translate all language pairs we have added into this multilingual translation model. Then we combine the cleaned back-translation data and provided parallel sentences to train a new multilingual translation model.

- **Forward-translation** Source language sentences are translated to target language, and then cleaned by language model. Again we add this forward translation data with cleaned back-translation data and provided parallel sentences to train another multilingual translation model.

- **Joint training** Repeat generating back-translation data and forward-translation data by currently trained best multilingual model until there is no improvement.

- **Transformer big** Using bilingual parallel data and synthetic data generated by cur-

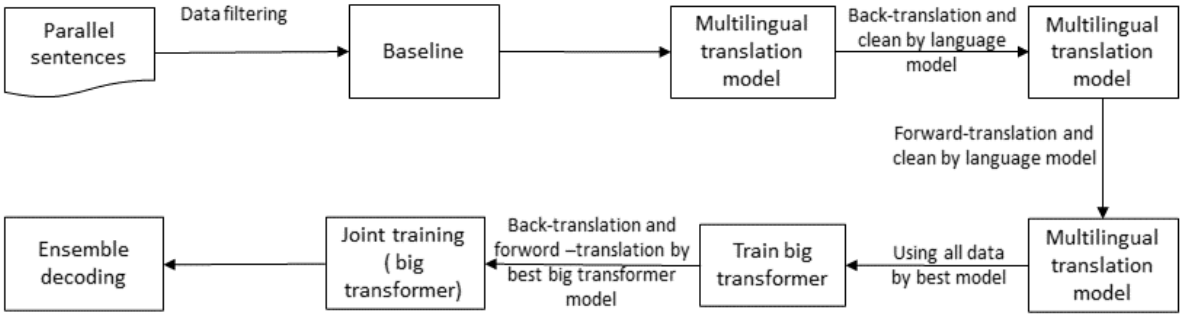


Figure 1: The whole work flow.

model	bn2hi	hi2bn
baseline	19.00	11.20
multilingual translation model	19.33	11.22
+ back-translation	23.63	14.80
+ forward-translation	23.95	14.95
+ joint training	24.05	15.02
big transformer	24.11	15.14
+ Ensemble Decoding	25.13	15.86

Table 3: The BLEU score between Hindi and Bengali.

model	en2ha	ha2en
baseline	11.04	12.02
multilingual translation model	12.20	13.09
+ back-translation	18.27	17.56
+ forward-translation	18.74	18.21
+ joint training	18.95	18.59
big transformer	19.32	18.91
+ Ensemble Decoding	21.09	21.58

Table 4: The BLEU score between English and Hausa after truecase.

rently best multilingual model to train a bilingual model with transformer big architecture and repeat back-translation step and forward-translation step, until there is no improvement.

- **Ensemble Decoding** We use GMSE Algorithm (Deng et al., 2018) to select models to obtain the best performance.

5 Result and analysis

Table 3, Table 4 and Table 5 show the BLEU score we evaluated on development set for Hind to/from Bengali, English to/from Hausa and Xhosa to Zulu

model	xh2zu	zu2xh
baseline	10.58	10.60
multilingual translation model	11.66	10.73
+ back-translation	12.48	10.76
+ forward-translation	12.70	10.86
+ joint training	12.74	10.92
big transformer	12.77	10.95
+ Ensemble Decoding	12.95	11.02

Table 5: The BLEU score between Xhosa and Zulu after truecase.

respectively. Back-translation is still the most effective method with improvement ranging from 0.03 to 6.07 BLEU score in low-resource condition. And multilingual translation model gets the improvement ranging from 0.02 to 1.16 BLEU score. Forward translation enrich the information in low-resource condition, with improvement of 0.1 to 0.65 BLEU score. Further, ensemble decoding increase the performance with 0.07 to 2.67 BLEU score.

6 Summary

This work mainly focus data augmentation and pay less attention on modeling. Because optimizing translation by data augmentation is the most elegant way for a commercial system. It can avoid many unexpected translation result generated by a newly proposed model which may give our customers worse translating experience.

This paper describes GTCOM's neural machine translation systems for the WMT21 shared news translation task. For all translation directions, we build systems mainly base on multilingual translation model and enrich information by back-

translation and forward-translation. The effect of increasing information is also dependent on data filtering.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the National Key Research and Development Program of China (2020AAA0108005). And this work is supported by Global Institute of Intelligent Language Technology[2] of Global Tone Communication Technology Co., Ltd.[3]

References

Chao Bei, Hao Zong, Qingmin Liu, and Conghu Yuan. 2020. GTCOM neural machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 100–104, Online. Association for Computational Linguistics.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, et al. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 368–376.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

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.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: 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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

[2]http://www.2020nlp.com/

[3]http://www.gtcom.com.cn/

The University of Edinburgh's English-German and English-Hausa Submissions to the WMT21 News Translation Task

Pinzhen Chen Jindřich Helcl Ulrich Germann Laurie Burchell Nikolay Bogoychev
Antonio Valerio Miceli Barone Jonas Waldendorf Alexandra Birch Kenneth Heafield
School of Informatics, University of Edinburgh
{pinzhen.chen, jhelcl, ulrich.germann, laurie.burchell, n.bogoych,
amiceli, jwaldend, a.birch, kenneth.heafield}@ed.ac.uk

Abstract

This paper presents the University of Edinburgh's constrained submissions of English-German and English-Hausa systems to the WMT 2021 shared task on news translation. We build En-De systems in three stages: corpus filtering, back-translation, and fine-tuning. For En-Ha we use an iterative back-translation approach on top of pre-trained En-De models and investigate vocabulary embedding mapping.

1 Introduction

We describe the University of Edinburgh's participation in English↔German (En↔De) and English↔Hausa (En↔Ha) at the WMT 2021 news translation task. We apply distinct sets of techniques to the two language pairs separately, as the two pairs are very different in terms of language proximity and the availability of resources. We follow the constrained condition where we only use the provided data available to all participants.

For En↔De we first employ rule-based and dual conditional cross-entropy filtering to clean the datasets. Then we add to training back-translations generated in a few ways: tagged, greedy, beam search and sampling. We fine-tune our models on past years' test sets, and finally tune a few configurations: length normalization, test sentence splitting, and German post-processing.

For En↔Ha we adopt iterative back-translation, where at each iteration we initialize the model parameters from an En-De model in the corresponding direction (En→De for En→Ha and De→En for Ha→En). These En-De models are trained in the same way as those submitted to the En-De task, except that their vocabulary includes subwords from the Hausa language. Besides, we experiment with vocabulary mapping at the embedding level.

Some configurations are kept consistent across language pairs and systems. Sentences are tokenized using SentencePiece (Kudo and Richardson, 2018) with a 32K shared vocabulary, except that we added a few extra tokens for tagged back-translation. All models are trained following Marian's Transformer-Big task preset (Vaswani et al., 2017; Junczys-Dowmunt et al., 2018) unless otherwise specified: 6 encoder and decoder layers, 16 heads, 1024 hidden embedding size, tied embeddings (Press and Wolf, 2017), etc.[1]

Section 2 and Section 3 describe the detailed model building process for En↔De and En↔Ha respectively. While awaiting human evaluation results, we summarize our automatic metric scores on the WMT 2021 test sets computed by the task organizers in Table 1.

Direction	BLEU	ChrF
En→De	29.90	0.59
De→En	51.78	0.66
En→Ha	14.81	0.45
Ha→En	14.89	0.42

Table 1: Automatic metric scores on WMT21 test computed by the task organizers.

2 English↔German

2.1 Data and cleaning

English-German is considered to be a high-resource language pair, with over 90 million parallel and hundreds of millions monolingual sentences provided in the shared task. Following our last year's submission (Germann, 2020), we divide the data into three categories, and we use all the parallel data, as well as monolingual news from 2018 to 2020:

- High-quality parallel: News Commentary, Europarl and Rapid.

[1] https://github.com/marian-nmt/marian/blob/master/src/common/aliases.cpp

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 104–109
November 10–11, 2021.

- Crawled parallel: ParaCrawl, WikiMatrix, CommonCrawl, and WikiTitles.
- Monolingual news: News Crawl

The majority of parallel data are mined and aligned sentences from the web (Bañón et al., 2020; Schwenk et al., 2021), so our first step is corpus filtering to remove noisy sentences which could harm neural machine translation (Khayrallah and Koehn, 2018). We run rule-based filtering using FastText language identification (Joulin et al., 2016), and various handcrafted features such as sentence length, character ratio and length ratio. Similar rules are applied on the monolingual data, omitting the features designed for parallel data. More details can be found in our cleaning script which is made public.[2]

We then train seed Transformer-Base models on the filtered high-quality data, as well as the crawled data separately, to (self-)score translation cross-entropy of the crawled parallel sentences. This enables us to rank and filter out sentences by their dual conditional cross-entropy (Junczys-Dowmunt, 2018). The method prefers the sentences in a pair to have low and similar translation cross-entropy given each other. After empirical trials, we find it is always better to score using models trained on the high-quality data, and we choose to keep the best 75% of the crawled data. The filtering efforts are reported in Table 2. Next, we train Transformer-Big models on the combination of filtered high-quality and crawled data. These models serve as baselines and are used for back-translation later.

Amount of crawled	Scoring model	De→En	En→De
top 25%	high-qual	41.47	-
	crawled	39.35	-
top 50%	high-qual	41.64	**43.68**
	crawled	41.51	-
top 75%	high-qual	**42.15**	**43.40**
	crawled	41.90	-
all	-	**42.02**	42.70

Table 2: BLEU of filtering experiments on WMT19 test used as dev.

2.2 Back-translation

Since its introduction, back-translation (Sennrich et al., 2016) has been widely used to boost NMT. We use ensembles of our best seed and baseline models trained on the filtered data, to generate back-translations from the monolingual news data from 2018 to 2020, hoping that the domains are similar to that of the test. For En→De we mix back-translations generated using greedy search, beam search, and sampling; for De→En, we adopt tagged back-translation (Caswell et al., 2019).

After merging the original and back-translated data, for each direction we train 4 standard Transformer-Big models, as well as a model with 8 encoder layers and 4 decoder layers. Specifically for De→En, we have an extra pre-layer normalized variant.

As we observed last year, validation BLEU does not improve after we add back-translated data to training. As a result, after the models converge, we continue training them on filtered parallel data only. The models' validation BLEU scores[3] on WMT19 test are displayed in Table 3.

Configuration	De→En	En→De
Baseline	42.2	43.4
+ BT	41.8	43.0
+ cont. training	**42.5**	**43.6**

Table 3: Average BLEU scores of BT experiments on WMT19 test used as dev.

2.3 Fine-tuning and submission

We grid search on length normalization during decoding, and find 1.2 to be ideal for En→De and 0.8 for De→En. Particularly for En→De, we have two more steps to make German text read more natural: 1) continued training on 25% title-cased parallel data to improve headline translation and 2) post-processing on German quotes to make them consistent.

Previous submissions show that fine-tuning on past years' test data helps model performance (Schamper et al., 2018; Koehn et al., 2018). In the early years of WMT news translation tasks, the test sentence pairs can originate in either source or target language, and are translated and merged into one set. However, the current evaluation is on translating sentences originally in the source language only. Therefore, we experiment with fine-tuning on the combined sets, as well as on sentence pairs originated from the source language. We fine-tune

[2] https://github.com/browsermt/students/tree/master/train-student/clean

[3] sacreBLEU (Post, 2018) with signature BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1

all our models on WMT 2008-2019 test sets and validate on WMT 2020 test set.

While the training data contain mainly one sentence per line, the test set can have multiple sentences in the same segment. As a result, we split each test instance into single sentences, translate, and rejoin them. We experiment with fine-tuning and sentence splitting on the 8-encoder-4-decoder variant for both languages. Table 4 indicates that the model achieves the best BLEU (and a significant improvement over BT baseline) if we fine-tune it on previous test sentences originating in the source language only, and split long sentences in both validation and test sets.

FT on	Dev split	Test split	De→En	En→De
BT baseline			30.8	31.9
none		✓	41.7	35.2
all			34.7	-
all		✓	41.1	-
all	✓		31.2	-
all	✓	✓	41.9	36.7
orig.	✓	✓	**42.5**	**36.9**

Table 4: BLEU of fine-tuning and sentence-splitting experiments on WMT20 test

For each translation direction, we apply the best configuration to each model and ensemble them by averaging their predictions post-softmax. Overall, we have a 5-model ensemble for En→De, and a 6-model ensemble De→En.

3 English↔Hausa

3.1 Data

The main sources of English-Hausa parallel data are OPUS (Tiedemann, 2012) and ParaCrawl. We also include data from WikiTitles[4] and the Khamenei[5] corpora, which are however much smaller. In total, we gather 759,061 parallel sentences. For back-translation, we use 9.5 million monolingual Hausa sentences from Common Crawl, Extended Common Crawl, and News Crawl provided by the task organizers. We randomly select 50 million English monolingual sentences from the News Crawl collections from 2018, 2019, and 2020.

[4] http://data.statmt.org/wikititles/v3/

[5] http://data.statmt.org/wmt21/translation-task/ha-en/khamenei.v1.ha-en.tsv

For training, we use a mix of back-translated monolingual data and parallel data. Since the dataset sizes differ substantially, we over-sample the parallel data to achieve a balanced mix: 10× for English→Hausa, and 50× for Hausa→English. Similar to our En-De models, we used tagged back-translation to distinguish synthetic and authentic sentences in the data.

3.2 Iterative back-translation and fine-tuning

In our experiments, we combine a transfer learning approach (Zoph et al., 2016; Kocmi and Bojar, 2018) with 3 iterations of back-translation (Hoang et al., 2018; Edunov et al., 2018). In each iteration, we initialize the En→Ha model with a pre-trained En→De Transformer-Big model (and vice versa for the other direction). Then, we fine-tune the model on the English-Hausa data created by the model from the previous back-translation iteration (the initial model for the first iteration is fine-tuned on parallel data only).

We notice that the model generates a large number of empty translations. We suppress this issue by taking the second-best candidate translation from the n-best list if the first one is empty. Another problem is heavy overfitting in the models. In many translations, the sentences begin with the prefix "Never miss an important update!", followed by the actual translation. Unfortunately, we only noticed this issue after the submission.

3.3 Vocabulary embedding mapping

An additional approach we investigate is mapping the Hausa vocabulary to the German embeddings of the En→De model, when initializing the En→Ha model. We train the models with a 32K SentencePiece vocabulary obtained from datasets in all three languages. Using the frequency-based metric introduced by (Wang et al., 2020) we assign each SentencePiece token to an English, German, Hausa or joint vocabulary. This results in 9192 German tokens, 6485 Hausa tokens and a joint vocabulary of approximately 11k. Having established a separate Hausa and German vocabulary it is then possible to map between the embeddings of the two.

In order to map the vocabularies, we independently train BWEs (bilingual word embeddings) using an implementation of Bivec (Luong et al., 2015) combined with FastText (Bojanowski et al., 2017). This implementation uses a joint learning objective as described by Liu et al. (2020) utilising alignments combined with sub-word information.

In lieu of a parallel De-Ha dataset an En→De NMT model is used to translate the English side of the En-Ha dataset. We constrain SentencePiece encoding using the previously extracted vocabularies for example the Huasa data is encoded using only the Hausa tokens and the joint tokens. Once both sides are encoded FastAlign is used to extract automatic alignments and the BWEs are trained.

We first map the Hausa tokens to their nearest neighbour using the Cross-Domain Similarity Local Scaling (Lample et al., 2018) distance metric in the order of Hausa tokens' frequency, and only permit a German token to be mapped to exactly one Hausa token. For tokens that do not have a one-to-one mapping, we adapt Gu et al. (2018)'s approach, whereby the embedding of a Hausa token is initialized to the weighted sum of all German embeddings. The weights are given by a probability distribution derived from the distance of the Hausa token to each German token in the bilingual embedding space. It is worth noting that we only map between the tokens in the Hausa and German vocabularies not any of the joint tokens. Finally, we initialize the embedding table using the new embeddings and remove all tokens identified as German. After initialization, we fine-tune the model using the parallel and back-translated data as described previously.

Our experiments show that although initializing the embedding table using a mapping-based approach results in faster model convergence, it does not improve the final BLEU score compared to just fine-tuning from the En-De models. This was observed for both the parallel data and the combined parallel and back-translated data. The outputs of the mapping approach to the baseline for the Ha-En system are qualitatively very similar and indicates that while the embedding mapping increases convergence there is no knowledge transfer from the German embeddings.

4 Conclusion

We describe our English-German and English-Hausa submissions to the news translation task at WMT 2021. For the En↔De task, fine-tuning and splitting test instances significantly boosts BLEU while back-translation alone does not help. In the En↔Ha task, we experiment with interesting low resource NMT techniques, but unfortunately, our submission contains translations from overfitted models.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825303 (Bergamot), 825627 (European Language Grid) and 825299 (GoURMET).

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

This work used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk) funded by the University of Edinburgh and EPSRC (EP/P020267/1).

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), 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).

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

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.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of*

the Fourth Conference on Machine Translation (Volume 1: Research Papers), pages 53–63, Florence, Italy. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Ulrich Germann. 2020. The University of Edinburgh's submission to the German-to-English and English-to-German tracks in the WMT 2020 news translation and zero-shot translation robustness tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 197–201, Online. Association for Computational Linguistics.

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, New Orleans, Louisiana. 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, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. 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.

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.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn, Kevin Duh, and Brian Thompson. 2018. The JHU machine translation systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 438–444, Belgium, Brussels. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Lu Liu, Yi Zhou, Jianhan Xu, Xiaoqing Zheng, Kai-Wei Chang, and Xuanjing Huang. 2020. Cross-lingual dependency parsing by POS-guided word reordering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2938–2948, Online. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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, Valencia, Spain. 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: Shared Task Papers*, pages 496–503, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. 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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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*, volume 30. Curran Associates, Inc.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime G. Carbonell. 2020. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. In *International Conference on Learning Representations*.

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.

Tune In: The AFRL WMT21 News-Translation Systems

Grant Erdmann, Jeremy Gwinnup, Timothy Anderson
Air Force Research Laboratory
{grant.erdmann, jeremy.gwinnup.1, timothy.anderson.20}@us.af.mil

Abstract

This paper describes the Air Force Research Laboratory (AFRL) machine translation systems and the improvements that were developed during the WMT21 evaluation campaign. This year, we explore various methods of adapting our baseline models from WMT20 and again measure improvements in performance on the Russian–English language pair.

1 Introduction

As part of the 2021 Conference on Machine Translation (wmt, 2021) news-translation shared task, the AFRL human language technology team participated in the Russian–English portion of the competition. We experiment with OpenNMT-tf [1] (Klein et al., 2018) and Marian [2] (Junczys-Dowmunt et al., 2018) transformer (Vaswani et al., 2017) models trained as part of our WMT20 (Gwinnup and Anderson, 2020) efforts and apply varying continued-training and fine-tuning approaches (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), including a new method to select a fine-tuning set from a separate, larger corpus not used in training.

We submit an OpenNMT-based transformer system fine-tuned on newstest test sets from 2014-2017 as our primary entry, and a Marian-based transformer system fine-tuned on newstest test sets from 2014-2018 as a contrast.

2 Data and Preprocessing

Since most of our efforts focus on fine-tuning existing models this year, we reuse the training corpus from our WMT20 systems which includes the following parallel corpora: Commoncrawl (Smith et al., 2013), Yandex[3], UN v1.0 (Ziemski et al., 2016), Paracrawl[4](Esplà et al., 2019), Wikimatrix (Schwenk et al., 2019), and backtranslated data from our WMT17 system (Gwinnup et al., 2017) as well as Edinburgh's WMT17 system (Sennrich et al., 2017) yielding a raw corpus of over 76.3 million lines.

The new Russian–English version 8 Paracrawl corpus is reserved for tuning set selection as described in Section 2.3.

2.1 Data Preparation

We re-use the fastText (Joulin et al., 2016b,a) based language ID filtered corpus with an ID threshold of 0.8 as described in Gwinnup and Anderson (2020), shown in Table 1, allowing us to make concrete progress comparisons to last year's systems.

2.2 Data Augmentation with Speech Recognition-like output

In order to build a larger pool of training data, we have created Automatic Speech Recognition (ASR) - like training data for the Russian–English translation task. Whereas written text can include upper and lowercase characters, punctuation, special symbols, and numbers written using digits, transcripts produced by ASR systems are typically uncased with no punctuation, no special symbols, and numbers written as spoken (e.g., 4.1% rendered as "four point one percent"). In previous experiments on an English-German spoken language translation task (Ore et al., 2020), we found that we could get an improvement in BLEU score by formatting the MT training data such that the source language text matched the output format of our ASR system, while leaving the target language text unmodified. We applied a similar procedure to the Russian side of the Russian-English training corpus using the text2norm.pl script from ru2sphinx.[5] This copy of the ASR-like training text was then appended to

[1] Available at: https://github.com/OpenNMT/OpenNMT-tf/

[2] Available at: https://github.com/marian-nmt/marian

[3] https://translate.yandex.ru/corpus?lang=en

[4] Version 1 Russian–English parallel data

[5] Available at: https://github.com/zamiron/ru4sphinx

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 110–116
November 10–11, 2021.

corpus	unfiltered lines	filtered lines	percent remain
commoncrawl	723,256	655,069	90.57%
news-commentary-v15	319,242	286,947	89.88%
yandex	1,000,000	901,318	90.13%
un-2016	11,365,709	9,871,406	86.85%
paracrawl-v1	12,061,155	5,173,675	42.90%
wikimatrix	5,203,872	4,287,881	82.40%
wmt17-afrl-bt	8,921,942	8,317,107	93.22%
wmt17-uedin-bt	36,772,770	29,074,022	79.06%
Total	76,367,946	58,567,425	76.69%

Table 1: Results of language-id based Russian–English corpus filtering with threshold of 0.8 as reported in (Gwinnup and Anderson, 2020)

the original training data, effectively doubling the size of the corpus.

2.3 Selecting Tuning Sets from Representative Data

We performed experiments involving automatic selection of fine-tuning corpora. Given a monolingual application corpus, we wish to test the possibility of selecting an appropriate fine-tuning set to improve a general-purpose neural MT system's performance on that application corpus. We anticipate such techniques to be of increasing importance, especially for high-value application corpora, as computational costs of subcorpus selection and fine-tuning continue to decrease.

2.3.1 Method

We performed subselection as in Erdmann and Gwinnup (2019), which can flexibly incorporate a text quality metric and multiple parallel text corpora. In short, this algorithm tries to simultaneously optimize the quality of the subset's text and its coverage of the vocabulary present in given application corpora.

Of special note is our use of clustering to select data. We hierarchically applied the MAPPER algorithm (Singh et al., 2007) to cluster sentence vectors of a monolingual corpus. The clusters deemed useful were then used to assign fuzzy clustering to the application corpus and the corpus from which we subselect. This clustering information was included as one of the text corpora.

2.3.2 Application

The application corpus we used was the Russian side of newstest2019 and newstest2020, totalling 6777 lines. The pool of possible parallel text for subselection we took to be the given 12.6M-line subset of Russian–English version 8 ParaCrawl corpus with LASER score at least 1.1. For subselection algorithms, we first preprocessed the Russian text, applying a 90k-element joint BPE. We used the algorithm in Erdmann and Gwinnup (2019) to subselect a corpus, using 3-grams in the vocabulary coverage. As a text quality metric in this algorithm we used either the provided Bicleaner scores (Sánchez-Cartagena et al., 2018; Ramírez-Sánchez et al., 2020) or the word-averaged scores provided by OpenNMT's scoring functionality, using the untuned OpenNMT model we developed for this year's task. In order to provide meaningful comparisons with our baseline fine-tuning set of newstest2014-2018, we matched its size by always subselecting a fine-tuning set with fifteen thousand lines. Fine-tuning was performed using a single-model Marian-based untuned MT system as a baseline.

Sentence vector clustering was learned using a 570M-line monolingual Russian corpus built from the concatenation of monolingual CommonCrawl (Smith et al., 2013) data provided by WMT organizers as part of our WMT18 efforts towards pretraining word embeddings. The word vectors were trained using word2vec (Mikolov et al., 2013) on this corpus, after applying a 90k-element joint BPE. These embeddings have a dimensionality of 512 to match our Marian transformer-base system configuration as described in Gwinnup et al. (2018). A randomly-chosen 100k-line subset of the corpus was used to find the clustering.

Several methods of converting word vectors to sentence vectors were considered, and we empirically chose a "softened sum" of the word vectors

w_i as the sentence vector s:

$$s = \frac{\sum w_i}{\log(1 + \text{number of words in sentence})}.$$

Clusters were considered to be useful if they covered between 1% and 5% of this corpus. In this case there were 19 such clusters, having between 1000 and 5000 representatives each. These clusters were found to have qualitative meaning to a Russian linguist: clusters with relatively high representation in our application corpus tended to be news-like, and clusters with relatively high representation in ParaCrawl tended to be noisier.

We computed membership of a given sentence vector in a fuzzy clustering sense, with weight of cluster i defined as

$$z_i = (\min \text{distance}/\text{distance}_i)^4$$

where we use Euclidean distance, and the minimum is taken over all 19 clusters. Although the exact form is empirical, note that the weight has a maximum of unity at the closest cluster and that a cluster will get lower weight if it is farther from the sentence vector. This fuzzy clustering was computed once using k-means (distance is to cluster mean) and once using single-linkage (distance is to nearest member) clustering. These two membership clusters were then averaged. Coverage of the clusters was encouraged by including the clustering as another text corpus in our standard algorithm (Erdmann and Gwinnup, 2019) — each sentence vector was converted into a 100-word "sentence," where each cluster's "word" appeared a number of times relative to the magnitude of its weight in the line's clustering[6]. Naturally, coverage of these clustering words was computed using only unigrams.

2.3.3 Results

Table 2 shows the results of our fine-tuning experiments. The "clustering" and "metric" columns designate whether clustering was incorporated and whether Bicleaner ("Bic") or NMT scoring ("NMT") was used as the text quality metric. We see consistent gains over the untuned set, even on newstest2021, which was not used in the selection. The three subselection methods produced similar results on the three test sets. Fine-tuning with our selected sets did not produce consistent improvement over our baseline fine-tuning using newstest2014-2018. Compared to this baseline fine-tuning, the new sets improved performance on newstest2019 (roughly $+0.7$ BLEU), but they lowered performance on newstest2020 (roughly -0.7 BLEU) and the unseen newstest2021 (roughly -1.1 BLEU). Our generated fine-tuning sets did not show a consistent benefit for this task, so they were not used in our submission systems. Without further information, we cannot attribute the quality of the results to the method, the quality of data in ParaCrawl, or other causes.

Our method generates a pseudo in-domain set for an unknown application domain, using only source-side data of the application corpus. This generated set can be used for fine-tuning, training, or other purposes in natural language processing. We believe that such techniques warrant further investigation, especially for an application corpus where the domain is unknown or human-curated parallel data are unavailable.

3 Machine Translation Systems

3.1 OpenNMT-tf

The OpenNMT-tf system trained for this task used the configuration for a big deep transformer network.

We used the following network hyperparameters:

- 1024 embedding size
- 4096 hidden units
- 12 layer encoder
- 12 layer decoder
- 16 transformer heads
- dropout 0.3
- attention dropout 0.1
- feed forward network dropout 0.1
- embeddings for source, target and output layers were not tied
- Layer normalization
- Label smoothing 0.1
- Learning rate warm-up 8000 steps

The corpus used for the initial model consisted of commoncrawl, paracrawl v1, and newscommentary-v13 from wmt19 and was processed

[6]For example, using a 10-word sentence for brevity, this process would convert the fuzzy cluster membership vector $[0.2, 0.0, 0.8, 1.0]$ into the sentence "0 2 2 2 2 3 3 3 3 3".

tuning set	clustering	metric	newstest2019	newstest2020	newstest2021
untuned			35.9	34.5	46.5
newstest2014-2018			37.5	35.7	49.3
selected	no	NMT	38.0	35.0	48.4
selected	no	Bic	38.3	35.0	48.2
selected	yes	Bic	38.2	34.9	47.9

Table 2: Tuning sets and resultant BLEU scores.

with SentencePiece(Kudo and Richardson, 2018) using a model with a vocabulary size of 40K trained on this ru-en corpus of 16,805,109 bi-text. This was one of our WMT20 submitted systems (Systems 3 and 4 in Table 3). Additionally the corpus was processed as described in Section 2.2 to resemble ASR output and the resulting data was combined with the above for a final count of 33,610,218 bi-text. The network was trained for 10 epochs of this training data using a batch size of 3124 and an effective batch size of 49984 using the lazy Adam (Kingma and Ba, 2015) optimizer with beta1=0.9, beta2=0.998 and learning rate 2.0. This was a system that had been originally trained for speech translation application but showed improvements in text translation as well. The final submitted system continued training an additional 2 epochs using the unfiltered data described in Table 1. This was done to try to take advantage of the larger data set and not having the computational resources or time to train a new system with with the larger data set in time for submission deadline. The output was an average of the last 8 checkpoints of training. Checkpoints were saved every 5000 steps. The system was then tuned with three epochs of newstest data from years 2014-2017 (Systems 5 and 6 in Table 3).

3.2 Marian

Our Marian systems utilize the transformer architecture in the transformer-base configuration. We use the WMT14 newstest2014 test set for validation during training and the following network hyperparameters:

- 512 embedding size
- 2048 hidden units
- 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

We experimented with tuning these systems with the concatenation of WMT newstest sets from 2014-2018 yielding a tuning corpus of 14,820 parallel sentences. For each of the five separate transformer models trained for the Marian transformer-base ensemble systems in Gwinnup and Anderson (2020), continued training was performed for 10 epochs on the concatenated tests sets. An ensemble of the five resulting tuned models is then used to decode newstest sets from 2019-2021. Resulting scores reported by SacreBLEU are shown as Row 2 in Table 3, while the baseline, untuned ensemble is shown as Row 1. We note gains between +2.0 and +3.5 BLEU as measured by SacreBLEU over the baseline ensemble system depending on test set.

4 Experimental Results

Results reported here and in Table 3 for Marian systems were scored with SacreBLEU (Post, 2018) while results for OpenNMT systems were score with mult-bleu-detok.perl from the Moses toolkit (Koehn et al., 2007). Internal comparisons between the two scoring methods have been in agreement. All scores are on detokenized cased output.

The primary submission system was the OpenNMT-tf configuration described in section 3.1 and shown in Table 3 as onmt+asr-tune. It resulted in official scores of 53.31 BLEU-all, 38.83 BLEU-A, 39.56 BLEU-B, 0.64 chrf-all, 0.63 chrf-A, and 0.64 for chrf-B on the 2021 test-set.

Post evaluation a model with the OpenNMT-tf configuration described in section 3.1 was trained on all the unfiltered data (approx. 76M million bi-text). The results are shown in Table 3 as onmt-large. The baseline onmt-large system was approx-

imately +1 BLEU better that the baseline onmt-asr system while the onmt-asr system which continued training with two epochs of the large data set and tuned with newstest2014-2017 (onmt-+asr-tune) was +2.5 BLEU better than the baseline onmt-large system which was trained with 10 epochs and comparable to the onmt-large system tuned with newstest2014-2017. Experiments were conducted on both onmt+asr and onmt-large with tuning sets comprised of different combinations of the supplied news test sets from 2014 to 2019. Tune7 is news test sets from 2014-2017 (11,820 bi-text), tune8 is news test sets from 2014-2018 (14,820 bi-text), and tune9 is news test sets from 2014-2019 (16,820 bi-text). Systems were tuned for three epochs using these tuning sets. Generally performance dropped off or decreased slightly with more than 3 epochs of tuning. To be consistent across systems and tuning sets we are only reporting results for 3 epochs. As can be seen in Table 3 all three tuning sets provided significant improvements over the baseline systems, generally in the range of +3.5 BLEU on test 2021. For onmt+asr there was little difference in tuning with tune7 or tune8 whereas tune9 was approximately +0.4 BLEU better than those two. For onmt-large tune7 did not provide as much benefit as tune8 and tune9 which were basically the same, less than 0.1 BLEU difference between the two.

5 Conclusion

While our two submission systems employ a standard method of fine-tuning to adapt models towards a test set, we find that our methods to sample a similarly-sized tuning corpus from a larger body of text while only using information about the source side of that data yields a reasonable improvement in translation quality. Such a technique could be useful in adapting translation models to specific domains where only the source language of a text source is available.

Using actual in-domain data, such as the provided news development sets, for fine-tuning provide a substantial gain in translation quality. Such data is not always available and thus other selection techniques as described in Section 2.3 come into play. Future work will investigate combining the two approaches to see if additional gains can be obtained.

The authors would like to thank Emily Conway and Braeden Bowen for their assistance in human evaluation of MT output.

References

2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Grant Erdmann and Jeremy Gwinnup. 2019. Quality and coverage: The AFRL submission to the WMT19 parallel corpus filtering for low-resource conditions task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 267–270, Florence, Italy. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation. *CoRR*, abs/1612.06897.

Jeremy Gwinnup and Tim Anderson. 2020. The AFRL WMT20 news translation systems. In *Proceedings of the Fifth Conference on Machine Translation*, pages 207–212, Online. 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: Shared Task Papers*, pages 394–398. 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*, pages 303–309, Copenhagen, Denmark. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks

#	system name	WMT newstest 2014	2015	2016	2017	2018	2019	2020	2021
1	marian-ens5-base	40.2	34.4	34.8	38.0	33.01	35.8	35.0	47.1
2	marian-ens5-tune	–	–	–	–	–	38.4	37.0	50.6
3	WMT20 onmt-base	36.87	32.58	32.48	35.50	30.76	38.26	–	–
4	WMT20 onmt-tune7	–	–	–	–	32.31	39.27	–	–
5	onmt+asr	–	–	–	–	33.17	38.08	35.86	51.05
6	onmt+asr-tune	–	–	–	–	35.71	40.39	37.61	54.49 (+3.44)
7	onmt+asr-tune7	–	–	–	–	36.15	40.91	37.54	54.58 (+3.54)
8	onmt+asr-tune8	–	–	–	–	–	40.72	37.67	54.72 (+3.67)
9	onmt+asr-tune9	–	–	–	–	–	–	38.04	55.08 (+4.03)
10	onmt-large	–	–	–	–	33.81	38.87	36.49	51.92
11	onmt-large-tune7	–	–	–	–	36.08	41.15	38.15	54.61 (+2.69)
12	onmt-large-tune8	–	–	–	–	–	40.90	38.40	55.48 (+3.56)
13	onmt-large-tune9	–	–	–	–	–	–	38.01	55.43 (+3.51)

Table 3: Experimental results for baseline and tuned systems. Marian systems are scored with SacreBLEU, OpenNMT-tf systems are scored with multi-bleu-detok.perl. Newstest2021 scored with the two supplied references. Systems 3 and 4 are WMT20 systems for progress comparison.

for efficient text classification. *arXiv preprint arXiv:1607.01759*.

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. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 177–184, New Orleans. 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 *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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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.

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*.

Brian Ore, Eric Hansen, Tim Anderson, and Jeremy Gwinnup. 2020. The AFRL IWSLT 2020 systems: Work-from-home edition. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 103–108, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bi-

fixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. 2018. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

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.

Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. 2007. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association.

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.

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.

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 2016)*.

The TALP-UPC Participation in WMT21 News Translation Task: an mBART-based NMT Approach

Carlos Escolano[1], **Ioannis Tsiamas**[1], **Christine Basta**[1 2], **Javier Ferrando**[1], **Marta R. Costa-jussà**[1], **José A. R. Fonollosa**[1]
[1] TALP Research Center, Universitat Politècnica de Catalunya, Barcelona
[2] Institute of Graduate Studies and Research, Alexandria University, Egypt
{carlos.escolano,ioannis.tsiamas,christine.basta
javier.ferrando.monsonis,marta.ruiz,jose.fonollosa}@upc.edu

Abstract

This paper describes the submission to the WMT 2021 news translation shared task by the UPC Machine Translation group. The goal of the task is to translate German to French (De-Fr) and French to German (Fr-De). Our submission focuses on fine-tuning a pre-trained model to take advantage of monolingual data. We fine-tune mBART50 using the filtered data, and additionally, we train a Transformer model on the same data from scratch. In the experiments, we show that fine-tuning mBART50 results in 31.69 BLEU for De-Fr and 23.63 BLEU for Fr-De, which increases 2.71 and 1.90 BLEU accordingly, as compared to the model we train from scratch. Our final submission is an ensemble of these two models, further increasing 0.3 BLEU for Fr-De.

1 Introduction

Monolingual data is usually more abundant than parallel data as it does not need any human processing. Neural Machine Translation (NMT) has focused traditionally on parallel data between languages and monolingual data as back-translation (Sennrich et al., 2016). This method consists of translating a monolingual corpus with an NMT system and training the model using the synthetic data. Alternatively, in recent years, pre-trained models have been proposed using monolingual data as a pre-training step with self-supervised learning, before performing task-specific fine-tuning. An example of this approach is BERT (Devlin et al., 2019), which is a Transformer model (Vaswani et al., 2017), pre-trained on masked language modeling and next sentences prediction on a large unlabeled corpus. While BERT is used primarily for classification tasks, BART (Lewis et al., 2020) has been proposed for sequence-to-sequence tasks. BART is a Transformer encoder-decoder, pre-trained as a Denoising Autoencoder (DAE) on monolingual unlabeled text. Since BART is pre-trained on monolingual data, an additional encoder should be introduced during fine-tuning to obtain a bilingual NMT system. mBART overcomes this restriction by being pre-trained on multilingual denoising. mBART (Liu et al., 2020; Tang et al., 2020) is liable to fine-tuning on several translation directions in order to obtain a multilingual NMT system.

Our participation to the news translation task at WMT focuses on translating between German (De) and French (Fr) in both directions, De-Fr and Fr-De. To accomplish this, we employ a pre-trained mBART model, and more specifically mBART50 (Tang et al., 2020), which is pre-trained with 50 languages. We fine-tune the mBART50 on both translation directions to obtain a single multilingual model for the task. To measure the importance of the pre-training step, we additionally train a Transformer with the same architecture and hyperparameters but randomly initialized. Our experiments show that the fine-tuned mBART50 can achieve better translation quality in both directions, with improvements of 2.71 for De-Fr and 1.9 for Fr-De. Apart from fine-tuning a pre-trained model, our approach also includes extensive filtering of a large bilingual corpus to ensure high-quality training data. Finally, we have considered ensembling the fine-tuned mBART50 and the trained-from-scratch Transformer for our submission. This ensembling has resulted in BLEU scores of 31.69 and 23.93 for De-Fr and Fr-De accordingly.

The rest of this paper is organized as follows: In Section 2 we describe the background techniques this work builds upon, multilingual NMT and mBART. In Section 3 we present the datasets we used for training and the techniques applied for filtering them. In Section 4 we provide the system description along with the implementation details and Section 5 involves the results of our experiments. Finally, in Section 6 we discuss the conclusions of this work and present possible directions for further research.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 117–122
November 10–11, 2021.

2 Background

2.1 Neural Machine Translation

Neural Machine Translation (NMT) uses sequence-to-sequence models, with an encoder-decoder architecture, built upon deep neural networks (Sutskever et al., 2014; Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017). In a sequence-to-sequence model, the source sentence is mapped to its contextualized representation and fed to the decoder to generate the translation output in an auto-regressive way. Traditionally, recurrent neural networks (Hochreiter and Schmidhuber, 1997) have been used for the encoder and decoder, with an attention mechanism (Bahdanau et al., 2014) that enables each target token to concentrate on certain tokens in the source sentence. Recently, the Transformer (Vaswani et al., 2017) led to large improvements on sequence-to-sequence tasks and NMT, relying exclusively on attention mechanisms. The systems trained in this work, are also based on Transformer models.

2.2 Multilingual NMT

Multilingual NMT aims to provide a single model that can translate several language directions (Firat et al., 2016; Johnson et al., 2017). These can be one-to-many, many-to-one, or many-to-many, with the "one" being usually English due to the broadly available corpora. Previous studies have explored different design approaches, focusing on sharing parts of the model between the different languages, with shared encoder-decoder attention between languages (Firat et al., 2016), a shared encoder (Sen et al., 2019), a task-specific attention (Blackwood et al., 2018), shared parameters (Zhu et al., 2020) and full model sharing (Johnson et al., 2017). Recently, the paradigm of full model sharing has been extended to accommodate for many more languages and directions by training huge models for massively multilingual NMT (Arivazhagan et al., 2019; Fan et al., 2020). Our submission is also based on multilingual models that are fully shared between the two language directions German-French and French-German.

2.3 mBART

BART (Lewis et al., 2020) is a Transformer encoder-decoder, which is pre-trained with self-supervised learning on reconstructing the text corrupted by a noise function. Its multilingual version, mBART (Liu et al., 2020), uses the same self-supervised approach, but reconstructs corrupted text from multiple monolingual corpora. The nature of this pre-training makes mBART a good initialization for a multilingual NMT system. mBART can be fine-tuned on multiple bitext corpora providing gains in all directions of 25 languages, except for the very highest resource ones (Liu et al., 2020). Our system is initialized with mBART50 (Tang et al., 2020) (an extension of mBART from 25 to 50 languages). This initialization is followed by multilingual fine-tuning on both directions of a large German-French bitext.

3 Data

In this section we introduce the datasets used for training our systems and we go through the data filtering process that was applied in each one of them.

3.1 Datasets

In order to train Transformer models for Machine Translation, commonly, a large parallel corpus is needed. For the purpose of this research, we focus on creating a French-German parallel corpus from several publicly available datasets. More specifically, these are the Europarl (Koehn, 2005), Paracrawl (Bañón et al., 2020), Common Crawl[1], News Commentary (Tiedemann, 2012), Wiki Titles[2], Tilde Rapid and EESC (Rozis and Skadiņš, 2017), WikiMatrix[3] and TED Talks (Cettolo et al., 2012). If the dataset contains more languages, we only keep examples that have non-empty sentences for French-German. We use the development and test data of the news test datasets of 2019 and 2020, as provided by WMT. The size of each dataset can be found in the first column of Table 1.

3.2 Data Filtering

We employ two stages of data filtering to ensure that our system is trained on high-quality data. In the first stage, we process each example, either from the French or German side, separately by altering its content. This process includes the following steps in the listed order:

1. Removal of non-utf8 characters
2. De-escaping html characters

[1] http://data.statmt.org/wmt19/translation-task/fr-de/bitexts

[2] http://data.statmt.org/wikititles/v3/

[3] https://github.com/facebookresearch/LASER/tree/master/tasks/WikiMatrix

3. Normalization of different types of punctuation

4. Normalization of spacing

5. Removal of redundant apostrophes

The normalization of spacing and punctuation is applied using the SacreMoses[4] package. During the second stage of filtering, we completely remove whole examples, when either the French or the German sentence contain noise or inconsistent information.

1. **Basic Filtering.** We remove examples where either side is empty, or the two sides contain the same lower-cased text.

2. **Language Filtering.** Here we intend to identify sentences that are written in languages other than French and German. We remove examples where the language of either sentence does not match the expected one. To predict the language of a sentence, we use a pre-trained language detection model from FastText (Joulin et al., 2016).

3. **Length Filtering.** Here we aim to identify sentences or examples with unnatural length characteristics that potentially result from noise. We remove examples where either side is found to have a large number of words (greater than 200), an extreme character-to-word ratio (lower than 1.5 or greater than 12), at least one word with a high number of characters (greater than 25), or the example has an extreme source-to-target word ratio (lower than 0.4 or greater than 2.5). For setting the boundaries of what is an acceptable length or ratio, we follow (Shi et al., 2020).

4. **Alignment Filtering.** At this point, we want to identify noisy pairs by computing their alignment scores. We use the fast-align library (Dyer et al., 2013) and remove examples where the alignment score is 2.5 times above the average alignment score of the corpus. The alignment score of an example is calculated as the normalized log-probability of the German-French alignment, and the alignment score of the corpus is the sum of the alignment scores of its examples. Specifically for the WikiMatrix pairs, which are not human-generated and possibly contain more noise, we follow a more aggressive approach and remove a pair if its alignment score is 15 absolute points above the average.

[4] https://github.com/alvations/sacremoses

Dataset	Size (thousands)	
	Original	Filtered
Europarl	1,803	1,480
Paracrawl	7,223	5,893
Common Crawl	622	523
News Commentary	304	236
Wiki Titles	1,007	134
Title Rapid	983	849
EESC	2,844	2,392
WikiMatrix	2,807	1,936
TED Talks	292	279
Total	**17,885**	**13,722**

Table 1: Training sets before and after filtering.

The size of the clean corpus can be found in the second column of Table 1.

4 System Description

In this section we are going to describe the two main steps of our submission, fine-tuning of a pre-trained with the provided data and ensemble of pre-trained and not pre-trained models.

4.1 Multilingual fine-tuning

Traditionally, models are trained from random initialization. We initialize our model with mBART50 (Tang et al., 2020) pre-trained weights. These weights act as a more informed initialization that already contains useful features for language representation. Given the support of the public model to the French and German languages, no modifications of the embedding model were needed. Following mBART50 strategy, we fine-tune all the layers of the multilingual model on all the filtered French-German and German-French data. To condition the language generation (Johnson et al., 2017), a source language token was added as the first token of the source sentence, and a target language token was added as the first decoder token to the decoder.

Implementation Both randomly initialized baseline and mBART50 models were trained using *fairseq*'s (Ott et al., 2019)[5] mBART large imple-

[5] https://github.com/pytorch/fairseq

mentation for multilingual fine-tuning. The architecture consists of 12 layers with 1024 embedding size, 4096 feed-forward size, and 16 attention heads, both for encoder and decoder, with a total number of parameters of 610, 878, 464. Models were trained for approximately 400k updates or seven epochs using validation loss as an early stopping criterion, with a learning rate of $1 * 10^{-4}$ and $3 * 10^{-5}$ for the baseline and fine-tuned model, respectively. Both models are trained using the original vocabulary of 250k tokens, at subword level using the *sentencepiece*[6] model available with the mBART50 model. Both models were trained on two Nvidia GTX 3090 with eight batches of gradient accumulation. At generation time, beam size was set to eight.

4.2 Model ensemble

Model ensembling is a popular technique to leverage the features learned by several models. This is especially important in our case as the pre-trained model has been trained on data not constrained to the domain. As the pre-training step is performed on data that does not belong to the task domain, it could generate structures or patterns that are not commonly used, even when keeping the sentence's intended meaning. In order to balance the provided data, we ensemble the best checkpoint from the baseline and the fine-tuned mBART. Thus, next token prediction during inference is done according to the combined probability from both models.

Implementation Ensembling was performed using the standard *fairseq* generation script. The two models ensembled were the same checkpoints at 400k updates reported for the individual systems. Beam size was set to eight as in the previous experiments.

5 Experiments and Results

Multilingual fine-tuning. Our first hypothesis is that the use of pre-trained models could improve translation performance. Therefore, we compare our system with a baseline system with the same vocabulary and parameter configuration with random initialization to measure the impact of the translation step. Tables 2 and 3 show the translation results for German-French and French-German translation directions, respectively. Results show that fine-tuning of pre-trained models improves translation quality in both directions by 2.3 BLEU points on average, showing that a more informed model initialization significantly impacts the final model performance. It is worth noticing that we expected the fine-tuning approach to converge faster than the randomly initialized baseline, but they both show similar behaviors and required approximately 400k training updates. Figure 1 show that the fine-tuned model's validation loss is lower over the entire training but both converge to the best value at the seventh epoch.

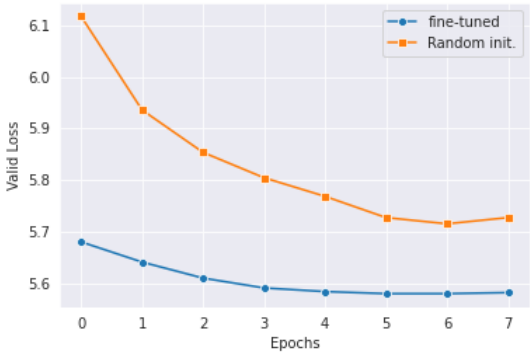


Figure 1: Validation loss during training for the fine-tuned mBART (fine-tuned) and randomly initialized (Random init.) models.

	BLEU	**ΔBLEU**
Baseline	28.98	-
mBART50	**31.69**	2.71
+Ensemble	**31.69**	0.00

Table 2: Results measured in BLEU for the German to French translation direction

Model Ensemble. Our second hypothesis is that the pre-training step on the out-of-domain data may affect the model's phrasing at inference time, and ensembling with the baseline trained only on the provided constrained data could improve its performance. Results show that, although a minor improvement of 0.3 BLEU points has been reported for the French to German translation direction, it is not consistent on German to French, where no performance difference has been observed. These results may indicate that the pre-training step has a limited impact on the final domain performance and that the fine-tuning step on the provided constraint data is the most crucial factor in the final model's domain adaptation.

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

	BLEU	ΔBLEU
Baseline	21.73	-
mBART50	23.63	1.90
+Ensemble	**23.93**	0.30

Table 3: Results measured in BLEU for the French to German translation direction

6 Conclusions

This work describes the TALP-UPC system for the WMT 2021 shared news translation task for French-German and German-French. Experimental results show that pre-trained models help improve translation performance in this kind of scenario, even for high-resource language pairs with millions of parallel sentences available, with 2.71 points for German-French translation direction and 1.90 points for French-German. Results also show that ensembling of pre-trained and randomly initialized models can lead to minor performance improvements (up to 0.3 BLEU) but not consistently on both tested languages.

In future work, better results may be obtained by combining fine-tuned pre-trained models with traditional back translation. Both techniques would benefit from the additional monolingual in two different aspects of the NMT model, initialization and additional training on the monolingual data provided.

Acknowledgments

This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 947657).

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. *ArXiv*, abs/1806.03280.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

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

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multilingual unsupervised NMT using shared encoder and language-specific decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3083–3089, Florence, Italy. 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.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao. 2020. OPPO's machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 282–292, Online. 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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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*, volume 30. Curran Associates, Inc.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

CUNI Systems in WMT21: Revisiting Backtranslation Techniques for English-Czech NMT

Petr Gebauer and **Ondřej Bojar** and **Vojtěch Švandelík** and **Martin Popel**
Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics,
Prague, Czechia
pgebauer27@seznam.cz bojar@ufal.mff.cuni.cz
vsvandelik@matfyz.cz popel@ufal.mff.cuni.cz

Abstract

We describe our two NMT systems submitted to the WMT2021 shared task in English-Czech news translation: CUNI-DocTransformer (document-level CUBBITT) and CUNI-Marian-Baselines. We improve the former with a better sentence-segmentation pre-processing and a post-processing for fixing errors in numbers and units. We use the latter for experiments with various backtranslation techniques.

1 Introduction

In this paper, we describe our two NMT systems submitted to the WMT 2021 English-Czech news translation shared task: "CUNI-DocTransformer" (Charles University document-level Transformer) and "CUNI-Marian-Baselines". In addition, we submitted also "CUNI-Transformer2018", which is exactly the same system (sentence-level) as submitted in 2018 (Popel, 2018).

CUNI-DocTransformer uses the same model as submitted last year (Popel, 2020), but with improved sentence segmentation (Section 3.1) and number-unit postprocessing (Section 3.2). This system was submitted for both English→Czech and Czech→English.

CUNI-Marian-Baselines is an attempt at reimplementation of the original CUNI-Transformer2018 in Marian (Junczys-Dowmunt et al., 2018), where we experiment with various setups of tagged backtranslation (Section 4). This system was trained only for English→Czech.

According to automatic evaluation provided by the WMT organizers (Table 1), CUNI-DocTransformer is the third best English→Czech system.

	cased BLEU		chrF
system	**ref A**	**ref B**	**ref A**
Facebook-AI	**24.80**	**(1) 22.69**	**(1) 0.5358**
Online-W	23.02	(2) 21.57	(2) 0.5285
CUNI-DocTransformer	22.19	(3) 19.85	(3) 0.5170
CUNI-Transformer2018	21.63	(4) 19.67	(4) 0.5091
eTranslation	21.03	(5) 19.38	(5) 0.5063
Online-A	20.16	(7) 18.18	(7) 0.4989
CUNI-Marian-Baselines	20.09	(6) 18.29	(6) 0.4992
Online-B	20.04	(8) 17.90	(8) 0.4956
Online-Y	18.13	(9) 16.13	(9) 0.4807
Online-G	15.30	(10) 13.87	(10) 0.4570

Table 1: Evaluation of English→Czech WMT21 systems. The systems are ordered by BLEU with reference A, ordering by the other metrics is provided in parentheses. Names of systems described in this paper are in bold.

2 Common settings

Both our systems use the Transformer (Vaswani et al., 2017) architecture, checkpoint averaging (using the last 8 checkpoints) and a 32k joint English-Czech subword vocabulary. Both systems are trained on CzEng 2.0 (Kocmi et al., 2020) with 61M authentic parallel and 127M synthetic (back-translated) sentences (see Table 2), but the English→Czech CUNI-DocTransformer does not use directly the EN-mono section,[1] while CUNI-Marian-Baselines uses all three sections including EN-mono (i.e. using forward-translation).

Both systems use Block-backtranslation (Popel et al., 2020), although CUNI-Marian-Baselines uses too small block size, so it does not have the expected positive effect as described in Section 4.

3 DocTransformer improvements

3.1 Sentence segmentation

CUNI-DocTransformer was trained on multi-sentence sequences of up to 3000 characters and

[1]The synthetic data in CzEng 2.0 were prepared using iterated backtranslation, so the EN-mono data were used for training a Czech→English system, which produced the English translation of the CS-mono data in CzEng 2.0. Thus, indirectly also the EN-mono data were used for training the English→Czech CUNI-DocTransformer.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 123–129
November 10–11, 2021.

data set	sentence pairs (M)	words (M) EN	CS
authentic	61	617	702
EN-mono (NewsCrawl 2016–2018)	76	1296	1474
CS-mono (NewsCrawl 2013–2018)	51	700	833
total	188	2613	3009

Table 2: Training data sizes (in millions). All the data are taken from CzEng 2.0.

750 subwords. However, the WMT submission format requires a segment-level alignment and also the CUNI-DocTransformer decoding employs overlapping sequences where sentence alignment is needed (for details see Popel (2020)). Thus, the sentences within a sequence are separated with a special token on both source and target side (both during training and at inference time), which allows a simple extraction of the sentence alignment.[2]

Some segments in the WMT input format contain multiple sentences. When treating such segments as a single sentence, the resulting translations often missed sentence-initial capital letters because there were almost no such examples in the training data, where multiple sentences would not be separated by the special token.

We thus decided to first split the input segments into sentences using UDPipe (Straka et al., 2016). Unfortunately, UDPipe tends to over-segment.[3] Such over-segmentation may lead to serious errors in the translation, even when using the document-level model. We thus restrict the sentences boundaries detected by UDPipe only to boundaries after sentence-final punctuation, using a simple rule-based segmenter from Udapi (Popel et al., 2017). This improved BLEU on our dev set slightly.

3.2 Number-unit post-processing

We noticed three types of translation errors related to numbers and units.

1. Attempt at converting numbers and units. For example, the Czech sentence *Je vysoký pouhých 190 cm* (meaning *He's only 190 cm tall*) was translated as *He's only six feet tall*. Note that six feet is 183 cm, so the translation was not exact.

2. Converting units without numbers. For example, *27 Kč* was translated as *$27*, while the correct translation should be *27 crowns* or *27 CZK*.

3. Not converting separators. English uses commas (or thin spaces) as thousand separators and dots as decimal separators, but Czech uses the opposite convention (with space being a more common thousand separator than dot). So e.g. Czech *179,500 kg* means 179 and a half kg (with precision up to 1 gram) and the correct translation to English should be *179.500 kg*, but CUNI-DocTransformer (and many other systems) keeps the phrase untranslated, resulting in a thousand times higher value.

The first type is quite rare – 0.7% of numerical expressions with units in cs-en and 0.6 in en-cs, according to Table 3, while some of these cases may be correct translation (correctly converted number and unit). The second type is more frequent – 11.1% and 6.5%, respectively. The third type is also frequent – in 100,000 Czech sentences from CzEng 2.0 cs-mono, there were 2594 numbers with a separator and out of these 275 (10.6%) were not correctly converted in the English CUBBITT translations; similarly in 100,000 English sentences in en-mono, there were 4376 numbers with a separator and out of these 263 (6.0%) were not converted in the Czech CUBBITT translations. We have noticed all three types of errors not only in CUBBITT, but we have not inspected these other MT systems in detail yet.

We implemented a rule-based tool which tries to fix such errors in post-processing.[4] It detects imperial/SI units of length, weight, speed, area and volume; units of temperature (Fahrenheit/Celsius) and currencies (USD, CZK, EUR), but it can be easily extended. By default, it keeps the units and numbers the same (except for the thousand/decimal separators), but it can be configured to convert the units and numbers. We had to deal with several edge cases, such as various ways how to write numbers and units or handling multiple numbers in a sentence with a possibly changed word order (using a word aligner).

[2] If the number of special tokens on the source side does not match the number of special tokens on the target side at inference time, we back off to translating each sentence in a given sequence independently.

[3] UDPipe is trained on Universal Dependencies (Zeman et al., 2018), where titles and headlines with no final punctuation are treated as sentences, which need to be detected by the sentence segmentation.

[4] https://github.com/vsvandelik/cubbitt-fixer

	kept		cs-en		en-cs	
	number	**unit**	#	**%**	#	**%**
A	yes	yes	2 689	86.5	3 548	85.7
B	yes	no	346	**11.1**	268	**6.5**
C	no	yes	21	**0.7**	24	**0.6**
D	no	no	21	0.7	13	0.3
E	detection failure		31	1.0	287	6.9
	total		3 108	100.0	4 140	100.0

Table 3: Automatic analysis of numerical expressions with units in a sample of 100 000 sentences from the synthetic parts of CzEng 2.0. Numerical expressions that were detected only in the source sentences, but not in the (MT) translation, are marked as *detection failure*. Cases B and C where only the unit or only the number were converted can be safely considered as errors – so the percentages are marked in bold.

Using our tool, we analyzed a sample of the synthetic training data in CzEng 2.0 and found out that at least 11.8% of Czech and 7.1% of English expressions with numbers and units are translated wrong, see Table 3.

After submitting CUNI-DocTransformer, we analyzed the WMT2021 news test sets and found out that there were only 4 sentences affected by our post-processing. All 4 cases were of the same type – "korun" was translated as "$", which was corrected to "crowns",

4 Experiments in Marian

The goals of the experiments described in this section were:

- Reimplement the Block-backtranslation training (Popel et al., 2020) in Marian (Junczys-Dowmunt et al., 2018). Block-backtranslation was first implemented in the Tensor2Tensor framework in the CUBBITT system, also known as CUNI-Transformer2018 (Popel, 2018).
- Explore the effect of Block-backtranslation (vs. standard shuffled backtranslation (Sennrich et al.)), checkpoint averaging and Tagged backtranslation (Caswell et al., 2019).
- Try a novel type of Tagged backtranslation with tags on the target side.
- Explore interactions of the above-mentioned methods.

4.1 Marian settings

We followed the standard Transformer Big hyperparameters, with 6 encoder and 6 decoder layers (unlike CUNI-DocTransformer, which has 12 encoder layers). Other differences from CUNI-DocTransformer are: Marian was trained on sentences (no document level) of up to 150 subwords (`--max-length 150`). It was trained on a single GPU (instead of 8), but using 8 batches per updated (`--optimizer-delay 8`), thus resulting in a similar effective batch size. Due to time reasons we trained all our Marian models just for a single epoch on the whole CzEng 2.0 training data, containing all three parts: authentic parallel data, synthetic CS-mono and synthetic EN-mono, i.e. using both backtranslation and forward translation (Ueffing et al., 2007; Kim and Rush, 2016). The English→Czech CUNI-DocTransformer was not trained on the EN-mono part, but it was trained "until convergence", for 700k updates (which is not easily converted to epochs because the authentic data was upsampled for the Block backtranslation), i.e. several times more updates that the Marian model. Finally, we accidentally used too small blocks in the Block backtranslation, as described in the following section.

4.2 Replicating CUBBITT

In our first experiment, we tried to replicate the CUNI-Transformer2018, which also uses the Transformer Big hyperparameters (with 6 encoder and 6 decoder layers) and sentence-level training. Our Marian results were about 1.5 BLEU worse on various WMT dev sets on average, which is better than we expected when training for a single epoch only. According to our preliminary experiments, including forward-translation data (EN-mono in our case) makes the initial training faster (i.e. better BLEU after the first epoch), although it does not improve the final BLEU when training until convergence. Forward translation data are great for fast uptraining and model distillation – the newly trained model is being trained to behave similarly as the original model used to produce the synthetic translations. The synthetic translations are consistent (if no noising is used) – the same sentence is translated always the same way.

While the final BLEU results are good enough, the learning BLEU curves on Figure 1 do not show the camel-shape progress typical for Block Backtranslation Popel et al. (2020). We also did not

observe the synergy effect of Block backtranslation and checkpoint averaging. The explanation is simple: when dividing the data for Block backtranslation, we accidentally used 10 blocks of authentic data and 20 blocks of synthetic data. Thus there was less than one checkpoint per each block on average, which goes against the main idea of Block backtranslation, where each block of authentic or synthetic data should be big enough to fit at least 8 checkpoints (considering checkpoint averaging with 8 checkpoints). We think this is the reason why we do not see any significant differences between *block* and *shuffled* in Tables 4 and 5 and also between these two tables (as an effect of checkpoint averaging).

4.3 Tagged backtranslation

For our experimenting, we decided to try labeling the data based on its authenticity — The labels would have two parts, one specifying whether the source side was an authentic sentence, or created using back translation or forward translation (Ueffing et al., 2007; Kim and Rush, 2016), and the other part specifying the same for the target side. We tried having no labels at all, labeling only one side or the other, or labeling both sides. However, in all these scenarios, every label that existed specified the authenticity of both the source and the target side. This is very similar to tagged backtranslation (Caswell et al., 2019) but we tried using our labels on the target side as well and we explored possible synergy between block ordering or checkpoint averaging by trying the different versions.

Since all of czeng20-train, czeng20-enmono, czeng20-csmono were used, the labels were `auth+auth`, `auth+synth` and `synth+auth`. In each dataset, where the label was present, it was situated at the beginning of each sentence, space-separated from the sentence itself.

In addition to the main experimenting with the four variants of source and target side labeling, we created versions with data ordered in blocks (of authentic vs backtranslated data). This resulted in eight versions being trained — all combinations of: source side labeling yes/no, targets side labeling yes/no, block order / completely shuffled data.

When the training data was ordered into blocks, there were about 10 blocks of each of the data kinds (czeng20-train, czeng20-csmono, czeng20-enmono) meaning 30 blocks in total. With our checkpoint frequency this meant that one block was slightly smaller than the data seen between two neighboring checkpoints, which are very small blocks. The completely shuffled datasets were created from the block ones by shuffling them using a random permutation. The order of data points was the same among all block-ordered datasets and same among all completely shuffled datasets.

For time reasons, we only managed to train each model on a single epoch (using marian's `--after-epochs 1`). From the training, we obtained eight variants, which we then did checkpoint averaging on, creating additional eight variants. We then evaluated these 16 variants on the wmt17 newstest dataset, and chose two representing models for each — one was the model at the end of the training, the other was the model that achieved the best BLEU score on wmt17. We evaluated these 32 models on concatenation of wmt15, wmt16 and wmt18 and chose those seven models that reached the best BLEU on this testset.

4.4 Results

We observed some differences in performance among the trained versions. The images below show the development of BLEU score (measured on a test set, not the training data) as the training progressed. We can see that there are differences among the versions but it is hard to find a pattern in them. They also do not seem to be consistent among the test sets — wmt15, wmt16, wmt17, wmt18. When wmt15, wmt16 and wmt17 are concatenated, the differences seem to largely disappear (see the tables below) and we still do not see any clear pattern in the results.

We also fail to see clear differences in performance between block ordering vs. completely shuffled corpora, and checkpoint averaging vs. no averaging. There is also no synergy between those two in our results, which is very likely caused by our setup of extremely small blocks. The blocks used in CUBBITT were large enough to contain all eight averaged checkpoints of certain models, while our blocks didn't even fully contain one checkpoint.

5 Conclusions

In this paper, we presented two sets of experiments: automatic correction of numeric expressions with units in rule-based post-processing and various settings of Tagged backtranslation.

The correction of numeric expressions with units focuses on errors which are relatively rare and do

Source labeling	Target labeling	Ordering	best-BLEU	final-BLEU
yes	yes	block	27.3	27.4
yes	yes	shuffled	27.3	27.2
yes	no	block	27.2	27.4
yes	no	shuffled	27.2	27.3
no	yes	block	27.2	27.2
no	yes	shuffled	27.4	27.4
no	no	block	27.3	27.3
no	no	shuffled	27.5	27.5

Table 4: Both BLEU scores shown were measured no the concatenation of wmt15, wmt16 and wmt18. best-BLEU is the score of the model that achieved the best BLEU on wmt17, while final-BLEU is the BLEU of the model at the end of the training. All of the models in this table are without checkpoint averaging.

Source labeling	Target labeling	Ordering	best-BLEU	final-BLEU
yes	yes	block	27.4	27.4
yes	yes	shuffled	27.2	27.2
yes	no	block	27.5	27.5
yes	no	shuffled	27.2	27.2
no	yes	block	27.3	27.3
no	yes	shuffled	27.4	27.4
no	no	block	27.3	27.3
no	no	shuffled	27.5	27.5

Table 5: This table contains the BLEU scores of models with checkpoint averaging. The columns are the same and have the same meaning as in the previous table.

Source labeling	Target labeling	Ordering	checkpoint averaging	point	wmt21 BLEU
yes	**no**	**blocks**	**yes**	**last**	**20.1**
yes	no	blocks	no	last	20.0
yes	no	blocks	yes	best	19.9
no	no	shuffled	no	last	19.9
no	no	shuffled	no	best	19.6
no	no	shuffled	yes	last	19.6
no	yes	shuffled	no	last	19.6

Table 6: These are the BLEU scores of the submited models on the wmt21 test set.

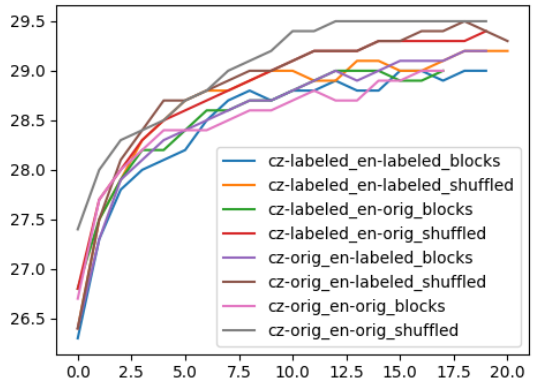


Figure 1: wmt16 BLEU training curves of averaged models

not affect automatic metrics such as BLEU much, but can result in serious misunderstanding of the meaning of the translation. Unfortunately, these errors won't be properly reflected even in the official WMT (context-sensitive, but sentence-level) manual evaluation, where each sentence's score is weighted the same, even if some errors are crucial for the meaning of the whole document.

The experiments with Tagged backtranslation using a Marian reimplementation of CUBBITT did not show any substantial differences in the results nor any consistent pattern. However, we hope that future work continuing the research on various types of training data (authentic vs. synthetic; forward vs backward; different domains) and their synergies may bring new results and better understanding of the backtranslation training etc.

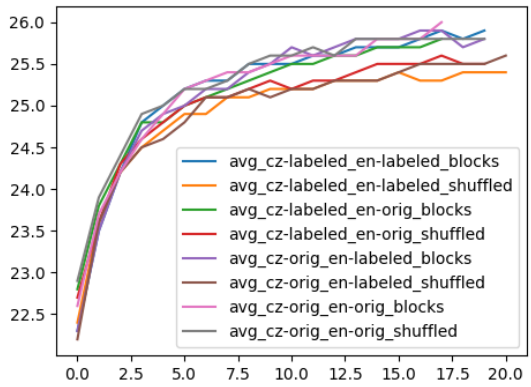


Figure 2: wmt17 BLEU training curves of averaged models

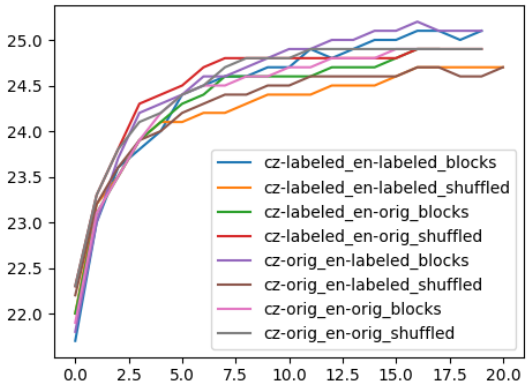


Figure 3: wmt18 BLEU training curves of averaged models

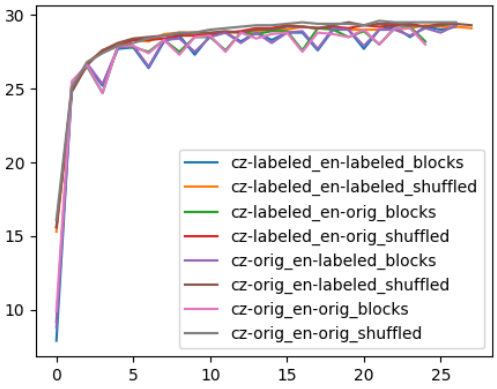


Figure 4: wmt16 BLEU training curves of non-averaged models

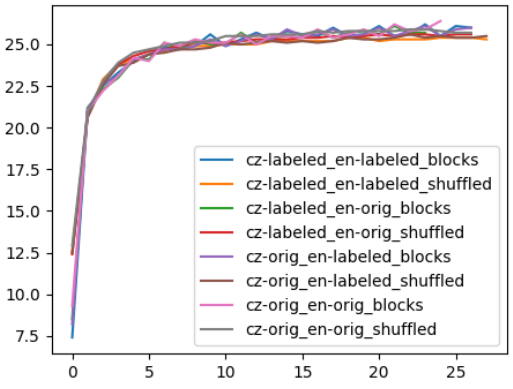


Figure 5: wmt17 BLEU training curves of non-averaged models

Acknowledgements

The work was supported by the grants 19-26934X (NEUREM3) and 20-16819X (LUSyD) by the

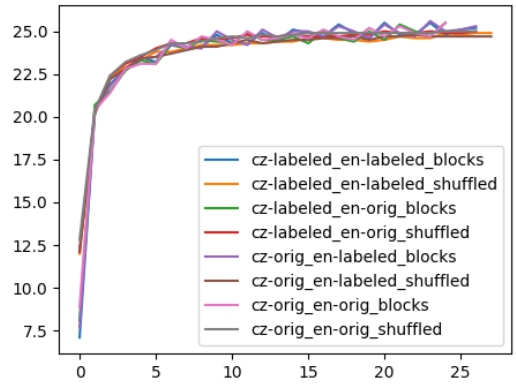


Figure 6: wmt18 BLEU training curves of non-averaged models

Czech Science Foundation. The work has been using language resources developed and distributed by the LINDAT/CLARIAHCZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. 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.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Martin Popel. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel. 2020. CUNI English-Czech and English-Polish systems in WMT20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation*, pages 269–273, Online. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch.2016b. Improving neural machine translation-models with monolingual data.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Semi-supervised model adaptation for statistical machine translation.

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.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Ensembling of Distilled Models from Multi-task Teachers for Constrained Resource Language Pairs

Amr Hendy[1], **Esraa A. Gad**[1], **Mohamed Abdelghaffar**[1], **Jailan S. ElMosalami**[1], **Mohamed Afify**[1], **Ahmed Y. Tawfik**[1] **and Hany Hassan Awadalla**[2]
[1] Microsoft Egypt Development Center, Cairo, Egypt
[2] Microsoft Corporation, Redmond, WA, USA
{amrhendy,v-egad,mohamed.abdelghaar,v-jailanel}@microsoft.com
{mafify,atawfik,hanyh}@microsoft.com

Abstract

This paper describes our submission to the constrained track of WMT21 shared news translation task. We focus on the three relatively low resource language pairs Bengali ↔ Hindi, English ↔ Hausa and Xhosa ↔ Zulu. To overcome the limitation of relatively low parallel data we train a multilingual model using a multitask objective employing both parallel and monolingual data. In addition, we augment the data using back translation. We also train a bilingual model incorporating back translation and knowledge distillation then combine the two models using sequence-to-sequence mapping. We see around 70% relative gain in BLEU point for $En \leftrightarrow Ha$ and around 25% relative improvements for $Bn \leftrightarrow Hi$ and $Xh \leftrightarrow Zu$ compared to bilingual baselines.

1 Introduction

Neural machine translation (NMT) witnessed a lot of success in the past few years especially for high resource languages (Vaswani et al., 2017). Improving the quality of low resource languages is still challenging. Some of the popular techniques are adding high resource helper languages as in multilingual neural machine translation (MNMT) (Dong et al., 2015; Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017; Arivazhagan et al., 2019), using monolingual data including pre-training (Liu et al., 2020), multi-task learning (Wang et al., 2020), back translation (Sennrich et al., 2016) or any combination of these methods (Barrault et al., 2020) and system combination of multiple systems (Liu et al., 2018).

This paper describes the Microsoft Egypt Development Center (EgDC) submission to the WMT21 shared news translation task for three low resource language pairs (six directions), Bengali ↔ Hindi ($Bn \leftrightarrow Hi$), English ↔ Hausa ($En \leftrightarrow Ha$) and Xhosa ↔ Zulu ($Xh \leftrightarrow Zu$). We focus on the constrained track because it is easier to compare different systems and it is always possible to improve performance by adding more data. The main features of our approach are as follows:

- Using a recently proposed multitask and multilingual learning framework to benefit from monolingual data in both the source and target languages (Wang et al., 2020).
- Using knowledge distillation (Freitag et al., 2017) to create bilingual baselines from the original multilingual model and combining it with the multilingual model.

The paper is organized as follows. Section 2 gives an overview of the data used in the constrained scenario, followed by section 3 that gives a detailed description of our approach. Section 4 presents our experimental evaluation. Finally, our findings are summarized in Section 5.

2 Data

Following the constrained track, we use bitext data provided in WMT21 for the following pairs: Bengali ↔ Hindi, English ↔ Hausa, Xhosa ↔ Zulu and English ↔ German. Statistics of the parallel data used for the three pairs in addition to the German helper are shown in Table 1. We also use monolingual data for all previously mentioned languages provided in WMT21 for techniques such as multi-task training and back-translation. Statistics of the monolingual data used for the 6 languages in addition to the German helper are shown in Table 2. For very low resource languages, Hausa, Xhosa and Zulu, we use all the available monolingual data, e.g. NewsCrawl + CommonCrawl + Extended CommonCrawl for Hausa, and Extended CommonCrawl for both Xhosa and Zulu. For relatively high resource languages, Bengali, Hindi, English and German, we only use a subset of the provided data mostly from NewsCrawl due to its high-quality. In addition to the NewsCrawl monolingual subset, we add a sampled subset from CommonCrawl to

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 130–135
November 10–11, 2021.

Language pair	# of sentences
Bengali ↔ Hindi	3.36M
English ↔ Hausa	750K
Xhosa ↔ Zulu	94K
English ↔ German	84.8M

Table 1: Bitext data used for bilingual and multilingual systems. For each language pair, we use all available sources released in WMT21

Language	# of sentences	
	Raw	Cleaned
Bengali	53.8M	53.3M
English	75M	73.5M
German	111.2M	109.9M
Hausa	10.8M	6.2M
Hindi	60.2M	59.8M
Xhosa	1.6M	950K
Zulu	2M	1.4M

Table 2: Monolingual data used for multi-task training and back-translation

avoid biasing into the news domain especially for Bengali ↔ Hindi and Xhosa ↔ Zulu whose target evaluation domain come from Wikipedia content.

2.1 Data Filtering

For Bengali, English, Hindi and German, we apply fastText [1] language identification on the monolingual data to remove sentences which are not predicted as the expected language. We do the same for Hausa, Xhosa and Zulu using Polyglot [2] because fastText does not cover these three languages. The resulting size of the monolingual data of each language is shown in Table 2.

3 System Architecture

The final MT system in each direction is an ensemble of two NMT models comprising a bilingual model (one for each of the six primary directions) and a multilingual model trained to provide translations for 8 directions (the six primary directions plus English ↔ German). The multilingual system uses a recently proposed multitask framework for training (Wang et al., 2020). We describe the individual systems in Subsection 3.1. This is followed by presenting our system combination techniques in Subsection 3.2. Finally we present the architecture of the submitted system highlighting our design decisions in Subsection 3.3.

3.1 Individual Systems

This subsection describes the individual systems and their training leading to the proposed system combination strategy in the following subsection. We first build bilingual models for the six primary directions using the data shown in Table 1 except the English ↔ German. These serve as baselines to compare to the developed systems. The models use a transformer base architecture comprising 6 encoder and 6 decoder layers and a 24K joint vocabulary built for Bengali ↔ Hindi, a 8K joint vocabulary built for English ↔ Hausa and a 4K joint vocabulary built for Xhosa ↔ Zulu using sentencepiece (Kudo and Richardson, 2018) to learn these subword units to tokenize the sentences. In addition to the baseline bilingual models, we use knowledge distilled (KD) data and back-translated (BT) data generated from a multilingual model to build another set of bilingual models for each of the six primary directions. This multilingual model is described below. The purpose of these models is to participate in the ensemble along with the multilingual models. The latter bilingual models follow the same transformer base architecture and joint vocabulary used in the baseline bilingual models.

The multilingual model combines the 8 translation directions shown in Table 1. These are the six primary directions plus English ↔ German as a helper. The latter is mainly used to improve generation on the English centric directions. The model uses a 64K joint vocabulary constructed using sentencepiece (Kudo and Richardson, 2018) from a subset of the monolingual data of each language as described in Section 2. The transformer model has 12 encoder and 6 decoder layers. In addition, a multitask objective is used during training to make use of monolingual data. The objective comprises the usual parallel data likelihood referred to as MT, a masked language model (MLM) at the encoder and a denoising auto-encoder (DAE) (similar to mBART (Liu et al., 2020)) at the decoder side. The latter two objectives help leverage monolingual data for both the encoder and the decoder sides. The three objectives are combined using different proportions according to a schedule during the training. Please refer to (Wang et al., 2020) for details.

[1] https://fasttext.cc/docs/en/language-identification.html
[2] https://github.com/aboSamoor/polyglot

To summarize we build the following models:

- Bilingual models trained using parallel data in Table 1 for the 6 primary directions. These are mainly used as baselines.
- Multilingual models trained using a multitask objective using parallel and monolingual data and comprising 8 directions.
- Bilingual models trained using KD and BT data generated using our best multilingual model. These are combined with the best multilingual model as described in 3.2.

3.2 System Combination

System combination or ensembling is known to improve the performance over individual systems. There are many ways to create an ensemble (Liu et al., 2018; Dabre et al., 2019). For example, individual models obtained from different checkpoints during the same training or by training models sharing the same vocab and architecture using different data or simply different random seeds can be combined using model averaging techniques. Here, we opt to combine different models since it generally leads to better performance because different models tend to be more complementary. To this end, we propose a simple and effective method to combine completely different architectures. The proposed method could be also used in conjunction with checkpoint and model averaging for further gains, but we haven't tried this in our experiments due to time limitations.

The basic idea of our combination is very simple. Assume we have the translation pair $x \rightarrow y$ where y is the reference translation. The output of model 1 is the pair $x \rightarrow y1$ and the output of model 2 is the pair $x \rightarrow y2$. This can be generalized to multiple systems but we limited our combination to only two models. We train a new model that takes the set of hypotheses (possibly augmented by the source sentence) from the two models to generate the target sentence. Thus this model combines the outputs of two models in the ensemble to produce a translation closer to the original target sentence i.e. $< HYP > y1 < HYP > y2 \rightarrow y$.We also experimented with adding the source to the input i.e. $< SRC > x < HYP > y1 < HYP > y2 \rightarrow y$ which led to around 0.3 BLEU improvement for $Ha \rightarrow En$, but we haven't tried on other pairs due to time limitation. All combination models use 6 layers encoder and decoder and a 64K vocabulary similar to the multilingual system. These combination models use the full bitext and dev data provided in WMT21 as shown in Table 1. The system combination is outlined in Figure 1. This ensembling technique can be thought of as providing both system combination and post-editing capabilities.

3.3 Overall System

Our overall system is depicted in Figure 2. The first module shows the data input where language identification (LID) is used to filter the monolingual data. As mentioned in Section 2.1 we use fastText and polyglot for LID depending on the language. We first build bilingual baselines which are not shown in the figure. Then as shown in the second module, we build 4 multilingual systems using different task objectives as follows: $MT, MT + MLM, MT + DAE$ and $MT + MLM + DAE$ trained on the 8 directions shown in Table 1 following the temperature-based strategy in (Arivazhagan et al., 2019) to balance the training data in different resource languages using T = 5. We pick the best system and use it to back translate the selected monolingual data. For most pairs, as detailed in Section 4, we find that $MT + DAE$ and $MT + MLM + DAE$ are quite close. Therefore, we use the $MT + DAE$ to do back translation for all submitted 6 pairs. We use beam search with beam size = 5 when generating the synthetic back-translated data. Once we get the back-translated data (called BT_1) we add it to our parallel and monolingual data and build a new multilingual model called $MT + DAE + BT_1$. We tag the back-translated data with <BT> tag at beginning of each source sentence so the model can differentiate between the genuine parallel and back-translated data quality. The resulting model is used to regenerate the back-translated data (called BT_2) and to knowledge distill the bitext (called KD). The latter two data sets are augmented and used to build a bilingual system (called $MT + KD + BT_2$). We upsample the KD data set and the upsampling ratio is selected based on parameter sweeping and validating the resulting improvement on the validation set. Finally, the latter bilingual model is combined with our final multilingual model using the method in Section 3.2 to create our submission.

4 Experimental Results

In this section, we describe the results of our intermediate and final systems. We report Sacre-

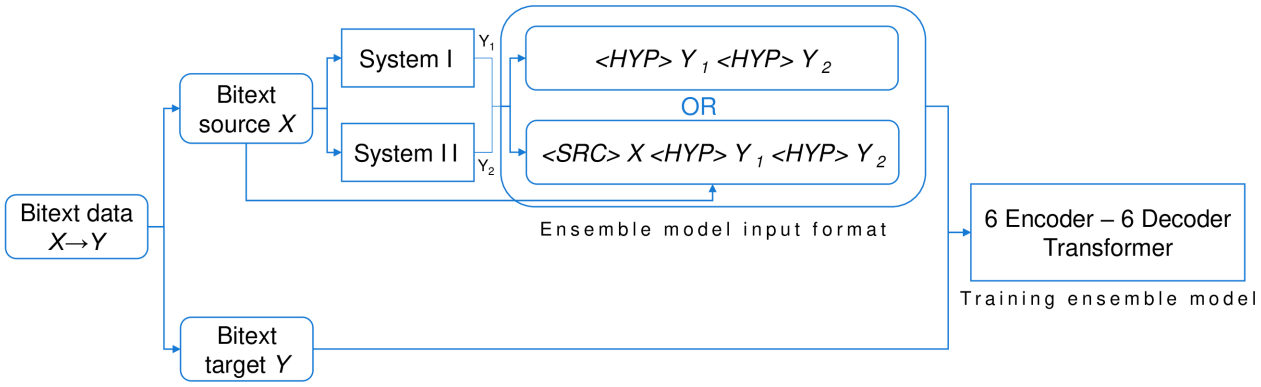


Figure 1: The system combination component used for our experiments.

BLEU (Post, 2018) on the validation set released in WMT21, and both SacreBLEU and COMET (Rei et al., 2020) using the available implementation [3] on the official test set released in WMT21. The results for the six submitted language pairs are in Tables 3-5. The first row in each table shows the bilingual baseline which performs relatively poor due to the limited amount of parallel data for each pair. This is followed by the four multilingual systems with different objectives. It is clear that adding a monolingual objective brings nice improvements for all language pairs. The $MT + DAE$ and $MT + MLM + DAE$ perform closely for all language pairs indicating that target monolingual data is most important. The next two rows show the results of adding back-translated data to the multilingual model and a bilingual baseline using back-translated and knowledge distilled data generated from the best multilingual model. As expected adding back translation brings significant improvement to all language pairs. Also using the multilingual model to create data for a bilingual model shows excellent results that outperform the multilingual model. Finally, the ensemble, as expected, performs better than the individual models. The significant difference between reported improvements in $Ha \leftrightarrow En$ and other directions shows the effectiveness of adding $De \leftrightarrow En$ parallel and monolingual data that helps English centric directions more than other directions. We evaluated the final submitted systems on the official test set released in WMT21 as shown in Table 6.

[3] https://github.com/Unbabel/COMET

System	Ha-En	En-Ha
bilingual baseline	14.10	13.78
multi. MT	14.32	13.16
+ MLM	16.18	13.94
+ DAE	18.05	14.91
+ MLM + DAE	17.35	15.03
multi. MT + DAE + BT_1	21.11	20.24
bilingual MT + KD + BT_2	24.43	20.68
ensemble	24.90	21.00

Table 3: Results of Ha-En and En-Ha systems. We report SacreBLEU scores on the validation set provided in WMT21

System	Bn-Hi	Hi-Bn
bilingual baseline	18.60	10.90
multi. MT	18.21	10.02
+ MLM	18.82	10.67
+ DAE	18.64	10.40
+ MLM + DAE	19.20	11.27
multi. MT + DAE + BT_1	20.18	12.29
bilingual MT + KD + BT_2	21.03	12.90
ensemble	21.20	13.30

Table 4: Results of Bn-Hi and Hi-Bn systems. We report SacreBLEU scores on the validation set provided in WMT21

5 Summary

This paper describes our submission to the constrained track of WMT21. We focus on the three relatively low resource language pairs $Bn \leftrightarrow Hi$, $En \leftrightarrow Ha$ and $Xh \leftrightarrow Zu$. To overcome the limitation of relatively low parallel data we train a multilingual model using a multitask objective recently proposed in (Wang et al., 2020). In addition,

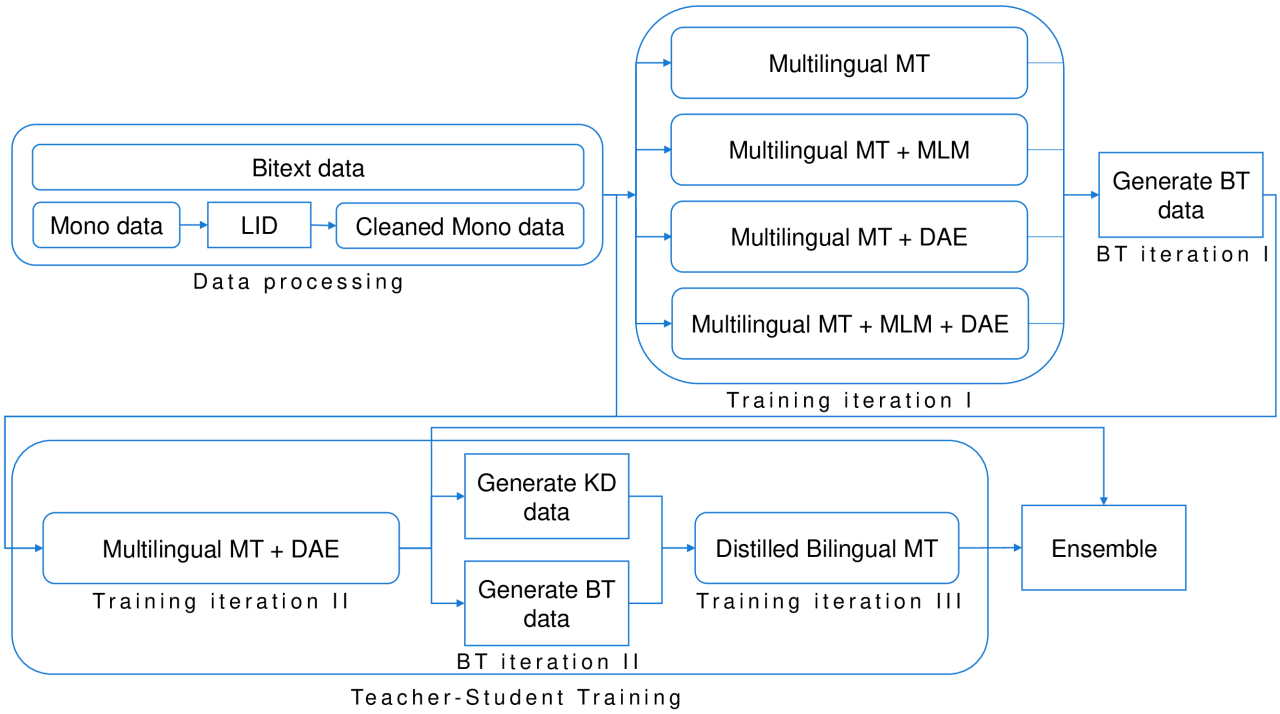


Figure 2: The overall system flow used for our experiments

System	Xh-Zu	Zu-Xh
bilingual baseline	8.00	7.60
multi. MT	7.53	7.47
+ MLM	7.23	7.02
+ DAE	8.53	8.24
+ MLM + DAE	8.20	7.80
multi. MT + DAE + BT_1	9.06	8.86
bilingual MT + KD + BT_2	9.80	9.17
ensemble	10.00	9.30

Table 5: Results of Xh-Zu and Zu-Xh systems. We report SacreBLEU scores on the validation set provided in WMT21

Translation direction	BLEU	COMET
$Ha \rightarrow En$	17.13	0.149
$En \rightarrow Ha$	16.13	0.086
$Bn \rightarrow Hi$	21.08	0.532
$Hi \rightarrow Bn$	10.93	0.411
$Xh \rightarrow Zu$	9.94	0.180
$Zu \rightarrow Xh$	9.25	0.299

Table 6: Results of the submitted systems. We report SacreBLEU and COMET scores on the official test set provided in WMT21. For COMET, we use the recommended model "wmt20-comet-da".

we augment the data using back translation. We also use the resulting multilingual model to create a bilingual model incorporating back translation and knowledge distillation. Finally, we combine the two models, using a flexible sequence-to-sequence approach, to yield our submitted systems. We see large gains up to 8-10 BLEU points for $En \leftrightarrow Ha$ and nice improvements of up to 2-3 BLEU points for $Bn \leftrightarrow Hi$ and $Xh \leftrightarrow Zu$.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Raj Dabre, Fabien Cromieres, and Sadao Kurohashi. 2019. Enabling multi-source neural machine trans-

lation by concatenating source sentences in multiple languages.

Daxiang Dong, Hua Wu, W. He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL*.

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.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder.

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.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Yuchen Liu, Long Zhou, Yining Wang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2018. A comparable study on model averaging, ensembling and reranking in nmt. In *Natural Language Processing and Chinese Computing*, pages 299–308, Cham. Springer International Publishing.

Matt Post. 2018. A call for clarity in reporting bleu scores.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034, Online. Association for Computational Linguistics.

Miðeind's WMT 2021 submission

Haukur Barri Símonarson, Vésteinn Snæbjarnarson,
Pétur Orri Ragnarsson, Haukur Páll Jónsson and Vilhjálmur Þorsteinsson
Miðeind ehf., Reykjavík, Iceland
{haukur, vesteinn, petur, haukurpj, vt}@mideind.is

Abstract

We present Miðeind's submission for the English→Icelandic and Icelandic→English subsets of the 2021 WMT news translation task. Transformer-base models are trained for translation on parallel data to generate back-translations iteratively. A pretrained mBART-25 model is then adapted for translation using parallel data as well as the last backtranslation iteration. This adapted pretrained model is then used to re-generate backtranslations, and the training of the adapted model is continued.

1 Introduction

Our work on machine translation for Icelandic has been going on for a couple of years as a part of the government sponsored Icelandic Language Technology Programme (Nikulásdóttir et al., 2020). By building on state-of-the-art solutions we have developed an open and effective translation system between Icelandic and English.

To achieve this, we collect parallel Icelandic and English texts which are filtered for good quality alignments. We also collect monolingual text for backtranslations. We follow tried and tested methods in neural machine translation using iterative backtranslation (Edunov et al., 2018) and adapt the multilingual denoising autoencoder model mBART-25 (Liu et al., 2020) for translation.

2 Datasets

We used several parallel and monolingual datasets, both publicly available and created in-house.

2.1 Parallel data

The parallel data used are ParIce (Steingrímsson et al., 2020) and the JW300 corpus (Agić and Vulić, 2019). In addition we used a parallel student theses and dissertation abstracts corpus, IPAC, generated in-house and sourced from the Skemman repository[1] as described in (Símonarson and Snæbjarnarson, 2021). A breakdown of the data is shown in Table 1.

Corpus	#Sentences
The Bible	33k
EEA regulatory texts	1,700k
EMA	404k
ESO	12.6k
OpenSubtitles	1,300k
Tatoeba	10k
Jehova's Witnesses (JW300)	527k
Other*	93k
IPAC	64k

Table 1: Parallel corpora used. #Sentences are the number of sentence pairs. *Other** denotes software localizations, Project Gutenberg literature and the Icelandic sagas.

Following (Pinnis, 2018) we apply simple heuristic filters to the parallel data, mainly for capturing OCR and PDF errors, and correcting or removing character encoding errors after deduplication. Filters include but are not limited to: empty sequence removal, length cut-offs, character whitelists, mismatch in case and symbols between languages, edit-distances between source and target, normalizing of punctuation, and ad-hoc regular expressions for Icelandic specific OCR/PDF errors. For a more in-depth description see (Jónsson et al., 2020).

Other potential parallel datasets are ParaCrawl (Bañón et al., 2020) and CCMatrix (Schwenk et al., 2021). Manual review of a couple of hundred randomly chosen lines from ParaCrawl revealed that the data quality is quite low for Icelandic, many lines are machine translated or badly aligned. We therefore did not include ParaCrawl. CCMatrix did not exist when the project started and we have not

[1] https://skemman.is

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 136–139

taken the time to review and integrate it although a quick inspection indicates that the quality is higher than in ParaCrawl.

2.2 Data used for backtranslation

We collected and translated monolingual data for backtranslations, made available in (Símonarson et al., 2020), mostly building on the work in (Edunov et al., 2018). The English sentences (44.7m) are retrieved from the Wikipedia, Newscrawl and Europarl corpora. The Icelandic sentences (31.3m) are sourced from the Icelandic Gigaword Corpus (Steingrímsson et al., 2018).

Lang.	Name	#Sentences
IS	Court rulings	1.8M
IS	Supreme court rulings	2M
IS	Laws	814k
IS	Web of Science	268k
IS	Wikipedia	405k
IS	Parliamentary proc.	6.2M
IS	Misc	350k
IS	Newspaper (Mbl)	13.6M
IS	Newspaper (Visir)	4.9M
IS	Radio transcripts	1M
EN	Newscrawl	33.4M
EN	Wikipedia	9.3M
EN	Europarl	2.0M

Table 2: Monolingual data used for backtranslation.

3 Training of small transformer models

Our earlier models were trained using the transformer-base configuration described in (Vaswani et al., 2017) as implemented in Google's Tensor2Tensor (TensorFlow-based) (Vaswani et al., 2018) library. For later models we switched to Facebook's Fairseq (Ott et al., 2019) library. An improved translation task was implemented in Fairseq to include BPE dropout; it is available in the `greynirseq`[2] library.

The transformer-base models were trained iteratively and used to generate new backtranslations. We stopped when each language direction had been trained on backtranslations that were produced by a model that had itself seen backtranslations at training time. We compared tagged and untagged backtranslations, sampling versus beam search and different mixing ratios (upsampling rate) between parallel and backtranslated data. Using tagged backtranslations as opposed to no tag showed an improvement from 16.5 to 17.5 BLEU[3] after the first iteration over the IPAC development set, while using no backtranslations gave 15.0, so we proceeded to use tagged translations.

Model	BLEU
Transformer-base	16.5
Transformer-base + bt	17.5
Transformer-base + iterative-bt	18.5
mBART (first run)	23.1
mBART (continued)	23.6

Table 3: BLEU scores over IPAC for the EN-IS direction.

We use the IPAC test set to measure BLEU since it was available, has a large range of topics (although maybe not a large range of style) and is very unlikely to be accidentally included in the training data. The IPAC data is out of distribution with the rest of the training data but we do not consider that to be a problem since our goal is a general purpose model. The WMT dev set did not exist at the time.

We used a joint BPE vocabulary of size 16k and shared input-output embedding matrices. We pretokenized the input using `tokenizer` [4] for the Icelandic side and spaCy (Honnibal et al., 2020) for the English side. A beam width of 4 was used for beam search during backtranslation. Each training iteration took approximately one week on a single GTX 1080 graphics card. We were pleasantly surprised with how far we got with only this modest hardware.

3.1 Translation mixing ratio selection and beam noise

We assessed the impact of the ratio of synthetic backtranslation data to authentic parallel data on translation performance. Best results were obtained with a 1:2 ratio of authentic to synthetic data, using IPAC (held out from training) for evaluation.

For noising the backtranslation beam outputs, we follow (Edunov et al., 2018) and used within-k permutation of whole words (with k=3), whole-word masking, and word dropout. Using sampling and noised beam outputs yielded comparable results.

[2] https://github.com/mideind/greynirseq

[3] SacreBLEU signature: `BLEU+case.mixed+lang.en-is+numrefs.1+smooth.exp+tok.13a+version.1.5.1`

[4] https://github.com/mideind/Tokenizer

4 Adapting mBART-25 for translation

The mBART-25 (Liu et al., 2020) (610M parameters) language model is far larger than the Transformer-base model (110M parameters). It was pretrained on 25 languages, including English and Swedish, but not Icelandic. We adapt it for translation from Icelandic to English and vice versa, using the same human-derived parallel translation data as for the transformer-base model along with the synthetic backtranslated corpus in a ratio of 1:2. We do not use any pre-tokenization and inherit the BPE sententencepiece vocabulary from mBART-25 (of size 250k) with the addition of an Icelandic language marker that was randomly initialized. We use the same hyperparameters as in (Liu et al., 2020) and the implementation from Fairseq (Ott et al., 2019). The models are trained until their performance on the development sets plateaus.

The initial learning rate was set to 3e-4. Sixteen 32GB nVidia V100 GPUs connected with Infiniband were used for training. The effective batch size was around 10k sequences and the training took around 4 days of wall clock time per model.

Subsequently, these trained models were used to generate improved backtranslations. We then continued training the first iteration of our models with the new backtranslated data for another 30,000 steps for the Icelandic-English direction, and 36,000 steps for the English-Icelandic direction. The same training configurations were maintained as for the earlier runs.

Dir.	Steps	'21 test	'21 dev	EEA
En-Is	40k	22.7	25.9	54.5
En-Is	40k + 36k	24.3	27.8	57.6
Is-En	36k	32.9	30.4	61.0
Is-En	36k + 30k	33.5	31.8	63.2

Table 4: BLEU scores for the mBART-25 adapted translation models over the newstest2021 and EEA evaluation sets.

The benefit of continuing training of the mBART-derived models ranges from 0.6 to 3.1 BLEU as shown in Table 4. BLEU performance is shown for both the newstest2021 development set as well as our cleaned-up dataset with sentence pairs from the EEA regulation corpus. Note that we do not fine-tune prior to evaluation nor do we perform checkpoint averaging.

5 Conclusion

We have shown how a small team with modest resources can adapt state-of-the-art methods to a medium resource language and achieve competitive results on machine translation between English and Icelandic.

The trained models are available for translation at `https://velthyding.is` and will be made available at the open `CLARIN-IS`[5] repository. While a formal human comparison of the current models to the popular Google Translate service has not been performed, hundreds of monthly active users choose our solutions for translation between Icelandic and English.

6 Future work

We note the relatively small training time of the mBART adaptation and the lack of Icelandic data in the pretraining task for mBART as primary factors that can be addressed for improving results. Additionally online (or semi-online) self-training instead of train-then-translate would also improve results, especially with selective loss truncation as described in (Zhou et al., 2021). The data selected for backtranslation should also be expanded for greater diversity of both genre and vocabulary. Finally, extending the translation context beyond the current sentence level is likely to improve results.

Acknowledgements

We would like to thank Prof. Dr.-Ing. Morris Riedel and his team for providing access to the supercomputer at Forschungszentrum Jülich where the mBART-25 translation models were trained.

This project was supported by the Language Technology Programme for Icelandic 2019–2023. The programme, which is managed and coordinated by Almannarómur, is funded by the Icelandic Ministry of Education, Science and Culture.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis,

[5] `https://repository.clarin.is`

Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Haukur Páll Jónsson, Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Steinþór Steingrímsson, and Hrafn Loftsson. 2020. Experimenting with Different Machine Translation Models in Medium-Resource Settings. In *Text, Speech, and Dialogue*, pages 95–103, Cham. Springer International Publishing.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

A. B. Nikulásdóttir, Jón Guðnason, Anton Karl Ingason, H. Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Steinþór Steingrímsson. 2020. Language Technology Programme for Icelandic 2019-2023. In *LREC*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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: Shared Task Papers*, pages 939–945, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Haukur Barri Símonarson, Vésteinn Snæbjarnarson, and Vilhjálmur Þorsteinsson. 2020. En-Is Synthetic Parallel Corpus. CLARIN-IS.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC 2018, Miyazaki, Japan.

Steinþór Steingrímsson, Hrafn Loftsson, and Andy Way. 2020. Effectively aligning and filtering parallel corpora under sparse data conditions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 182–190, Online. Association for Computational Linguistics.

Haukur Barri Símonarson and Vésteinn Snæbjarnarson. 2021. Icelandic Parallel Abstracts Corpus.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz 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 Track)*, 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, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Allegro.eu submission to WMT21 News Translation Task

Mikołaj Koszowski, Karol Grzegorczyk, Tsimur Hadeliya
ML Research Lab at Allegro.eu
{mikolaj.koszowski, karol.grzegorczyk, tsimur.hadeliya}@allegro.pl

Abstract

This paper describes Allegro.eu submission for the WMT21 news translation shared task. We focus on exploring data filtering and data augmenting methods. We submitted two single-directional models, one for English→Icelandic direction and other for Icelandic→English direction. Our news translation system is based on the transformer-big architecture, it makes use of corpora filtering, back-translation and forward translation applied to parallel and monolingual data alike.

1 Introduction

We participated in the WMT21 news translation shared task for English↔Icelandic language pair. It is a medium-resource regime with under 10M parallel sentences. In our experiments we focused on two approaches for improving translation system: data filtering methods inspired by work of (Jónsson et al., 2020) and data augmentation methods like back-translation or self-training (Edunov et al., 2018; Sennrich et al., 2016; He et al., 2019). We tried to use bi-directional translation models but single-directional proved to be better. We also tried to make use of pretraining on monolingual corpora, but it also was unsuccessful. Krubiński et al. (2020) showed in their ablation study that pretraining is the most successful for low-resource regimes under 1M parallel sentences.

2 Data

2.1 Data Preprocessing

We removed malformed utf-8 encodings, normalized text with NFKC Unicode normalization form, unescaped HTML, removed control characters and converted different whitespaces to a basic space character.

2.2 Data Filtering

We took part in a constrained track for the English↔Icelandic language pair for the news translation task. We used similar heuristic for filtering monolingual and parallel data. A proper sentence pair should fulfil these criteria:
For each sentence separately:

- length in chars $\in (10, 500)$
- length in words $\in (2, 100)$
- average word length in chars < 12
- max word length in chars < 28
- digit ratio < 0.15
- outside alphabet ratio < 0.015
- language detection probability > 0.9

Criteria calculated on a sentence pair:

- no digit sequence mismatch
- Levenshtein distance > 5
- Poisson based length logprob $>$ -10

For language identification we used the CLD2 library. We arrived at these threshold values by analyzing outliers of clean corpora: newsdev2021 development dataset and Jónsson's cleaned ParIce corpus (Jónsson et al., 2020). Our filtering procedure is inspired by Jónsson's and extracts 72% of the same sentences they extracted from the raw ParIce corpus (Barkarson and Steingrímsson, 2019). Each heuristic removes up to 5% of lines from those clean corpora, when all thresholds would be applied they would remove around 9% from the cleaned ParIce corpus. For all available raw parallel corpora this procedure would remove 35% of sentences. Table 1 shows sizes of raw and filtered corpora available in the constrained track.

2.3 Poisson based length filtering

This section describes an improved method of filtering sentences based on their lengths. A simple ratio of sentence lengths is a common method, but it is often too strict for short sentences and too loose

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 140–143
November 10–11, 2021.

Parallel corpora	raw	filtered	left
ParIce.1_1	3.56M	1.98M	0.56
ParaCrawl.7_1	2.39M	1.95M	0.81
WikiMatrix.1	313k	177k	0.57
wikititles.3	50k	2k	0.04
Total	6.31M	4.1M	0.65

Table 1: Sizes of parallel corpora.

for longer ones. We are using a simple assumption, that the distribution of lengths of expected translation is given by the Poisson distribution with a mean equal to a length of the source sentence. This type of length filtering is used by bicleaner framework (Sánchez-Cartagena et al., 2018). We use a correction factor $scl = 1.04$, which is a ratio of chars in the English side to the Icelandic side for the whole parallel corpus. We multiply source length by it or by its reciprocal before calculating probabilities, depending on the context. Figure 1 compares this method with a ratio-based heuristic where the allowable ratio range is $(0.5, 2)$. For this language pair the correction factor is close to 1.0, but for other language pairs it can deviate more, which can lead to bias when using a simple ratio-based heuristic.

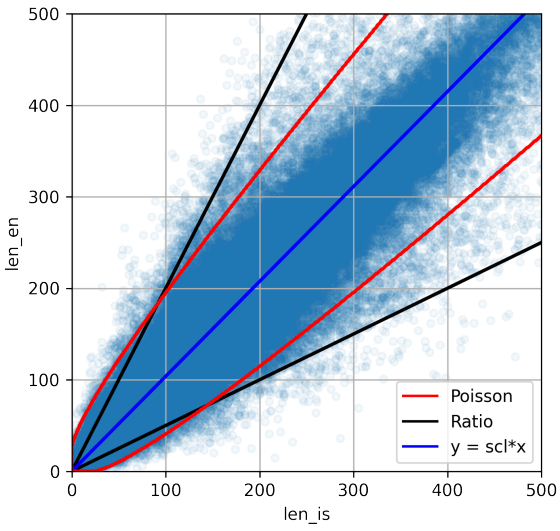


Figure 1: Distribution of lengths of parallel corpora. As depicted, Poisson-based heuristic allows more variation for shorter sentences and lower variation in length for longer ones.

2.4 Translation postprocessing

Our system has a tendency to generate the same quotation as in source text. Therefore, before submitting our translations for evaluation, we applied simple regular expressions to fix quoting. We made sure that only (" ") for English submission was used and for Icelandic we made sure that („ “) was used.

3 System overview

All of our models are based on the Transformer big architecture, as described in Vaswani et al. (2017). For training we used OpenNMT-py framework (Klein et al., 2017) together with sentencepiece tokenizer (Kudo and Richardson, 2018) unigram model of size 32k with full character coverage. We trained models on A100 GPU for 210k steps with a batch of 8192 tokens which amounts to around 12h per model. We used half-precision and tied embeddings. For optimization we used Adam (Kingma and Ba, 2014), with a linear warmup for learning rate for 15k steps up to 0.0005 and inverse square root decay afterwards. Additionally, all of our models were randomly initialized.

4 Results

Results are presented in Table 3. We trained a tokenizer on a cleaned ParIce corpus. A baseline model we trained on all available parallel corpora and achieved 18.1 BLEU in English→Icelandic direction and 24.0 BLEU in Icelandic→English direction.

4.1 Data filtering impact

We ran 4 variants with the same parameters as described at the beginning of section 3, but only for 100k steps. We compared the translation quality of models trained with filtered training corpus and the impact of cleaning data used in training tokenizer. We used the aforementioned cleaned ParIce corpus (Jónsson et al., 2020) to train the tokenizer. Table 2 presents the results of this comparison.

4.2 Back-translation of monolingual corpora

We took 10M monolingual sentences for each language and filtered them as described in section 2.2. For English we took only News Crawl from 2020, for Icelandic we used News Crawl 2020 and also Icelandic Gigaword to obtain full 10M sentences. We translated the English source to Icelandic, then translated it back to English. Then we compared those second translations to source by GLEU score

	clean tokenizer	raw tokenizer
clean corpus	16.6/22.6	14.0/19.4
raw corpus	16.2/22.2	14.2/18.9

Table 2: Comparison of impact of filtering data. Values reported are BLEU scores for en→is/is→en direction for newsdev2021. We can easily see that training tokenizer on clean data has a big impact. Also we can notice that removing 35% of parallel corpora can improve the quality of the model given the same amount of compute.

(Wu et al., 2016) and filtered the best 40% of pairs of original source and first translation based on that. GLEU score is a variation on the BLEU score. It is claimed to be a more accurate measure of single sentence translation quality. We repeated this procedure for 10M Icelandic monolingual sentences. It is interesting to note that 4.4% and 2.0% of second translations were the same as the original source, for English and Icelandic respectively. We then created English biased corpus which consisted of:

- 4M of clean parallel corpus
- 4M of English based back-translation where we used original source as target
- 4M of Icelandic base forward translation where we used our first translations as target

Then we used this corpus to train a new model, it achieved 26.8 BLEU in Icelandic→English direction.

4.3 Back-translation of parallel corpora

We used this newly acquired model to translate the Icelandic side of clean parallel corpus to English and likewise filtered by GLEU score for the English side of the corpus, finally we extracted 75% of most similar pairs. It is interesting to note that 11% of translations were the same as the English side of the parallel corpus. We then created a corpus for training English→Icelandic model, this time with typical setup for back-translation where original sentences were used as a target:

- 4M of clean parallel corpus
- 4M backtranslated monolingual corpus
- 3M backtranslated parallel corpus

Then we used this corpus to train a new model. It achieved 23.6 BLEU in English→Icelandic direction and that was our final model for this direction.

	newsdev2021	
Model	**En→Is**	**Is→En**
baseline	18.1	24.0
BT and FT mono	-	26.8
BT mono and parallel	23.6	-
BT mono and parallel	-	27.2
final models	23.6	27.4
	newstest2021	
final submission	22.7	33.3

Table 3: Comparison of forward-translation (FT) and back-translation (BT) model trained on monolingual and parallel corpora

Then, analogously, we used this model to translate the other side of the clean parallel corpora and filter by GLEU score. It is interesting to note that also 11% of translations was the same as the Icelandic side of the parallel corpus. We then created a corpus and trained Icelandic→English model which achieves 27.2 BLEU on the development set. For this direction our final system was an ensemble of this new model and previous best.

4.4 Denoising

As it has been recently demonstrated by Raffel et al. (2020), transfer learning can be successfully applied to sequence-to-sequences models. Therefore, we tried doing unsupervised de-noising pre-training based on provided monolingual data. We experimented with three different denoising schemes:

- Token-based masked language modeling (Devlin et al., 2019)
- Whole Word Masking objective inspired by BERT models released in May 2019
- BART-like denoising with text infilling and sentence permutation (Lewis et al., 2020)

We tried it in two regimes. One where we pretrain model and then finetune it on translation downstream task. The other where we train both denoising and translation objectives simultaneously. However, we didn't observe any benefits from doing this. The reason for this is unknown.

5 Conclusion

This paper describes Allegro.eu submission for the WMT21 news translation shared task. We took part in constrained track for the English↔Icelandic language pair only. Participation in this task allowed

us to deepen the understanding of filtering methods common in NMT. The experiments demonstrated the importance of data filtering in medium-resource regime machine translation. In this regime, less data but of higher quality can lead to superior results.

References

Starkaður Barkarson and Steinþór Steingrímsson. 2019. Compiling and filtering ParIce: An English-Icelandic parallel corpus. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 140–145, Turku, Finland. Linköping University Electronic Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *CoRR*, abs/1909.13788.

Haukur Páll Jónsson, Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Steinþór Steingrímsson, and Hrafn Loftsson. 2020. Experimenting with different machine translation models in medium-resource settings. In *Text, Speech, and Dialogue - 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8-11, 2020, Proceedings*, volume 12284 of *Lecture Notes in Computer Science*, pages 95–103. Springer.

Diederik P 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.

Mateusz Krubiński, Marcin Chochowski, Bartłomiej Boczek, Mikołaj Koszowski, Adam Dobrowolski, Marcin Szymański, and Paweł Przybysz. 2020. Samsung R&D institute Poland submission to WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 181–190, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. 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.

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*, volume 30. 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.

Illinois Japanese ↔ English News Translation for WMT 2021

Giang Le and **Shinka Mori** and **Lane Schwartz**
Department of Linguistics, University of Illinois Urbana-Champaign
gianghl2@illinois.edu, shinkam2@illinois.edu, lanes@illinois.edu

Abstract

This system paper describes an end-to-end NMT pipeline for the Japanese ↔ English news translation task as submitted to WMT 2021, where we explore the efficacy of techniques such as tokenizing with language-independent and language-dependent tokenizers, normalizing by orthographic conversion, creating a politeness-and-formality-aware model by implementing a tagger, back-translation, model ensembling, and n-best reranking. We use parallel corpora provided by WMT 2021 organizers for training, and development and test data from WMT 2020 for evaluation of different experiment models. The preprocessed corpora are trained with a Transformer neural network model. We found that combining various techniques described herein, such as language-independent BPE tokenization, incorporating politeness and formality tags, model ensembling, n-best reranking, and back-translation produced the best translation models relative to other experiment systems.

1 Introduction

Despite recent advances in machine translation made possible by neural networks with attention mechanism (Bahdanau et al., 2014; Luong et al., 2015), the Japanese-English pair remains a challenging language pair for machine translation systems to handle. Challenges posed by this language pair are multifaceted, starting from seemingly trivial differences in orthographic representations to deep structural divergence in syntax. This paper describes an end-to-end neural machine translation system and related experiments dedicated to the News Translation Shared Task where the target language pair is Japanese ↔ English, as part of a submission to the Sixth Conference on Machine Translation - WMT 2021. In our experiments, we explored the efficacy of techniques such as tokenizing with language-independent and language-dependent tokenizers, normalizing by orthographic conversion, creating a politeness-and-formality-aware model by implementing a tagger, back-translation, model ensembling, and n-best reranking. We found that normalizing the text by orthographic conversion did not improve over the baseline but controlling for politeness and formality levels of the text increased BLEU by 1.2 points for the En→Ja direction, and other techniques such as back-translation, model ensembling, n-best reranking also produced improvements.

The paper gives a detailed review of prior work, with a particular focus on WMT 2020 submissions, and then proceeds to describe our data, model architecture, experiments, results, and discussion of their implications.

2 Prior Work

In this section, techniques and development in neural machine translation will be reviewed with a focus on the techniques and implementation most recently used for the Japanese-English language pair. General techniques deployed across papers submitted to WMT 2020 are bitext data filtering, back-translation, fine tuning with in-domain data, knowledge distillation, rule-based reranking, transfer learning, co-reference processing, hyperparameter search, segmenting by subword units, BPE dropout, model ensembling, pre-training with monolingual data, experimenting with different word segmentation methods, context word embedding, domain adaptation, using related languages in joint training, domain tagging, reranking using backward and forward scores, and dual conditional cross-entropy filtering

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 144–153
November 10–11, 2021.

(Barrault et al., 2020). In subsequent subsections, representative methods and techniques will be described and the impacts of these methods presented, in so far as they are applicable to the Japanese-English pair.

2.1 Data Preprocessing

Data filtering, cleaning, and normalizing are essential steps in an NMT pipeline, due to the noisy nature of text corpora. A cursory glance at some of the given parallel corpora shows that our data could benefit from additional filtering and cleaning. For instance, the Paracrawl corpus contains a fair amount of duplicates or near duplicates and about 6 percent of the WikiMatrix corpus contains texts outside the source and target language.

Previous submissions to WMT 2020 utilized a mix of language-independent and language-dependent data preprocessing methods to prepare the corpora for training. Researchers also noted a few issues in the parallel corpora requiring special attention; for example, Kiyono et al. (2020) remarked that their translation output contains additional transliteration in brackets after names already transliterated into *katakana*, because these patterns are very common in the KFTT training corpus. They advised that this issue be handled during preprocessing, because postprocessing clean-up, while possible, tended to hurt brevity (Kiyono et al., 2020). Following this suggestion, we incorporated a preprocessing step (described in section 3) to handle these patterns.

2.2 Tokenization

Tokenization is an indispensable step in many natural language processing (NLP) applications. Byte-Pair-Encoding (BPE) by Sennrich et al. (2016c) is a popular compression algorithm that takes care of splitting words into subword units based on how frequent these units are. The main idea of BPE is to recover smaller subwords that are recurring in fuzzy 'word' boundaries in order to compress the vocabulary and decomposes rare words into known subwords. BPE is an effective solution to the issue of rare words, open vocabulary, and agglutinating morphology in some languages. The algorithm works by splitting all words into individual characters, adding them to a vocabulary, and then iteratively merging the most frequency subword pairs and adding them to the vocabulary.

Kudo and Richardson (2018) implemented BPE in SentencePiece, an unsupervised toolkit for word segmentation. A language-agnostic tokenizing and detokenizing algorithm that implements subword unit BPE (Sennrich et al., 2016c) and unigram language model (Kudo, 2018) to tokenize the data, SentencePiece also provides a convenient interface to quickly tokenize and detokenize the data, because its implementation of BPE treats the sentences as sequences of Unicode characters, does not rely on language-dependent logic, and allows training from raw texts. The developers of SentencePiece experimented their toolkit with and without pre-tokenization for an English-Japanese translation task, and found that the performance of training on raw texts is comparable to training with pre-tokenization.

Previous submissions to WMT 2020 are divided when it comes to which method was preferred for tokenization. Three submissions (Kiyono et al., 2020; Oravecz et al., 2020; Marie et al., 2020) used SentencePiece and three submissions (Kim et al., 2020; Shi et al., 2020; Zhang et al., 2020) used language-specific tokenizers to preprocess Japanese (MeCab) and English (Moses) corpora. MeCab is a popular lattice-based tokenizer for Japanese. It builds a graph-like data structure to hold possible tokens in the text and then uses the Viterbi algorithm to find the best path through the graph. Moses is a well-known statistical machine translation toolkit; its perl scripts are often used to preprocess English corpora for NMT training (Koehn et al., 2007). We experimented with both SentencePiece and language-dependent tokenizers prior to submission. The details will be outlined in section 5.1 of this report.

2.3 Model Architecture

Most of the papers submitted to WMT 2020 used the Transformer Big settings described in Vaswani et al. (2017) for their NMT model architecture (Marie et al., 2020; Kiyono et al., 2020; Shi et al., 2020; Oravecz et al., 2020; Zhang et al., 2020).

Prior to the publication of *Attention is All You Need*, prominent approaches to sequence-

to-sequence modeling include recurrent neural networks, long short-term memory (Hochreiter and Schmidhuber, 1997), and gated recurrent neural networks. All of these approaches suffer from computational bottleneck due to their sequential nature, which prevents parallelization within training examples. The Transformer did away with convolution and recurrence and focused on attention mechanisms, allowing for modeling of long-distance dependencies in parallel. Subsequently, it has been proven to be very successful at handling long distance dependency in natural language, as it allows the model to focus attention on particular source tokens via computation of an attention score. The attention score can be determined by way of different methods, such as a (scaled) dot product (implemented in Vaswani et al. (2017)), bilinear functions, or multi-layer perceptrons. The Transformer achieved state-of-the-art results in English ↔ French and English ↔ German translation tasks while cutting down on training time thanks to parallelization.

2.4 Back-Translation

Back-translation is a commonly used method in NMT to augment bitext training data by creating an additional synthetic parallel corpus from monolingual corpora (Sennrich et al., 2016b). To create back-translated data, a model that translates from target to source is required. First, a monolingual corpus of the target language is used to obtain translations in the source language. Subsequently, this monolingual corpus and the translated synthetic data are appended to the original training data to train the source to target model. It is ideal to have a lower ratio of synthetic data to parallel corpus in training the desired model. As the amount of bitext corpora available for the Japanese-English pair is well under 20 million sentence pairs, Japanese-English can be considered to be a medium-resource language pair and additional back-translated data could help improve translations. It should also be noted that there are limited domain-specific corpora for the language pair, and adding additional synthetic data back-translated from NewsCrawl and NewsCommentary may help augment the models.

2.5 Model Reranking

Zhang et al. (2020) implemented model reranking following Ng et al. (2019). N-best reranking scores and chooses a translation hypothesis from a list of n-best hypotheses. This method is based on a noisy channel model and Bayesian theorem of conditional probability in log scale, where the weight parameters are learned from fine tuning a validation set. For decoding, they used beam search to generate an n-best candidate list and chose the candidate hypothesis that maximizes the objective conditional probability as the best hypothesis.

Besides the noisy channel approach, reranking can be done using various criteria, such as distortion score, word penality, phrase penality, and so on. Shi et al. (2020) generated n-best candidates by model ensembling of forward translation models, backward translation models, and language models of the target language and then apply K-batched MIRA (Cherry and Foster, 2012) or noisy channel (Yee et al., 2019) to score them. Kiyono et al. (2020) generated n-best candidates from Source-to-Target L2R, R2L models, Target-to-Source L2R, R2L models, Unidictionary Language models, and Masked Language models to compute the scores for reranking.

We reranked translation hypotheses using perplexity as a criteria.

3 Data

Our system was trained, developed, and tested fully on data provided by the WMT 2021 organizers, making it a constrained submission. Details of the raw parallel corpora prior to substantial filtering[1] used in our baseline and experiment models can be viewed in Table 1.

We used the WMT 2020 development and test sets to compare various experiment models against the baseline: 1998 sentences in the development set in both directions, 1000 test sentences for the En→Ja direction, and 993 sentences for the Ja→En direction.

From the raw datasets, we applied data filtering to remove noisy data based on two main criteria, alignment confidence and language

[1]The original raw WikiMatrix corpus contains 3.8M sentences. We obtained 3.6M after eliminating sentence pairs that do not have the correct language codes in the corpus. That is the only filtering applied to the bitext corpora in Table 1

Corpus	Sentences (M)
JParacrawl 2.0	10.12
News Commentary v16	0.0019
Wiki Titles v3	0.757
WikiMatrix	3.6448
Subtitle Corpus	2.8013
KFTT	0.4438
Ted Talks	0.4462
Total	**18.215**

Table 1: Size of parallel corpora before filtering

Hyperparameters	T-Base	T-Big
Encoder layers	6	6
Decoder layers	6	6
Hidden layers	8	16
RRN	512	1024
d_{ff}	2048	4096
Dropout	0.1	0.3
Optimization	Adam	Adam
Decay	noam	noam
Learning rate	0.2	0.2
Warmup steps	8,000	8,000
Train steps	20,000	300,000

Table 2: Model Hyperparameters

identification. An alignment score is available for both JParacrawl and WikiMatrix corpora; we chose 0.6 and 1.0 as the threshold for alignment confidence in JParacrawl and WikiMatrix respectively. We used fasttext (Joulin et al., 2017) and its pre-trained language identification model to identify the language of our text sentence-by-sentence, and then we filtered sentence pairs where the language identification confidence score is less than 0.8. We also applied on-the-fly filtering of sentences longer than 100 tokens during training.

According to Kiyono et al. (2020), the KFTT corpus contained instances of having Japanese names followed by its English equivalent in parentheses, which caused their model to append English names after the Japanese name in the translation output, for example キャシディ・ステイ (Cassidy Stay). To avoid this, we filtered out English translations of names in Japanese source text, specifically WikiMatrix and KFTT, so that any English names in parentheses following its Japanese equivalent were removed. For English, we normalized punctuation and remove non-printing characters using the Moses scripts (Koehn et al., 2007). The amount of parallel training data after filtering was 12.7 M for training our submission models.

4 Model Architecture

We trained the parallel corpora using the Transformer base and Transformer big settings as described in Vaswani et al. (2017), presented in Table 2. Pre-submission experiments were trained under the Transformer Base setting while all submission models were trained under the Transformer Big setting. We used the same optimization settings in the Transformer big model as in the Transformer base model. We utilized the OpenNMT toolkit (Klein et al., 2017) with a Pytorch backend to train our models. Most submission models took about 7 days to train on one single NVIDIA GeForce GTX 1080 GPU under the Transformer Big setting.

5 Experiments

5.1 SentencePiece and Language-Dependent Tokenizers

We compared two methods of tokenization for our system. The first is a tokenization method based on BPE and SentencePiece, as described in 2.2. We used SentencePiece (Kudo and Richardson, 2018) to train SentencePiece models for Japanese and English with 32,000 as the vocabulary size. SentencePiece is used to create a tokenizer that depends on subword units, similar to Byte Pair Encoding (BPE). This method of tokenization is especially effective for languages such as Japanese which does not use whitespace to separate words, has agglutinating morphology, and contains many compound words. Using SentencePiece helps extract subwords within compound words and create a more robust tokenizer. The tokenizer model was used with OpenNMT, which performed tokenization on-the-fly. SentencePiece was used again to detokenize by removing the meta symbols from the output translation.

The second tokenization method that we experimented with is language-dependent. We tokenized English using Moses, following the steps described in Hieber et al. (2018), namely normalizing punctuation in the

raw data with normalize-punctuation.perl, removing non-printing characters with remove-non-printing-char.perl, and tokenizing by tokenizer.perl.

For Japanese, we tokenized the data with *fugashi* (McCann, 2020), a Python wrapper of the MeCab morphological analyzer described in 2.2. After tokenization, we applied BPE (Sennrich et al., 2016c) on both Japanese and English with 25,000 merge operations to constrain the vocabulary size.

For this comparison, we used a mid-sized corpus to save time and resources instead of the full 18M corpus. The number of sentences after filtering and preprocessing is 6.4M sentences. We trained the models using the Transformer Base settings, as described in Table 1.

5.2 Normalizing by Orthographic Conversion

The Japanese writing system uses a combination of three distinctive orthographic scripts: *kanji*, *hiragana*, and *katakana*. *Kanji* are Chinese characters, used to write content words such as nouns, verb stems, adjectives, and so on. *Hiragana* was derived from *kanji*. It is a phonetic syllabary, typically used to write conjugational endings, particles, and grammatical words. *Katakana*, also a phonetic syllabary much like *hiragana*, is typically reserved to write foreign words, loan words, or strengthen the emotive content of the texts. In modern times, the Latin alphabet also has increased visibility due to the popularity of English, and the Japanese language can be transliterated using this alphabet as well. This way of writing Japanese is called *romaji*.

We were interested in examining if converting the raw training texts to other orthographic scripts such as *hiragana* and *romaji* affects the translation quality of the output. Because *hiragana* and *katakana* have a one-to-one correspondence, it sufficed to experiment with either one of them. Converting the raw text to *hiragana* has a normalizing effect as what it does is reducing the logographic/ideographic *kanji* characters to their pronunciation, the moraic units written in the *hiragana* syllabraries. In that sense, it helps reduce variability in the data and perhaps is beneficial. However, normalizing also strips the text off many contextual cues that would be helpful in translation. The dispersion of *hiragana* in between the content words written in *kanji* is arguably systematic enough for our model to learn that one is used to represent grammatical particles and the other is used to represent objects, names, actions, and so on. Similarly, converting the raw text to *romaji* has a normalizing effect at the quasi-phonemic level. In a related manner, Du and Way (2017) looked at how a model trained on *pinyin* performed on a Chinese → English translation task. They found that using *pinyin* can help alleviate the problem of rare words, although it can introduce ambiguities.

To investigate the question of what impact normalizing the Japanese source text in *hiragana* and *romaji* does, we experimented training three Ja→En models where the source text is written in three orthographic scripts, the regular mixed style (baseline), the normalized moraic level *hiragana*, and the normalized quasi-phonemic level *romaji*. Each training corpus contained 4M sentence pairs, after being filtered by setting the language identification score threshold at 0.85 and sampled. The data were preprocessed with SentencePiece and trained under the Transformer Base setting, as described in Table 1.

5.3 Politeness and Formality Tagger

Previous work showed that controlling politeness levels has a positive impact on machine translation systems. Feely et al. (2019) implemented a formality-aware tagging method for En→Ja NMT. The authors classified formality levels into three categories (informal, polite, and formal) and found that using a heuristics-based tagger improved the system's performance. Similar to Feely et al. (2019), Sennrich et al. (2016a) and Yamagishi et al. (2016) improved on the stylistics of the output (politeness and honorific forms, respectively), by applying a side-constraint approach where target and source suffixes were added during training to add more meta-textual information to the corpora. We tested the effectiveness of this technique on an En→Ja translation system.

The news genre is frequently written in fairly formal Japanese. Makino (2008) described politeness and formality in Japanese as orthogonal concepts. It's possible to use polite

but informal language in daily polite conversations as well as formal language devoid of polite conjugations such as in news articles, academic papers, and so on. While the given parallel corpora are generally of the latter type, the subtitles corpus contains mostly colloquial language and the Ted talks corpus contains polite endings not intended to be used in news articles.

Due to the presence of mixed writing styles in the training data, we developed a politeness and formality tagger that works in conjunction with the Kytea tokenizer (Neubig, 2011) to address this issue, because we observed that our initial translation outputs often contained polite forms not commonly used in the news genre. Makino (2008) notes that verbs and i-adjectives have distinct forms for plain and polite but do not have distinct forms to indicate the formality levels because the same forms are used in both non-formal and formal writings. Furthermore, the copula *da* conjugation is critical to indicate formality. The tagging schema developed in Feely et al. (2019) combines the formal, plain form *dearu* into the polite category, and the formal category is what is typically referred to as *keigo* (honorifics). Our tagging schema is tailored towards the news corpus where *dearu* features often as a marker of formal writing while polite endings and *keigo* do not typically surface (see Appendix A for the detailed schema). Our tagger extracts the verb endings from the annotated sentences returned by Kytea and appends a <polite> or <formal> tag to the beginning of the source (English) side. Plain forms are left untagged as they are the default forms in the news genre.

Applying this tagger on a 12.7M training corpus results in 34.76% tagged as polite and 3.81% tagged as formal. We tokenized the data using SentencePiece transforms, implemented in the OpenNMT toolkit. We also filtered out sentence pairs longer than 100 tokens. We trained the models using the Transformer Big settings, as described in Table 1.

5.4 Back-Translation

For back-translation, we preprocessed a subset of 4M sentences from the monolingual Newscrawl corpus in the same manner described in 3. The filtered corpus was 3,344,628 lines each. We then used the previously trained Ja→En and En→Ja model to translate the monolingual data to create synthetic data, setting a beam size of 1 during decoding. We obtained 2.4M and 2.6M sentences of Japanese and English synthetic data from back-translation, respectively. This was combined with the existing parallel data to create a corpus of approximately 15M sentences.

5.5 Model Ensembling and N-Best Reranking

For n-best reranking, we used a script by Xu Song, `bert-as-a-language-model`[2], which calculates the probability of tokens and perplexity of sentences given a corpus. Using OpenNMT's option to produce n-best translations from an emsemble of several high-performing checkpoints, we created 10 best translations, and used `bert-as-a-language-model` to pick the hypothesis with the best perplexity score. This method ensures the selected hypothesis has maximized fluency compared to other candidates.

6 Results and Discussion [3]

6.1 SentencePiece and Language-Dependent Tokenizers

We obtained the BLEU scores in Table 3 for our models. The comparison is not entirely fair because the amount of data trained for the Moses and *fugashi* tokenizer to translate in the Ja→En direction is 7.3M instead of 6.4M like other models. Additionally, the number of BPE merge operations learned for the language-dependent tokenizer case should have been set to the same as that of SentencePiece for a more equitable comparison.

Using SentencePiece appears to yield better BLEU result in this experiment; however, we also did not keep the other factors constant across the different models under comparison.

[2]https://github.com/xu-song/bert-as-language-model

[3]Please note that the baseline models for experiments vary, as some experiments related to data preprocessing such as tokenization method and normalizing by orthographic conversion were conducted very early on in our project. These models were also trained under the Transformer Base setting, unlike later models trained under the Transformer Big setting. It follows that the baseline results vary from experiment to experiment, except for the tagger and back-translation experiments, where the same En→Ja baseline was used.

Tokenizer Models Ja→En	BLEU
SentencePiece	14.0
Moses and *fugashi*	9.9
Tokenizer Models En→Ja	**BLEU**
SentencePiece	16.0
Moses and *fugashi*	9.9

Table 3: Tokenizer Comparison

Orthographic Scripts Ja→En	BLEU
Mixed scripts (baseline)	14.2
Hiragana	12.6
Romaji	12.8

Table 4: Orthographic Script Comparison

Models En→Ja	BLEU	chrF
Baseline	18.6 ±0.8	28.4 ±0.7
With tagger	19.8 ±0.8	29.5 ±0.7

Table 5: Politeness-and-Formality-Aware Model vs. Baseline

Models En→Ja	BLEU	chrF
Baseline	18.6 ±0.8	28.4 ±0.7
With BT data	18.8 ±0.8	28.8 ±0.7
Models Ja→En	**BLEU**	**chrF**
Baseline	17.0 ±0.8	44.7 ±0.8
With BT data	18.7 ±0.8	46.6 ±0.8

Table 6: Back-Translation vs. Baseline

Nonetheless, this experiment's result led us to adopt SentencePiece as our preferred method for segmentation in other experiments.

6.2 Normalizing by Orthographic Conversion

We obtained the BLEU scores in Table 4 for our models. It can be seen from the results that training with normalized data by orthographic conversion does not improve the models over the baseline. The models trained on normalized data also have similar performances.

The result of this experiment suggests that normalizing by orthographic conversion might have removed too many contextual cues for the model to perform well. Possible work for future experiments include investigating whether normalizing *katakana* in mixed-script text into *hiragana* could have a positive impact, because doing so would remove variability but would not introduce ambiguity to the extent it might have done when the content words in *kanji* were also normalized. Another direction for future research involves looking at training NMT models using sub-character units such as radicals or strokes, as was done in Zhang and Komachi (2018).

6.3 Politeness and Formality Tagger

BLEU and chrF scores with a 95% confidence interval from a baseline model and a tagger model as seen in Table 5 shows that using a formality and politeness aware model improves the model's performance.

The result of this experiment is very encouraging to us as the score increase is notable. It also suggests that the proposed classification of predicate endings works well for the news training data available. The training data used for this experiment contains 12.7M sentence pairs. Developing a politeness and formality aware model applicable to a wider selection of genres in Japanese remains future work, where careful consideration of different writing styles and additional classification of stylistic markers are needed.

6.4 Back-Translation

Using back-translated data improved the results (reported with a 95% confidence interval), although the gain in the En→Ja direction was modest, as shown in table 6. The results reinforce previous findings that back-translation generally improves translation quality, and for languages with low resources, it can be especially useful. Although the Ja-En pair is not considered low-resourced, the parallel data for news-specific corpus was very scarce, so using the monolingual newscrawl and newscorpus was beneficial to the model learning.

6.5 Model Ensembling and N-Best Reranking

During the decoding phase, we ensembled the highest performing checkpoints and obtained 10 best translations from those checkpoints. The best hypothesis was determined by the best perplexity score of the language model.

Models En→Ja	BLEU
Baseline	32.9
N-best reranking	34.0
Models Ja→En	**BLEU**
Baseline	17.6
N-best reranking	18.6

Table 7: N-best reranking vs. Baseline

We found that for both directions, this method resulted in improved translations, as demonstrated in table 7. This evaluation result was done on the WMT 2021 test set and was obtained during the submission period using the submitted models.

7 Conclusion

We produced several models to tackle the task of translating Japanese to English and English to Japanese. Namely, we have used BPE, employed a politeness and formality tagger, and during decoding, utilized model ensembling and n-best reranking. Normalizing by orthographic conversion did not produce improvement compared to the baseline, but the other techniques have all proven to be effective and thus have been employed in our final submissions. We also found that for both En→Ja and Ja→En, adding back-translated data improved the results. This may be explained by the fact that there is very little parallel data in the news domain, and adding synthetic data from alternative in-domain sources helped tune the model. While improvement in the BLEU score is modest for En→Ja, we expect the results to improve further if we increase the amount of back-translated data. We also showed that employing a tagger to introduce more contextual cues related to politeness and formality to our translation system is an effective technique. Differences in formality and politeness levels present are issues often encountered when using training data in languages with rich honorifics. Thus the technique employed in this paper could be extended to other languages such as Korean.

Acknowledgement

We would like to thank the reviewers for their assessment of this paper as well as their valuable feedback, which have helped us tremendously in the revision process.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. 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.

Jinhua Du and A. Way. 2017. Pinyin as subword unit for Chinese-sourced neural machine translation. In *AICS*.

Weston Feely, Eva Hasler, and Adrià de Gispert. 2019. Controlling Japanese honorifics in English-to-Japanese neural machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 45–53, Hong Kong, China. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. Sockeye: A toolkit for neural machine translation.

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

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*, pages 427–431. Association for Computational Linguistics.

Jiwan Kim, Soyoon Park, Sangha Kim, and Yoonjung Choi. 2020. An iterative knowledge transfer NMT system for WMT20 news translation

task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 139–144, Online. Association for Computational Linguistics.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-AIP-NTT at WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 145–155, Online. 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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. 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.

Seiichi Makino. 2008. *A dictionary of advanced Japanese grammar = Nihongo bunpo jiten. Jokyu hen*, first edition. edition. The Japan Times, Tokyo.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Combination of neural machine translation systems at WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 230–238, Online. Association for Computational Linguistics.

Paul McCann. 2020. fugashi, a tool for tokenizing japanese in python.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, László Tihanyi, David Kolovratnik, Bhavani Bhaskar, Adrien Lardilleux, Szymon Klocek, and Andreas Eisele. 2020. eTranslation's submissions to the WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 254–261, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. pages 35–40.

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.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao. 2020. OPPO's machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 282–292, Online. 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.

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, Osaka, Japan. The COLING 2016 Organizing Committee.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Longtu Zhang and Mamoru Komachi. 2018. Neural machine translation of logographic language using sub-character level information. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 17–25, Brussels, Belgium. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345, Online. Association for Computational Linguistics.

A Appendix

<polite>
です *desu*,
ます *masu*,
でした *deshita*,
ました *mashita*,
まして *mashite*,
ません *masen*,
ましょう *mashou*,
なさい *nasai*,
ください *kudasai*,
くださいませ *kudasaimase*
<formal>
である *dearu*,
であろう *dearou*,
であるだろう *dearudarou*,
であった *deatta*,
であったろう *deattarou*,
であっただろう *deattadarou*,
であっている *deatteiru*,
であっていた *deatteita*,
であれる *deareru*,
であらせる *dearaseru*,
であられる *deararareru*,
であらない *dearanai*,
であらないだろう *dearanaidarou*,
であらなかった *dearanakatta*,
であらなかっただろう *dearanakattadarou*,
であれない *dearenai*,
であらせない *dearasenai*,
であられない *deararenai*

Table 8: Tagging Rules

MiSS@WMT21: Contrastive Learning-reinforced Domain Adaptation in Neural Machine Translation

Zuchao Li[1,2,3], **Masao Utiyama**[4,*], **Eiichiro Sumita**[4], **and Hai Zhao**[1,2,3*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[4]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
charlee@sjtu.edu.cn, {mutiyama, eiichiro.sumita}@nict.go.jp, zhaohai@cs.sjtu.edu.cn

Abstract

In this paper, we describe our MiSS system that participated in the WMT21 news translation task. We mainly participated in the evaluation of the three translation directions of English-Chinese and Japanese-English translation tasks. In the systems submitted, we primarily considered wider networks, deeper networks, relative positional encoding, and dynamic convolutional networks in terms of model structure, while in terms of training, we investigated contrastive learning-reinforced domain adaptation, self-supervised training, and optimization objective switching training methods. According to the final evaluation results, a deeper, wider, and stronger network can improve translation performance in general, yet our data domain adaption method can improve performance even more. In addition, we found that switching to the use of our proposed objective during the finetune phase using relatively small domain-related data can effectively improve the stability of the model's convergence and achieve better optimal performance.

1 Introduction

News translation (Bojar et al., 2017, 2018; Barrault et al., 2019, 2020) is one of the most prominent and appealing tasks in machine translation evaluation (Wu et al., 2020b; Li et al., 2020c). Our MiSS system took part in the WMT21 news translation task, including English → Chinese (En → Zh), Chinese → English (Zh → En), and Japanese → English (Ja → En) translation directions. We developed translation systems for this year's submission to investigate machine translation techniques from two perspectives: model structure and model training. All of the data used by the submitted systems is constrained. Due to a lack of training resources, the English->Japanese translation direction is only investigated from the model structure perspective.

From the perspective of model structure, we choose the Transformer (Vaswani et al., 2017; Li et al., 2021c) model based on self-attention, which is extensively utilized in neural machine translation systems, as our basis (Zhang et al., 2020b; Li et al., 2020d). On this strong foundation, we opt to simply deepen the model by increasing the number of encoder layers or widen the model by increasing the hidden size of the model to obtain a deeper or wider model. When deepening or widening the model, we found that there is no need for additional sophisticated structure design (e.g., layer drop (Fan et al., 2020) / sublayer drop (Li et al., 2021a)) or training strategy when there is adequate training data available. In addition to Transformer architecture, Wu et al. (2019) propose a dynamic convolution structure that can perform competitively or better to the self-attention structure. Follow the practice in WMT20 (Wu et al., 2020a), we also applied the dynamic convolution architecture as another basis.

According to our preliminary results on the development set, domain has a significant impact on performance, despite the fact that we are working with the resource-rich En-Zh and En-Ja language pairs. This year's submissions are mostly concerned with utilizing training approaches to mitigate the impact of domain differences. Specifically, we first use data in all hybrid domains to train the initial NMT model, and then, based on sentence embedding model enhanced by contrastive learning, the parallel/monolingual corpus is filtered monolingually or cross-lingually, and the filtered domain-related parallel corpus is used for further finetuning, and the domain-related monolingual corpus is used for in-domain back-translation enhancement. In addition, we also adopted a self-supervised training method to train the model on the given source text of the test set and its domain-related monolingual text obtained by filtering. In self-supervised

*Corresponding author. Zuchao Li was limited technical researcher at NICT when this work was done. This work was partially supported by the Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 154–161
November 10–11, 2021.

training, we combine our *Data-dependent Gaussian Prior Objective* (D2GPo) objective (Li et al., 2020b) to alleviate the collapse due to non-golden targets. In the finetune stage with the domain-related parallel corpus, we adopted the training strategy of switching the optimization objective from the MLE to our proposed *Dual Skew Divergence* (DSD) (Li et al., 2019). The results demonstrated that switching to the DSD objective resulted in improved convergence.

From the evaluation results, we observe substantial improvements over the strong baseline with 4.3 (En → Zh), 4.8 (Zh → En), 3.2 (Ja → En) BLEU scores on the development sets, respectively. The gains can be attributed to larger model capacity and better training strategies. And the results suggest that the cost of domain adaptation to improve performance is less than the cost of increasing model capacity.

2 Model Perspective

With the development of deep learning in NLP (He et al., 2018; Cai et al., 2018; He et al., 2019; Li et al., 2021d), model ensembling can usually produce better results than single models, and the bigger the difference between the models used for ensembling, within a certain limit, the higher the improvement will be. As a result, we chose four distinct typical architectures as the basis for single NMT models and trained them on the same data. The detailed parameters of each model architecture are shown in Table 1.

Deep Transformer Some related works (Zhang et al., 2019; Wang et al., 2019; Li et al., 2020a, 2021a) have revealed that deep networks have great advantages in NMT performance compared to shallow networks recently. Based on the Transformer NMT model architecture, we found that in the presence of sufficient training data, merely increasing the number of stacked layers of the encoder can fulfill the goal of deep Transformer without the use of additional initialization, dropout, or layer skipping techniques.

Wide Transformer Recent researches (Sun et al., 2019; Wu et al., 2020a; Zhang et al., 2020a; Wu et al., 2020b; Meng et al., 2020) have demonstrated that, in addition to deepening the NMT model, widening the model can also effectively improve translation performance, with increasing the feed-forward network (FFN) size in the Transformer model bringing less training and inference cost than increasing the overall hidden size of the model. We took a same practice in our work by increasing the FFN size and established a Wide Transformer baseline.

	Deep Transformer	**Wide Transformer**	**Deep DynamicConv**
Enc. Layers	40	20	20
Dec. Layers	6	6	6
Attn. Heads	16	16	16
Hidden Size	1,024	1,024	1,024
FFN Size	4,096	8,192	4,096

Table 1: Hyper-parameters of different model architectures. Note that Wide Transformer with relative position encoding was also used as baseline models.

Deep DynamicConv Dynamic convolution (DynamicConv) (Wu et al., 2019) was proposed as a replacement for Transformer architecture and has piqued much interest (Wu et al., 2020a) due to its good speed advantage and comparable performance. To enhance the performance of single model, we also deepen the DynamicConv model by increasing the number of encoder layers, denoted as Deep DynamicConv. The original DynamicConv model consists of 7 encoder layers and 6 decoder layers. We deepen the DynamicConv model's encoder layers to Deep DynamicConv. Because the kernel size of each convolution layer in the DynamicConv model differs, we set the kernel sizes of the 16 encoder layers in Deep DynamicConv to `[3, 7, 15, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31]` and leave the other settings unchanged from the original model.

Relative Position Encoding Because self-attention in the convention Transformer model is position-independent, the encoded features must be enhanced with explicit positional information for natural language processing. Absolute position encoding is usually employed in the Transformer NMT model. Shaw et al. (2018) proposed to add relative position encoding (RPE) for improving self-attentional features and shown additional performance gains. We also applied relative position encoding to the Wide Transformer model and created another strong baseline.

We use the identical vocabulary and data to train these four baseline models separately, and then average the best 5 checkpoints in each model's training phase to generate the final model output

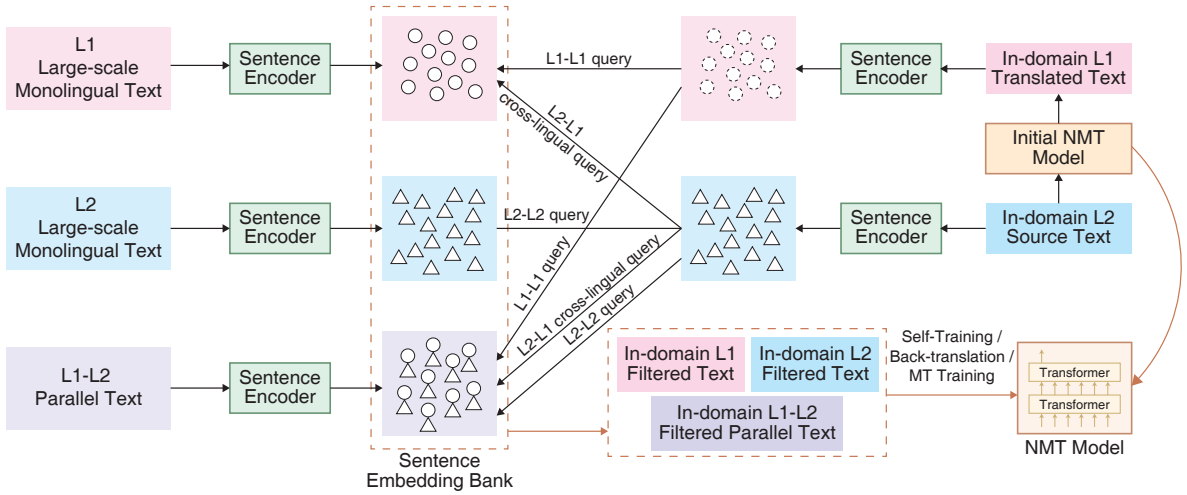


Figure 1: Illustration for contrastive learning-reinforced domain adaptation

in the corresponding stage. According to Wu et al. (2020a)'s experience, the best 5 checkpoints are determined based on the BLEU metric on the development set rather than the perplexity (PPL) metric. Furthermore, we applied the D2GPo objective (Li et al., 2020b) in the training process to obtain more stable convergence and decrease the impacts of overfitting resulting from the training set's noise.

3 Training Perspective

Contrastive Learning-reinforced Domain Adaptation Data domain issues have been found to have a significant impact on machine translation performance (Saunders, 2021). The official training data is of hybrid domain, despite the fact that the evaluation task is news translation. And, while news translation corpora can be deemed to be in the news domain, there are significant variances in news styles within the same domain. As a result, one of the keys to performance enhancement will be how to utilize the data training model that is closer to the evaluation data domain and style.

Using languages L_1 and L_2 as an example, the data that may be used comprises the parallel corpus $D^P_{L_1-L_2}$, as well as their respective large-scale monolingual corpus $D^M_{L_1}$ and $D^M_{L_2}$. Parallel corpora are typically utilized for direct training of NMT models, whereas monolingual corpora are used for back-translation (Edunov et al., 2018) and self-supervised training (Jiao et al., 2021). The domain filtering method can be utilized in these three training procedures to create corpus whose domain is more similar to the development and test sets.

Instead of relying on the co-occurrence probability of the surface tokens in the sentence, we based the domain filtering on the hypothesis that the more similar the sentence representations generated by the Transformer encoder are, the more likely they are to be dispersed in the same domain. Because the current Transformer encoder's representation is based on the bidirectional and full attention of all tokens, the combination and order of tokens have a significant impact on the final representation, the sentence representation is adequate for capturing domain information. As a result, we use the sentence embedding distance to measure the domain similarity.

We leveraged a universal paraphrastic sentence encoder (Wieting et al., 2016; Ethayarajh, 2018; Li and Zhao, 2020) to embed each given sentence to a dense representation. On a large scale monolingual corpus, we train our own monolingual and multilingual sentence encoder, a Transformer that has been pre-trained using masked language modeling (Devlin et al., 2019; Zhang et al., 2020c; Li et al., 2021b), with the XLM toolkit (Conneau et al., 2020) and fine-tuned to maximize cosine similarity between similar sentences. Contrastive learning seeks to acquire effective representation by pulling semantically close neighbors and pushing non-neighbors apart (Hadsell et al., 2006). Since this criterion precisely meets the requirements of sentence representation learning, we use contrastive learning to finetune the pre-trained sentence encoder. Figure 1 illustrates our contrastive learning-reinforced domain adaptation method.

According to the domain adaptation requirements in actual machine translation, the trained sentence encoder needs respond to four scenar-

ios: *Original Input Monolingual Filter*, *Translated Input Monolingual Filter*, *Original Input Cross-lingual Filter*, *Translated Input Cross-lingual Filter*. Because the fourth scenario can be covered by the first, we only employ the first three scenarios in our experiment.

For all scenarios, we first follow Gao et al. (2021)'s approach to perform unsupervised training in which the input sentence itself is used as a positive instance due to there will be some differences between the sentence representations of the two pass input with the presence of the model dropout, and other sentences in the in-batch are used as negative instances.

The unsupervised contrastive learning-trained monolingual sentence encoder can be used directly as an evaluator of the similarity of sentences in the same language and to mine similar sentences from the sentence bank. However, for the non-gold translated sentences filtering, we apply the baseline NMT models to translate parallel corpus and to back-translated monolingual corpus to generate pseudo-paraphrase corpus. And then triplet loss is used to fine-tune the unsupervised sentence encoder:

$$\mathcal{L}(x, y) = max(0, \alpha - cos(x, y)) + cos(x, y_n),$$

where positive pairs (x, y) are paraphrases from translation or back-translation, y_n are in-batch negative instances.

Likewise, we still need cross-language filtering, therefore we use parallel corpus instead of synthetic pseudo-restatement corpus and triplet loss for additional finetuning on the multilingual sentence encoder.

As shown in Figure 1, taking the L_2 in-domain source sentences in development set as an example, we first use the initial NMT model to translate these sentences to L_1 translated text. The different trained sentence encoder is then used to encode these sentences and the large-scale monolingual or parallel corpus based on different scenarios respectively. Then, using the faiss toolkit[1], a query procedure is performed to locate related in-domain monolingual or parallel corpora with similarity calculation and ranking.

Back-translation and Self-supervised Training Using the in-domain monolingual and parallel corpus, we may train the initial model using back-translation and self-supervised training approaches. For back-translation, we leverage the original multiple NMT models to translate these monolinguals into various pseudo-parallel corpora, and then combine them with the in-domain parallel corpus to finetune the NMT model. For self-supervised training, we use a variety of models to perform ensemble translation on the in-domain monolingual text as the translation target and combine the in-domain translation corpus to fine-tune the model. In the specific implementation, we perform back-translation and self-supervised training consecutively such that the self-supervised training stage can exploit the stronger NMT model trained during the back-translation stage.

Optimization Objective Switching Training It is easier to fall into a local optimum in the process of back-translation and self-supervised training because there are relatively fewer in-domain data and input or output in part of the data utilized is not gold. According to our experience in (Li et al., 2019), switching the training objective to the adversarial learning objective after MLE training converges might help jump out of the local optimal state and get better performance. Follow this practice, in the back-translation and self-supervised training stages, we first employ MLE target training to converge on a development set and then switch to Li et al. (2019)'s DSD loss for further training:

$$\begin{aligned}\mathcal{L}_{DSD} = -\frac{1}{n}\sum_{i=1}^{n}[&\beta(t)\mathbf{y_i}\log((1-\alpha)\hat{\mathbf{y}}_\mathbf{i} + \alpha\mathbf{y_i}) \\ &-(1-\beta(t))\hat{\mathbf{y}}_\mathbf{i}\log(\hat{\mathbf{y}}_\mathbf{i}) \\ &+(1-\beta(t))\hat{\mathbf{y}}_\mathbf{i}\log((1-\alpha)\mathbf{y_i} + \alpha\hat{\mathbf{y}}_\mathbf{i})],\end{aligned}$$

where $\mathbf{y_i}$ is the i-th token in the target sequence $\mathbf{y}$, $\hat{\mathbf{y}}_\mathbf{i}$ is the i-th predicted token, α is a hyper-parameter in α-skew divergence (Lee, 1999), and $\beta(t)$ is the controllable weight from the PID controller.

4 Data Setup

English↔Chinese In the English↔Chinese translation, we used all official parallel corpus, including ParaCrawl v7.1, News Commentary v16, Wiki Titles v3, UN Parallel Corpus V1.0, CCMT Corpus and WikiMatrix. For English, we use the tokenization tool provided by *Moses*[2], and

[1] https://github.com/facebookresearch/faiss

[2] https://github.com/moses-smt/mosesdecoder

	Systems	En→Zh		Zh→En		En→Ja		Ja→En	
		Dev	*Test*	*Dev*	*Test*	*Dev*	*Test*	*Dev*	*Test*
	Transformer-big	31.67	–	33.26	–	23.31	–	21.61	–
	Deep Transformer	32.48	–	34.18	–	24.68	–	22.78	–
①	++ID-BT	35.30	–	38.94	–	–	–	24.46	–
②	++ID-ST	35.95	–	39.18	–	–	–	**25.82**	–
	Wide Transformer	32.67	–	34.01	–	24.27	–	23.20	–
③	++ID-BT	35.37	–	38.82	–	–	–	24.55	–
④	++ID-ST	**36.15**	–	39.13	–	–	–	25.71	–
	Deep DynamicConv.	32.39	–	33.68	–	24.08	–	21.91	–
⑤	++ID-BT	35.01	–	38.66	–	–	–	24.37	–
⑥	++ID-ST	36.03	–	39.05	–	–	–	25.66	–
	Wide Transformer w/ RPE	32.52	–	34.35	–	**24.76**	–	22.78	–
⑦	++ID-BT	35.55	–	38.91	–	–	–	24.48	–
⑧	++ID-ST	36.08	–	**39.20**	–	–	–	25.71	–
	Baseline Ensemble	32.79	31.9	34.47	27.8	24.79	42.6	23.15	23.8
	Ensemble: ① + ③ + ⑤ + ⑦	35.62	35.7	38.98	32.4	–	–	24.63	26.4
	Ensemble: ② + ④ + ⑥ + ⑧	**36.41**	**36.2**	**39.25**	**32.6**	–	–	**25.99**	**27.0**

Table 2: BLEU evaluation results on the WMT 2021 development and test sets. The BLEU in the development set is a word-level MultiBLEU score, but the BLEU in the test set is from the official evaluation. Due to a lack of resources, En→Ja only completed the baseline training and ensemble submission.

for Chinese, we use *pkuseg* (Luo et al., 2019) as the word segmentor. We adopt a joint byte pair encoding (BPE) (Sennrich et al., 2016) with 44K operations for subword vocabulary in English and Chinese. Punctuation normalization is not employed to preprocess the training data in order to prevent complex post-processing of punctuation restoration. For English post-processing, we use the script in *Moses* to de-tokenize the translation, whereas for Chinese, we employ sacremoses[3] for de-segmentation.

English↔Japanese In the English↔Japanese translation, data for training were combined from ParaCrawl v7.1, News Commentary v16, Wiki Titles v3, WikiMatrix, The Kyoto Free Translation Task Corpus, and TED Talks. Similarly, the Japanese sentences are segmented using the Mecab[4] segmentor, while the English sentences are processed using the Moses tokenizer. The size of the English and Japanese joint BPE is also set to 44K. In post-processing, Moses script and sacremoses are also employed for detokenization.

We merged the whole news-crawl corpus for monolingual data. However, in Chinese and Japanese, news-crawl corpus alone is insufficient to train the sentence encoder, so we sampled some data from the common-crawl corpus and eventually produced the data in English, Chinese, and Japanese 100M sentences each. For pre-processing, we exclude sentences that are more than 175 words long, and the word ratio between the source and the target greater than 1:2 or 2:1.

5 Model Training

All of our NMT models are built using the Fairseq toolkit. Except for the switching training phase, all models are optimized with Adam optimizer, and SGD optimizer is utilized for optimization training when switching to DSD loss. During the baseline model training process, the learning rate is scheduled using the inverse sqrt scheduler with 4000 warm-up steps, maximum learning rate 5e-4, and betas (0.9, 0.98). Each model is trained on 8 NVIDIA V100 GPUs, with batch size limited to 8192 tokens per GPU. FP16 is emploted to save GPU memory and speed up calculations. To increase the virtual batch size, we set the gradient update steps to 8 during the training phase. The label smoothing and dropout values are both set to 0.1. In the finetuning stage, we utilize a smaller batch size, 4,096 tokens per GPU, and train the model at a fixed learning rate of 1e-4. Sentence encoder models are developed with the XLM toolkit, and the architecture is based on the BERT-base. The hidden size, heads, hidden layers, and FFN size are 768/12/12/3072 respectively. During training, a early stop mechanism is applied in which the training will stop when the PPL on the development set does not decrease after 25 epochs.

[3] https://github.com/alvations/sacremoses

[4] https://github.com/taku910/mecab

6 Results and Analysis

Table 2 shows the results on the development sets as well as the official evaluation results on the WMT21 test sets. First, when comparing Deep Transformer, Wide Transformer, and Transformer-big, we observed that increasing the number of model layers or widening the model to increase the number of model parameters can result in large performance benefits. Second, Deep DynamicConv has shown comparable results to Deep Transformer in multiple data sets, demonstrating that DynamicConv is a viable replacement option for Transformer. Third, the Deep Transformer w/ RPE model outperforms Deep Transformer model in most circumstances, demonstrating that machine translation benefits from additional relative position encoding information. Fourth, in-domain back-translation (ID-BT) and in-domain self-supervised training (ID-ST) improve the model's performance substantially more than the increased model parameters, indicating that the data domain is a primary factor limiting translation performance. Furthermore, these enhancements demonstrate that our domain adaption approach of contrast learning-reinforced is a effective approach. Finally, we performed ensemble on the four finetuned baselines and received even higher results, demonstrating that the models of the four architectures differ from each other.

7 Conclusion

In this paper, we introduce our MISS translation system, which participated in the WMT21 news translation task. We developed a new contrast learning-reinforced domain adaptation strategy in this work, and the experimental findings suggest that this method may significantly increase translation performance. Furthermore, we conducted experiments on a range of model architectures. Our domain adaption strategy improved these strong baseline models significantly, illustrating the method's generality and indicating that the performance deficiency is not due to a specific model structure.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. 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*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*,

pages 4171–4186, Minneapolis, Minnesota. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

Shexia He, Zuchao Li, and Hai Zhao. 2019. Syntax-aware multilingual semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5350–5359, Hong Kong, China. Association for Computational Linguistics.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2061–2071, Melbourne, Australia. Association for Computational Linguistics.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2840–2850, Online. Association for Computational Linguistics.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, College Park, Maryland, USA.

Bei Li, Ziyang Wang, Hui Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2021a. Learning light-weight translation models from deep transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13217–13225. AAAI Press.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020a. Shallow-to-deep training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 995–1005, Online. Association for Computational Linguistics.

Yian Li and Hai Zhao. 2020. Learning universal representations from word to sentence. *arXiv preprint arXiv:2009.04656*.

Zuchao Li, Kevin Parnow, Hai Zhao, Zhuosheng Zhang, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2021b. Cross-lingual transferring of pre-trained contextualized language models. *CoRR*, abs/2107.12627.

Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020b. Data-dependent gaussian prior objective for language generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zuchao Li, Zhuosheng Zhang, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2021c. Text compression-aided transformer encoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zuchao Li, Hai Zhao, Shexia He, and Jiaxun Cai. 2021d. Syntax Role for Neural Semantic Role Labeling. *Computational Linguistics*, pages 1–46.

Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2020c. SJTU-NICT's supervised and unsupervised neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 218–229, Online. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020d. Reference language based unsupervised neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4151–4162, Online. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, Yingting Wu, Fengshun Xiao, and Shu Jiang. 2019. Controllable dual skew divergence loss for neural machine translation. *arXiv preprint arXiv:1908.08399*.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain Chinese word segmentation. *CoRR*, abs/1906.11455.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 239–247, Online. Association for Computational Linguistics.

Danielle Saunders. 2021. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *arXiv preprint arXiv:2104.06951*.

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.

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)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. 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. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020a. The volctrans machine translation system for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 305–312, Online. Association for Computational Linguistics.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020b. Tencent neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 313–319, Online. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020a. The NiuTrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Zuchao Li, and Hai Zhao. 2020b. Neural machine translation with universal visual representation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020c. Semantics-aware BERT for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press.

The Fujitsu DMATH submissions for WMT21 News Translation and Biomedical Translation Tasks

Ander Martínez
Fujitsu, Ltd.
4-1-1, Kamikodanaka, Nakahara-ku,
Kawasaki 211-8588, Japan
ander@fujitsu.com

Abstract

This paper describes the Fujitsu DMATH systems used for WMT 2021 News Translation and Biomedical Translation tasks. We focused on low-resource pairs, using a simple system. We conducted experiments on English-Hausa, Xhosa-Zulu and English-Basque, and submitted the results for Xhosa→Zulu in the News Translation Task, and English→Basque in the Biomedical Translation Task, abstract and terminology translation subtasks. Our system combines BPE dropout, sub-subword features and back-translation with a Transformer (base) model, achieving good results on the evaluation sets.

1 Introduction

WMT has been exploring the state of the art in MT for many years, and, particularly in recent editions, the participants have shown impressive results. However, often times, these results require very heavy or complex systems, trained on dozens of GPUs. Participants compete for a margin that places them above the rest, combining multiple methods from the latest research.

In recent years, different variants of the Transformer (Vaswani et al., 2017) architecture have been popular for NMT, so can be seen when inspecting the submissions to previous editions of WMT. In our systems, we use the Transformer `base` configuration, the smaller one. Our implementation is based on Sockeye 2 (Hieber et al., 2020; Domhan et al., 2020).

We combine several techniques or strategies for low-resource pairs. These techniques are described in Section 2.

We conducted a few experiments on language pairs Xhosa-Zulu and English-Hausa, from the News Translation task, and on English-Basque, from the Biomedical Translation task. The results of our experiments are shown in Section 3.

2 Techniques

This section describes the strategies used for our NMT models. The first two, *bpe dropout* and *sub-subword features*, were used in all the subtasks, while the last one was only used for the biomedical translation subtasks.

2.1 BPE dropout

BPE dropout (Provilkov et al., 2020) was introduced as an alternative to Kudo (2018). Provilkov et al. found that the main drawback to the subword regularization method is its complexity, since it requires training a unigram language model and uses uses EM and Viterbi algorithms to sample segmentations.

BPE dropout works on BPE vocabulary models (Sennrich et al., 2016b), that is, the vocabularies are built in the same way as vanilla BPE. While the unigram language model subword regularization method uses a statistical model and dynamic programming to be able to sample different segmentations from the same sequence, BPE dropout uses random noise to discard certain merges, randomly generating a different sequence of subwords each time. This is so because BPE does not store the frequencies of each subword, only the order of the merges. Merges are discarded with a probability p, which is usually 0.1. Provilkov et al. concluded through several experiments that BPE dropout achieves better results.

Our systems use BPE dropout during training, with a dropout proability p of 0.1.

2.2 Sub-subword features

The main idea of the Sub-subword feature method (Martinez et al., 2021) is to build the embedding matrices from the n-gram features of the subwords in the vocabulary. The features used to produce the embeddings are selected by an algrithm before training, and the neural network that produces the embeddings is trained with the rest of the model.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 162–166
November 10–11, 2021.

	Sentences	Words in source	Words in target	Word ratio
Xh-Zu	94,323	1,356,127	1,325,168	1.02
En-Ha	752,287	11,044,101	11,713,109	1.06
En-Eu	2,627,745	23,225,786	17,472,145	1.33

Table 1: Statistics of the datasets used for BT experiments. The rows of the table are ordered from smallest to largest, the source language being that of the pair. The ratios are the number of words of the largest language compared to the other.

The method has a regularizing effect, particularly effective under low-resource settings. The sub-subword feature method can be used with BPE and BPE dropout, to achieve better results.

2.3 Back-Translation

Back-translation (BT) (Sennrich et al., 2016a) can be used with monolingual data of the target language, to improve low-resource language pair performance. BT is a type of distant supervision, in which a model of the opposite direction to the one that one wants to build is used to synthesize more parallel data. The method requires training an opposite model, and the synthesized data is noisy. Still BT has been used extensively with good results reported (Poncelas et al., 2018; Edunov et al., 2018).

The effectiveness of the BT method depends largely on the quality of the monolingual corpora used. Monolingual corpora compiled automatically using web crawlers in combination with automatic language detection are prone to be noisy. Particularly for low-resource languages for which language detection has lower accuracy.

For example, we noted that the Hausa Extended Common Crawl corpus published for WMT21 contained a large number of Japanese song lyrics written in Latin alphabet.

Our systems used 2 million backtranslated sentences to improve performance.

2.4 Multilingual model

Johnson et al. (2017) introduced multilingual models to NMT. Multilingual models are capable of translating more than one pair. For this, they used a simple approach that consists of using a special symbol inserted in the source sentence, indicating the target language. The architecture of the model can be the same as that of non-multilingual models. In their experiments, they showed that, although the performance of pairs with more resources worsens when sharing a model with other pairs, the performance of pairs with fewer resources improves. Multilingual models allow translation between pairs with zero resources. This is known as zero-shot translation.

Much research has been done on Multilingual Neural Machine Translation (MNMT). Dabre et al. (2020) published a comprehensive survey that summarizes different ideas and techniques for MNMT.

For the English-Basque Biomedical task, we tried using multilingual models too. In particular, for the terminology translation subtask, we included the English-Spanish terminology from MeSpEN (Villegas et al., 2018). The terminology was included as training data, using the method described in this section. A more sophisticated vocabulary integration method could have given better results (Post and Vilar, 2018; Bergmanis and Pinnis, 2021).

3 Experiments

We conducted experiments on Xhosa → Zulu, Zulu → Xhosa, English → Hausa, Hausa → English and English → Basque. Notice that the WMT21 Biomedical Translation Task for English-Basque was only in the English → Basque direction, and not Basque → English.

Table 1 shows the statistics for three language pairs. The rows are ordered from smallest to largest. The Xhosa-Zulu and English-Hausa data were published in the WMT21 news translation task. Both are classified as low-resource in the task description, but Xhosa and Zulu are two closely-related languages, and English and Hausa, two distant languages. The English-Basque data were published for the biomedical task of WMT21. The English-Basque dataset cannot be considered low-resource, with 2.6M parallel sentences, but it represents two distant languages. The Basque language has a complex morphology that makes its generation difficult.

Word ratios can hint about the similarity or dissimilarity of the languages. Xhosa and Zulu are related languages, and that is why they show a ra-

	Xh→Zu	Zu→Xh	En→Ha	Ha→En	En→Eu
Baseline	6.5 (.416)	6.3 (.421)	12.0 (.412)	13.0 (.403)	16.5 (.456)
+SSWF	9.3 (.470)	8.5 (.468)	12.5 (.420)	14.5 (.429)	**17.3** (.471)
+BT	9.2 (.471)	8.6 (.467)	17.5 (.480)	**16.7** (.460)	16.4 (.462)
+BT+SSWF	**9.7** (.478)	**8.8** (.470)	**18.0** (.482)	15.5 (.461)	16.4 (.463)

Table 2: BT results for various language pairs. (+/-) BT indicates the use or non-use of BT data. The results follow the format "BLEU (CHRF2)". Best BLEU results are shown in bold and the best CHRF2 are underlined.

tio close to one. English and Hausa are distant languages, but their morphological characteristics result in sequences of similar length.

Table 3 shows the hyperparameters used to train the models. The Transformer hyperparameters are those of the *base* model. We use a relatively large vocabulary size of 32k subwords. Although Sennrich and Zhang (2019) showed that smaller vocabularies give better results on low-resource datasets, larger vocabularies work well when using sub-subword features (Martinez et al., 2021).

We used 4,000 warmup steps schedule as described in Vaswani et al. (2017) with an initial learning rate of 2.0 and evaluated the development cost every 2,000 updates. The model was reloaded from the best checkpoint when the development cost did not improve, and training stopped after 3 consecutive stallings.

Hyperparameter	Value
Vocabulary size	32,000 subwords
BPE dropout p	0.1
Batch size	4,096 (×2 GPUs)
Warmup steps	4,000
Learning rate	2.0
Encoder layers	6
Decoder layers	6
Attention heads	8
Transformer size	512
Hidden layer size	2,048
Dropout	0.1
Label smoothing ϵ	0.1
FTE layers †	3
FTE size †	3,072

Table 3: Hyperparameters used in our models. † FTE (feature-to-embedding) network size for sub-subword feature (+SSWF) models.

For the *News Translation Task* participants need agree to contribute to the manual evaluation about eight hours of work, per system submission. In consideration of this workload, we decided to submit only the Xhosa → Zulu system to the News Translation Task.

Table 2 shows the results for the languages in Table 1. The BT data were translated using the sub-subword feature (+SSWF) model. The BT data contain 2 million pairs of sentences. The English-Basque model shown in this table does not use the multilingual approach described in Subsection 2.4.

The results show that the sub-subword features (+SSWF) improve the results of the corresponding −SSWF models under low-resource settings. In the case of Hausa → English, the +SSWF system did not achieve better BLEU scores than the corresponding −SSWF system, but achieved better CHRF2.

Despite its noisy nature, we decided to use the *Extended Common Crawl* Hausa corpus. The results show that the data, although noisy, was effective in improving the performance.

The English → Basque biomedical abstract translation did not improve when using back-translation data. It is possible that the cause for this was the domain mismatch of the monolingual data, that was not exclusively from scientific papers' abstracts.

All models were trained on two *NVIDIA Tesla P100* GPUs. The Xhosa-Zulu models are trained in about 2.5 hours, and the English-Hausa models are trained in about 10 hours.

Table 4 shows the result of combining the English-Basque training data with the MeSpEN English-Spanish terminology (Villegas et al., 2018). The MeSpEN terminology dictionary that we used contained 125,519 term pairs after cleaning.

Model	BLEU	chrF2
En→Eu	16.47	.456
En→Eu +SSWF	17.34	**.471**
En→ {Eu,Es} +SSWF	**17.44**	.470

Table 4: NMT result of combining the English-Basque training data with the MeSpEN English-Spanish terminology.

The scores displayed were obtained by evaluating the trained models on a test set sampled from the provided data for abstract translation. The data used to build the development and test sets were removed from the training data. The results show the BLEU and CHRF2 scores for abstract translation, but we did not prepare any evaluation set for terminology translation, as we wanted to include WMT20 terminology in the training data.

The same models were used for abstract translation and terminology translation. Manual examination of the produced transations hinted better performance for the the model trained with English-Spanish terminology.

In consideration of the results, we decided to submit two systems to the abstract translation and terminology translation subtasks. One of the systems incorporated the MeSpEN terminology, and the other one did not. Both systems did not use backtranslated data.

4 Conclusions

We built and submitted three lightweight systems that used sub-subword features to build the embeddings. We evaluated the approach with different configurations and the results showed the adequacy of the approach.

The relatively small models could possibly use larger hyperparameters and other techniques to achieve better results, but we think the current results can show the strenght of the techniques that were applied.

References

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 2020. The sockeye 2 neural machine translation toolkit at AMTA 2020. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 110–115, Virtual. Association for Machine Translation in the Americas.

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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458, Lisboa, Portugal. European Association for Machine Translation.

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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Ander Martinez, Katsuhito Sudoh, and Yuji Matsumoto. 2021. Sub-subword n-gram features for subword-level neural machine translation. *Journal of Natural Language Processing*, 28(1):82–103.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. 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.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. 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*, volume 30. Curran Associates, Inc.

Marta Villegas, Ander Intxaurrondo, Aitor Gonzalez-Agirre, Montserrat Marimon, and Martin Krallinger. 2018. The mespen resource for english-spanish medical machine translation and terminologies: Census of parallel corpora, glossaries and term translations. *Language Resources and Evaluation*.

Adam Mickiewicz University's English-Hausa Submissions to the WMT 2021 News Translation Task

Artur Nowakowski and **Tomasz Dwojak**
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Poland
{artur.nowakowski,t.dwojak}@amu.edu.pl

Abstract

This paper presents the Adam Mickiewicz University's (AMU) submissions to the WMT 2021 News Translation Task. The submissions focus on the English↔Hausa translation directions, which is a low-resource translation scenario between distant languages. Our approach involves thorough data cleaning, transfer learning using a high-resource language pair, iterative training, and utilization of monolingual data via back-translation. We experiment with NMT and PB-SMT approaches alike, using the base Transformer architecture for all of the NMT models while utilizing PB-SMT systems as comparable baseline solutions.

1 Introduction

We describe the Adam Mickiewicz University's submissions to the WMT 2021 News Translation Task. We focused on translation between Hausa and English – a low-resource translation scenario between distant languages. Our methods combine data cleaning with OpusFilter (Aulamo et al., 2020) and fastText (Joulin et al., 2016), transfer learning (Aji et al., 2020; Zoph et al., 2016), iterative training, and back-translation (Sennrich et al., 2016a).

All NMT models were trained with FAIRSEQ (Ott et al., 2019), while the first iteration of the back-translation was generated with Moses (Koehn et al., 2007).

The results presented in the paper are based on the first released development set ("Dev-1"), which consists of 1000 sentences, the final development set ("Dev-full"), which adds additional 1000 sentences to the first development set, and the released test set without additional test suites ("Test"). The test set consists of 1000 sentences in English→Hausa direction and 997 sentences in Hausa→English direction.

The final submissions significantly outperform the vanilla NMT baselines in terms of BLEU (Papineni et al., 2002) metric results, as implemented in SACREBLEU (Post, 2018) with default settings.

All systems were trained in a constrained scenario i.e., using the data provided by the organizers of WMT 2021 only.

2 Data preparation

The quality of the training data has a great impact on the final performance of the NMT models (Rikters, 2018). The data preparation consisted of data cleaning and filtering performed by using OpusFilter (Aulamo et al., 2020) pipelines. We specified separate pipelines for monolingual and parallel data. Data cleaning phase consisted of normalizing punctuation, removing non-printable characters, and decoding HTML entities by using Moses (Koehn et al., 2007) pre-processing scripts.

We applied subword segmentation on filtered data by using SentencePiece (Kudo and Richardson, 2018) tool with byte-pair-encoding (BPE) (Sennrich et al., 2016b) algorithm. The corpora we used for model training, along with the number of sentences before filtering, are specified in Table 1. Number of sentences after filtering is presented in Table 2.

Monolingual data filtering For the monolingual data filtering, we defined an OpusFilter pipeline that consists of the following filters:

- deduplication filter,
- sentence length filter,
- word length filter,
- Latin character score filter,
- language identification filter.

The sentence length filter requires that the sentence contain a minimum of 3 and a maximum

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 167–171
November 10–11, 2021.

Data type	Sentences	Corpora
Parallel en-ha	751,560	Khamenei, Opus, ParaCrawl
Monolingual en	41,428,626	News crawl (only 2020)
Monolingual ha	2,311,959	News crawl, CommonCrawl
Parallel de-en	8,600,361	Tilde Rapid, CommonCrawl, Europarl, News commentary, ParaCrawl

Table 1: Corpora statistics before filtering.

of 100 words. A maximum of 40 characters is required for the word length. The required Latin character score for a sentence is set to 100%. Language identification filter is based on a fastText (Joulin et al., 2016) language identifier. The open-source fastText language identification models do not identify Hausa, so we used the JW300 corpus from the English-Hausa Opus collection to train our custom language identifier. A sentence must pass all filters to be included in the training data.

Data type	Sentences
Monolingual en	39,812,834
Monolingual ha	1,227,921
Parallel ha-en	494,246

Table 2: Monolingual corpora statistics after filtering.

Parallel data filtering The filters used in the parallel data filtering pipeline are nearly identical to those used in the monolingual data filtering pipeline. Filters are applied to both the source and target sentences in this scenario. We also included a length ratio filter with a threshold of 2, indicating that a sentence on the source side can be up to twice as long as a sentence on the target side and vice versa.

A similar pipeline was applied to the German-English data that was used for transfer learning. We downsampled 3M sentence pairs from ParaCrawl due to the imbalance in the German-English data.

3 Approach

Our models combine transfer learning from a high-resource language pair (German-English), iterative training, and back-translation. We used FAIRSEQ (Ott et al., 2019) toolkit in our experiments with NMT models, while we used Moses (Koehn et al., 2007) toolkit for our experiments with PB-SMT models.

All of our NMT models follow the base Transformer architecture (Vaswani et al., 2017), using ReLU as the activation function and Adam (Kingma and Ba, 2015) as the optimizer with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1\text{e}{-8}$. We set the inverse square root learning rate scheduling with a peak value of $1\text{e}{-3}$. We used learning rate warmup stage for 4000 updates with initial learning rate of $1\text{e}{-7}$. Dropout probability was set to 0.2, while the attention dropout probability was set to 0.1. We also used label smoothing with a value of 0.1. In the case of baseline English-Hausa models, the joint vocabulary was based on both English and Hausa data. In all cases, the vocabulary size was set to 32,000.

The PB-SMT models were trained with default settings with Moses (Koehn et al., 2007) toolkit. In addition, we trained a 5-gram Operation Sequence Model (Durrani et al., 2013). All language models are 5-gram models and were binarized with KenLM (Heafield et al., 2013). The models were trained on tokenized, word-level, lowercased sentences. Re-casing was applied to the model outputs. After training the base models, we also applied MERT (Minimum Error Rate Training) (Och, 2003; Bertoldi et al., 2009) tuning on the development set.

3.1 Baseline systems

We decided to train baseline models of two types: vanilla Transformer (base) and PB-SMT. The experiments conducted on the first release of the development set showed that PB-SMT performs significantly better than NMT: we achieved +1.8 BLEU score on Hausa→English and +0.7 on English→Hausa. Based on these results, we decided to use PB-SMT models to generate data for the first iteration of iterative training.

When the test set was published, we computed the scores for the baselines. To our surprise, the scores obtained by NMT are much higher than PB-SMT, especially in the Hausa→English direction.

System	HA → EN		EN → HA	
	Dev-1	Test	Dev-1	Test
NMT baseline	12.21	11.44	10.28	11.05
PB-SMT baseline	14.00	6.59	11.02	9.36

Table 3: Baseline results according to the automatic evaluation with BLEU metric.

3.2 Transfer learning

According to recent studies, transfer learning (TL) enhances translation quality in low-resource scenarios (Zoph et al., 2016; Aji et al., 2020). We chose the German→English translation direction as a base. In general, we followed (Nguyen and Chiang, 2017) and trained a shared Hausa-German-English vocabulary (BPE). Then, we trained a German→English model using parallel data from the WMT 2021 Translation Task, which was filtered similarly to Hausa-English data. Finally, we used the Hausa-English data to fine-tune the pre-trained German→English model. We obtained a BLEU score of 13.31 on the "Dev-1" development set (+1.1 BLEU compared to the NMT baseline), which was lower than the PB-SMT baseline.

3.3 Iterative back-translation

Monolingual data has been widely employed in MT to enrich parallel corpora with synthetic data to improve the quality of MT systems, particularly in low-resource scenarios (Bojar and Tamchyna, 2011; Bertoldi and Federico, 2009). We applied the back-translation technique (Edunov et al., 2018) iteratively (Hoang et al., 2018) to translate Hausa and English monolingual data into the other language, using intermediate models to generate incrementally better translations.

1. First, we used the best baseline model (PB-SMT based on Moses) in English→Hausa direction to translate 5M English sentences into Hausa.

2. We used this additional data to train the Hausa→English model by applying transfer learning from the German→English model. We upsampled the original parallel data 10 times to match the size of the back-translated data. We used the resulting NMT model to translate all Hausa monolingual data into English via sampling.

3. We combined the obtained back-translated data with the original parallel corpora to train the English→Hausa model in a manner similar to step 2, with the exception that we did not upsample the parallel data in this scenario due to the fact that back-translated data was generated through sampling.

4. This technique was applied iteratively, resulting in the systems shown in Table 4. In all Hausa→English systems except the last, we utilized 5M English monolingual sentences in the model training; in the last system, we used 25M sentences. We used all accessible Hausa monolingual data in all English→Hausa systems.

System	HA → EN	EN → HA
1	16.22	-
2	-	13.04
3	20.05	-
4	-	14.38
5	22.85	-
6	-	14.77

Table 4: Iterative back-translation results of the NMT systems on the "Dev-1" development set according to the automatic evaluation with BLEU metric.

4 Final results

Table 5 presents the final results for both the English→Hausa and Hausa→English translation directions for both the development and test sets. These results were produced by the final models from the iterative back-translation step described in section 3.3.

Direction	Dev-1	Dev-full	Test
EN → HA	14.77	21.21	16.15
HA → EN	22.85	25.23	14.13

Table 5: Final results according to the automatic evaluation with BLEU metric.

We notice a severe decrease in BLEU metric results on the test set as compared to the development set, particularly in the Hausa→English direction. This could suggest a domain shift between the two sets. Because our models are heavily based on the back-translated data, some vocabulary, especially proper names, may be missing from the training data.

5 Post-submission work

Due to a lack of computing power and time, our experiments and submissions were based on single model training. After the submission deadline, we retrained the final models three times with different seeds. Table 6 presents the results for the ensemble of four models in both directions. We obtained slight improvements on both test sets, but the differences are insignificant. On the other hand, the ensemble performed worse on the development set, especially on the first version.

Direction	Dev-1	Dev-full	Test
EN → HA	14.68	21.00	16.34
HA → EN	21.24	26.25	14.87

Table 6: Post-submission models ensemble results according to the automatic evaluation with BLEU metric.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156. Association for Computational Linguistics.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.

Nicola Bertoldi, Barry Haddow, and Jean-Baptiste Fouet. 2009. Improved minimum error rate training in moses. *The Prague Bulletin of Mathematical Linguistics*, 91:7–16.

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.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov models over minimal translation units help phrase-based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 399–405, Sofia, Bulgaria. 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*, pages 489–500, Brussels, Belgium. 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, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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, Taipei, Taiwan. 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 of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: 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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

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.

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.

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 International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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.

eTranslation's Submissions to the WMT 2021 News Translation Task

Csaba Oravecz† **Katina Bontcheva**† **David Kolovratník**†
Bhavani Bhaskar† **Michael Jellinghaus**⋆ **Andreas Eisele**⋆
DG Translation – DG CNECT, European Commission
†firstname.lastname@ext.ec.europa.eu
⋆firstname.lastname@ec.europa.eu

Abstract

The paper describes the 3 NMT models submitted by the eTranslation team to the WMT 2021 news translation shared task. We developed systems in language pairs that are actively used in the European Commission's eTranslation service. In the WMT news task, recent years have seen a steady increase in the need for computational resources to train deep and complex architectures to produce competitive systems. We took a different approach and explored alternative strategies focusing on data selection and filtering to improve the performance of baseline systems. In the domain constrained task for the French–German language pair our approach resulted in the best system by a significant margin in BLEU. For the other two systems (English–German and English-Czech[1]) we tried to build competitive models using standard best practices.

1 Introduction

The eTranslation team is behind the translation services of the European Commission's eTranslation project[2]. This is a building block of the Connecting Europe Facility (CEF), with the aim of supporting European and national public administrations' information exchange across language barriers in the EU. The project is described in more details in (Oravecz et al., 2019).

The team's participation in the WMT shared tasks has provided valuable insights to improve the quality of our production systems and allowed us to explore languages and domains beyond the formal language of EU institutions, leading to a continuous extension of the eTranslation service and helping in the search for the right balance between the use of resources in production environments and the best possible performance of models.

This year the team participated in the news translation shared task with 3 different language pairs: English → German, English → Czech and French → German. The selection was motivated by the fact that these language pairs can all be considered as high or medium resource, which is the main scenario in the eTranslation service, while the constrained domain in Fr→De offered a good opportunity to focus on and experiment with data selection and filtering techniques, which is a more viable alternative in our environment than the resource demanding (brute-force) increase in model complexity.

2 Data Preparation

Here we briefly describe the base data sets, the general selection and filtering methods we applied to prepare these initial data sets used to train the first models. Further data selection and augmentation methods to improve the quality of baseline models are described in Section 3.2. For all models we only used the provided parallel and monolingual data, so our 3 submissions fall into the constrained category.

2.1 Base Data Selection and Filtering

As a first baseline approach, we tried to make use of all provided original parallel (OP) data to build the first models for reference or back-translation. Since these data sets were fairly similar to those from last year we followed the same practice and trained baseline models from all OP data. There was, however, a significant increase in the ParaCrawl data, which for En→De for example, doubled its size. As it turned out, the increase in size did not necessarily mean a better translation model trained from the full data set so we explored different subsets based on scoring by both source and target language models (see Section 4.1 for the details of these experiments).

The domain distribution of the data sets was not

[1]Due to returning problems of resource availability, the En→Cs experiments did not finish until the submission deadline so we could finally only submit last year's system.

[2]https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 172–179
November 10–11, 2021.

Data set	En→De	Fr→De	En→Cs
Europarl v10	1.77M	1.79M	0.62M
Common Crawl	2.16M	0.56M	0.11M
News Commentary v16	0.38M	0.29M	0.25M^{v15}
Tilde Rapid corpus	0.99M	–	0.28M
Wiki Titles v3	1.31M	0.52M	0.32M^{v2}
ParaCrawl v7.1	79.2M	6.30M	4.90M$^{v5.1}$
WikiMatrix	5.46M	2.80M	1.92M
CzEng 2.0	–	–	41.6M
Total:	91.27M	12.26M	50.0M

Table 1: Number of segments in the filtered parallel data used for baseline models.

uniform across language pairs, which had some influence on some of the workflows but the basic procedure of data cleaning was similar in all cases. As a general clean-up, we performed the following steps on the parallel data:

- language identification with FastText[3] (Joulin et al., 2016),
- segment deduplication with masked numerals, i.e. we deleted duplicate segments regardless of differences in numerals,
- deletion of segments where source/target token ratio exceeds 1:3 (or 3:1),
- deletion of segments longer than 100-150 tokens (depending on language pair),
- exclusion of segments where the ratio between the number of characters and the number of words was below 1.5 or above 40,
- exclusion of segments without a minimum number of alphabetic characters (2–5 depending on the data set).

These filtering steps led to an average reduction of about 15-20% of the training data with the number of segments as shown in Table 1.

2.1.1 Monolingual data

To build language models or create synthetic parallel text from monolingual data, we generally selected recent target language News Crawl data sets filtered according to the above steps (where applicable) with some minor adjustments. For En→De, we used the 2016–2020 German News Crawl data but as in the previous years excluded the 2018 set due to the high number of garbage segments with scrambled tokens, we set a threshold on the maximum length of a token (40) and the minimum ratio of letters to digits in a segment (4), and reduced the maximum segment length to 80 tokens, resulting in a 167M segment monolingual German data set. A similar procedure applied to the 2016–2020 English NewsCrawl corpus resulted in a monolingual English data set of 133M segments.

To create domain specific back-translation data for Fr→De we used the same data as for En→De, but due to the document based filtering method (see Section 3.2.2) the versions with document boundaries were used.

2.1.2 Development and test data

Development and test data sets were selected from the development suites provided. For En→De, we used the 2019 test set as validation set in the trainings and the 2020 test set as the test set to evaluate the trained models[4]. These data sets already contained only source original segments. We also extracted a source original subset from the full En→De development set, which was used in fine tuning of the final En→De models (see Section 3.2.3).

For Fr→De, the development set was shuffled and split into 3000 segment pairs for validation set and the rest (1813 segment pairs) for a general test set. To get an indication of the effect of data selection as described in Section 3.2.2, it was necessary to create a domain specific custom test set as well. The Fr→De 2008–14 development sets were filtered using a pattern based approach based on a

[3] https://fasttext.cc/docs/en/language-identification.html

[4] The reverse direction was used for the back-translation engines.

small list of 50 manually selected domain specific keywords[5], as well as scored and ranked by a target language model built from selected monolingual data (see Section 3.2.2). These two candidate lists were then manually revised and filtered to result in a 2k domain specific test set. These segments were removed from the training data.

2.2 Pre- and Postprocessing

Similarly to previous years (Oravecz et al., 2019, 2020) we opted for the simplest possible workflow leaving out the standard pre- and postprocessing steps of truecasing, or (de)tokenization, and simply used SentencePiece (Kudo, 2018), which allows raw text input/output within the Marian toolkit (Junczys-Dowmunt et al., 2018)[6] in the experiments. In some language pairs some simple normalization steps were applied in post-processing, which are described in the language pair specific result sections.

3 Trainings

In competitive systems big transformer architectures have become the norm in recent years (Barrault et al., 2020). We can in general see a significant increase in the need for computational resources to train deeper and more complex architectures up to 40–50 encoder layers (Wu et al., 2020b; Zhang et al., 2020; Wu et al., 2020a). Our resource environment does not allow us to fully follow this trend, limiting the complexity of the models as well as the scope of the experiments. Similarly to previous years, in all experiments we used Marian, as the core tool of our standard NMT framework in the eTranslation service. All trainings were run as multi-GPU trainings on 2 or 4 NVIDIA V100 GPUs with 16GB RAM, while for one training we were able to use a server with 8 32GB V100 GPUs.[7] Base transformers were typically trained for 20-30 epochs, whereas big transfomers were generally trained for 4–9 epochs for very high resource setups (>400M segments) and 20–25 epochs for medium resource.

[5]For example: Abwicklung, Betrug, Finanzbeitrag, Kapital etc.

[6]We did not change the default settings for Marian's built-in SentencePiece: unigram model, built-in normalization and no subword regularization.

[7]Access to high capacity resources at an affordable price has been especially challenging for us this year. In a race where computational power plays a crucial role (particularly in high resource settings) this might lead to an inherent disadvantage, which can be difficult to handle.

3.1 NMT Models

We only used base transformer models (Vaswani et al., 2017) for the first baseline models and for models used for back-translation to gain time and efficiency in back-translating large amounts of target monolingual data. For more competitive systems we switched to big transformer architectures, which resulted in significant improvements but at the same time the rise in computing costs and training time was also substantial. Due to the limitations of available resources we could build only one set of a 2–4 member ensemble from big transformers as our submission systems for En→De and Fr→De; again a high cost for a relatively smaller scale improvement. Our training settings have not changed from last year's setup: for most of the hyperparameters we used the default settings for the base transformer architecture in Marian[8] with dynamic batching and tying all embeddings. To save time and resources, we stopped the trainings if sentence-wise normalized cross-entropy on the validation set did not improve in 5 consecutive validation steps. In the big transformer experiments, also following recommended settings for Marian, we doubled the filter size and the number of heads, decreased the learning rate from 0.0003 to 0.0002 and halved the update value for `--lr-warmup` and `--lr-decay-inv-sqrt`.

Following common ranges of subword vocabulary sizes, we set a 36k joint SentencePiece vocabulary in En→De and En→Cs, and 30k in Fr→De.

3.2 Improving Baseline Models

In this section we briefly describe the methods we experimented with to improve the baseline models, such as selecting and filtering domain specific monolingual corpora to build additional synthetic data sets with back-translation (Sennrich et al., 2016), using development data (where available) or language model scored subsets of original parallel data to continue the training of already converged models and building ensembles of deep models originally trained from different seeds. Evaluation scores are reported in Section 4.

3.2.1 Filtering ParaCrawl

Training the En→De baseline model from the original parallel (OP) data (Table 1) we noticed that the model performed only as well (32.8 BLEU

[8]See eg. `https://github.com/marian-nmt/marian-examples/tree/master/transformer`.

on the 2020 test set) as our comparable model from last year despite having about twice as much ParaCrawl data while the other datasets remained basically very similar. This suggested that the v7.1 ParaCrawl (PC) data might have been noisier or contained more out of (news) domain data than expected. This was confirmed by training an alternative baseline excluding the whole ParaCrawl data set, which in the end resulted in a better score (33.3). To find a more beneficial subset of the PC data we first experimented with the stock Bicleaner filtering (Ramírez-Sánchez et al., 2020), setting higher thresholds of 0.65 and 0.75, which filtered the PC data to 51M and 26M segments, respectively. Adding either of these subsets to the other OP data sets did not lead to a significant increase (33.4 in both setups), however, we used the 51M segment subset instead of the full PC data in some further filtering experiments (see Section 3.2.3).

As a second filtering method we trained transformer language models (LM) with Marian from the filtered monolingual English and German data sets, scored both sides of the ParaCrawl data and ranked the segments (by simply averaging the scores). We experimented with models trained by adding the top 10, 20 and 30M highest scoring PC segments to the other OP data and found the 20M segment subset to produce the best baseline score (35.2), therefore we selected this data set (non ParaCrawl OP data plus the 20M segment LM scored ParaCrawl subset) as the initial parallel data for more complex models as well as for back-translation.[9]

3.2.2 Synthetic Data

Back-translation (BT) is the most used data augmentation technique in neural machine translation, but one which can introduce a wide range of scenarios in the search for finding the most optimal setup in the amount of synthetic data, the ratio of bitext to back-translation data or in the methods to generate the synthetic source (Edunov et al., 2018; Hoang et al., 2018). Tagged back-translation (Caswell et al., 2019) has been proposed as a simple and efficient alternative to noising techniques, arguing that it is the indication of the data being synthetic that is relevant for the model. This has been confirmed in our experiments in previous years, therefore we tried to use this technique in our workflows.

In the En→De system, we trained the reverse engine as a base transformer from the best baseline data setup mentioned above. After the convergence of this model we continued the training with a 30M segment subset of the OP data created by language model scoring (with the same models as for ParaCrawl). This gave an additional small increase in BLEU (0.4). With this model we back-translated an aggressively sentence segmented version of the filtered German monolingual data (see Section 2.1), which increased the size of the training set from the initial 167M segments to 219M. Our first intention was to build strong sentence based models and postprocess their output with dedicated sentence-to-document methods (which we describe in Section 3.2.5), so we tried to build one sentence per segment back-translated data sets by splitting up segments containing several sentences.

To train the submission ready systems we upsampled the best baseline OP data set to a 1:1 ratio with the BT data (Ng et al., 2019; Junczys-Dowmunt, 2019). This setup was a one shot configuration, we had no time and resources to experiment with other OP-BT combinations.

The task in the Fr→De language pair was domain specific, which offered us the opportunity to follow suit with the more recent shift from model centric approaches to data centric ones and focus on methods for finding the optimal subsets of the provided data which help improve performance in the selected domain. Therefore we tried to tune our models towards the domain by making use of guided topic modeling[10]. We created financial seed word lists by manually selecting 40 and 175 domain specific tokens from the top of a raw frequency list from a few million German News Crawl segments, and then we clustered the documents in the 2016, 2017, 2019 and 2020 German News Crawl data set into different topics guided by the selected seed word list.[11] By selecting the documents clustered into the seed word list induced topic we finally collected ca. 12M German News Crawl segments derived from two topic modelling runs based on one or the other list. These segments overlapped to a great extent. We back-translated both selections then cleaned up the back-translated data the way

[9] Clearly, there are other data selection combinations possible, for example, by taking only the 0.65 threshold Bicleaner subset as the base data for the LM based filtering, however, we did not have the time and resources to explore more scenarios for this language pair.

[10] https://github.com/vi3k6i5/guidedlda

[11] The text was tokenized and we used a German stopword list but no lemmatization in creating the document-term matrices.

we cleaned up the OP data but removed additionally pairs of segments that contained more than 15 numeric characters or more than 15 non-decimal commas. We also used the two sets to train two domain specific language models to score and rank the original parallel data set.

After that we took the union of the filtered BTs and deduplicated it. This gave us ca. 15M BT segment pairs which was at almost 1:1 ratio with the OP data. We explored training with subsets of the BT data but this did not give any improvement so we decided to use it all. We also experimented with tagged and untagged BT data, of which somewhat unexpectedly the latter gave the better result. The reason might be that the BT data was more in-domain, while most of the OP data was out of (news) domain and the explicit OP vs. BT distinction might have presented a harmful signal to the model here.

3.2.3 Continued Trainings and Fine Tuning on Dev Sets

As last year, in the En→De system we followed a two-stage continued training process to improve performance as domain adaptation (Luong and Manning, 2015). We scored the non ParaCrawl OP plus the 0.65 threshold ParaCrawl subset (see Section 3.2.1) with the language models used for filtering the ParaCrawl data set (Section 3.2.1). Then we used the top 10, 20 and 30M subset to continue the training of the OP+BT converged models until the BLEU score on the test set increased (Junczys-Dowmunt, 2019); typically 2 epochs with an increase of 0.5 points. The second stage utilized the 2008–2019 development sets (34k segments) as fine tuning data in the experiments and for the final submission it was extended with the 2020 test set. We trained with reduced batch size and learning rate for 4 epochs on this set and then for additional 3 epochs we switched to a source original subset (16k) to reach the highest BLEU score. In the end this process gave only a minor improvement of 0.3 BLEU points.

For Fr→De, we experimented with fine-tuning the best converged models (see Section 4.2) by using different sets of in-domain data. We scored the OP data for domain, using the two different LMs as mentioned above. Then, we selected the top 1M segments of each scored set of OP data and intersected them. This gave us ca. 0.85M segment pairs. However, this approach was not successful. In the other setup, we selected the top 2M segments of each scored set of OP data and intersected them, which gave us ca. 1.75M segments. We fine-tuned with reduced batch size until the BLEU score increased, which gave us an increase of 0.8 points on the domain specific test set.

3.2.4 Ensembles

The En→De final submission consisted of a modest 4 model big transformer ensemble, trained with the same best configuration and workflow but with different seeds. This approach usually gives a small but steady improvement (about 0.5 BLEU points here) but for substantially high resource settings it also comes with large computational costs. It is not uncommon to use ensembling already for back-translation (Wu et al., 2020b) but for lack of time and resources we had to limit this technique to the submission setups.

The Fr→De ensemble was composed of 4 big transformer models – three of them trained on original parallel data and back-translated data in ratio 1:1. The 4th big transformer was one of the 3 big transformers, additionally fine-tuned for 7 epochs on the 1.75M OP data scored with the domain LMs. For lack of time it was only one experimental setup out of many other possible ones but proved to be better than our previous systems.

3.2.5 Methods Tested but not Selected for Submission Models

In the En→De system, this year we experimented with a two-stage translation process of using a strong sentence-level system at the first step and post-process its output with a dedicated sentence-to-document level model. Following the method proposed by Voita et al. (2019), we created a 100M segment synthetic dataset by round-trip translating the (filtered) 2019 and 2020 German News Crawl with document boundaries with the baseline sentence level (forward and reverse) systems, and then generating 1, 2, 3 and 4 sentence long "source German"–"target German" pairs from the round-trip translated segments and the sentences in the original News crawl documents. We trained a base transformer from this data set and used it as a second stage repair on the output of the best En→De sentence level system. Unfortunately, we observed a significant drop in BLEU (almost 5 points) and although this is somewhat consistent with what for example Ma et al. (2021) reports on automatic evaluation for this method, we did not want to take the risk of submitting a system with such a quality drop

on the automatic metric to manual evaluation.

4 Results

We submitted a constrained system for each of the 3 language pairs. For En→Cs, we ran out of time and had to reuse our last year submission. For the other language pairs, we provide the evaluation scores for models at important stages in the development, which reflect how the models got better as we tried various methods for improvement. All results are reported in detokenized BLEU.[12]

4.1 English→German

		Test sets	
System	Data	2020	2021
M1: Baseline	12M	33.3	–
M2: M1+PC	32M	35.2	–
M3: M2+BT^{bigT}	450M	36.7	–
M4: M3 tuned	450M+36k	37.5	–
M5: M4 ensemble	450M+36k	**38.0**	**29.6**

Table 2: Results for En→De models. The 2021 result is from the Ocelot submission.

In Table 2 we present the main stages of the development of the En→De systems. Model 1 was the initial baseline model and used only the original parallel data excluding ParaCrawl altogether. In Model 2 we added the language model filtered and scored top 20M subset from ParaCrawl (PC). For Model 3, we switched to the big transformer architecture and used the large aggressively segmented back-translation (BT) dataset with 1:1 upsampled original parallel data (OP). The next model (M4) was tuned for 3 additional epochs with the top 10M LM scored OP data and then with the development set, leading to a small but steady increase. Finally the system we submitted was an ensemble of four M4 models. Our primary system being a sentence-level model, we performed sentence segmentation as a preprocessing step and then simply remerged the sentence level hypotheses on the target side where needed. Finally, as in previous years, a post-processing step normalizing German punctuation and some space fixing around the % sign was run on the final output.

[12]sacreBLEU signatures: `BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+tok.13a+version.1.4.13`

4.2 French→German

Table 3 summarizes the results of the Fr→De experiments. The first baseline model (M1) was trained only on the original parallel data with news data upscaled 5 times (NewsCrawl, NewsCommentary), while in model 2 and 3 (M2, M3) we added the domain specific back-translated data set (as described in Section 3.2.2). Switching from base transformers (M1 to M3) to the big transformer architecture in model 4 (M4) led to a decent improvement. This setup was used for the models in the M5 three model ensemble. In the primary submission (M6) this was extended with a 4th big transformer. In M6, the 4 models were trained on the original parallel (OP) data and back-translated data (in ratio 1:1), and one of the models was additionally fine-tuned for 7 epochs on the 1.75M domain LM scored original parallel data subset (see Section 3.2.3).

4.3 English→Czech

Due to problems with computational resources, the En→Cs trainings had not finished until the submission deadline. Our primary submission presented in Table 4 is therefore a clone of the 2020 system (trained on OP plus BT data).

5 Conclusion

We presented the submissions of the eTranslation team to the WMT 2021 news translation shared task on 3 language pairs: English-German, French-German and English-Czech. Unlike in previous years, we had to face a few unexpected challenges with respect to resource availability, which inevitably affected some experiments we planned to carry out. We tried to put more emphasis on data selection, filtering and domain specific evaluation with custom test sets in the task where it seemed to be most rewarding and automatic evaluation results justified this approach.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages

System	Data	Test sets		
		Dev	Domain	2021
M1: Baseline	13.6M	27.1	–	–
M2: M1+BT (tagged)	27.3M	29.1	–	–
M3: M1+BT (untagged)	27.3M	30.9	–	–
M4: M3 as big Tr.	27.3M	31.9	24.0	–
M5: M4 ensemble	27.3M	32.5	24.5	40.2
M6: M5+FT	27.3M	32.3	**25.0**	**40.6**

Table 3: Results for Fr→De models. The 2021 results are from the Ocelot submissions.

System	Data	Test sets	
		2020	2021
Submission	166M	**35.7**	**21.5**

Table 4: Results for En→Cs models. The 2021 result is from the Ocelot submission.

1–55, Online. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. 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*, pages 489–500, Brussels, Belgium. 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, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. 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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75. 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.

Zhiyi Ma, Sergey Edunov, and Michael Auli. 2021. A comparison of approaches to document-level machine translation. *CoRR*, abs/2101.11040.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, Adrien Lardilleux, László Tihanyi, and Andreas Eisele. 2019. eTranslation's submissions to the WMT 2019 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 320–326, Florence, Italy. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, László Tihanyi, David Kolovratnik, Bhavani Bhaskar, Adrien Lardilleux, Szymon Klocek, and Andreas Eisele. 2020. eTranslation's submissions to the WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 254–261, Online. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

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.

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.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020a. The Volctrans machine translation system for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 305–312, Online. Association for Computational Linguistics.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020b. Tencent neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 313–319, Online. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345, Online. Association for Computational Linguistics.

The University of Edinburgh's Bengali-Hindi Submissions to the WMT21 News Translation Task

Proyag Pal **Alham Fikri Aji** **Pinzhen Chen** **Sukanta Sen**
School of Informatics, University of Edinburgh, Scotland
{proyag.pal, a.fikri, pinzhen.chen, ssen}@ed.ac.uk

Abstract

We describe the University of Edinburgh's Bengali↔Hindi constrained systems submitted to the WMT21 News Translation task. We submitted ensembles of Transformer models built with large-scale back-translation and fine-tuned on subsets of training data retrieved based on similarity to the target domain. For both translation directions, our submissions are among the best-performing constrained systems according to human evaluation.

1 Introduction

We present the University of Edinburgh's participation in the WMT21 news translation shared task on the Bengali→Hindi (Bn→Hi) and Hindi→Bengali (Hi→Bn) language pairs. We followed the constrained condition, i.e. only using the data provided by the organizers. The training data for these language pairs consisted of noisy crawled data, and was mostly out-of-domain with respect to the validation and test domain. Therefore, most of our efforts concentrated on fine-tuning models to adapt to the target domain. We also explore multiple back-translation methods, and ensembles of models trained and fine-tuned with different methods.

Building our systems consisted of the following steps, each of which is described in more detail in the remaining sections of this paper:

- Cleaning the noisy parallel data (Section 3).
- Training ensembles of Transformer models on the cleaned provided data for back-translation; and using the back-translated data along with the clean parallel data to train new models (Section 4).
- Fine-tuning the models on subsets of training data retrieved that are similar to the target domain, based on different similarity measures (Section 5).
- Ensembling various models and decoding with optimal parameters (Section 6).

We also report some methods that we tried to use but did not work in Section 8.

2 Model Configuration

Our models follow the Transformer-Big architecture (Vaswani et al., 2017): 6 layers of encoders and decoders, 16 heads, an embedding size of 1024, a unit size of 4096, etc. We found that smaller Transformer architectures performed worse.

All models are trained with the same vocabulary of 32k SentencePiece subwords (Kudo and Richardson, 2018) to allow ensembling. We use a shared vocabulary between source and target, as well as tied embeddings (Press and Wolf, 2017). We tried other vocabulary sizes too: 5k, 10k, and 20k, though all of them had similar performance. We also included several special tokens in the vocabulary, of which we finally used only one for tagged back-translation (Caswell et al., 2019).

We train models with 32GB dynamic batch size and an optimizer delay (Bogoychev et al., 2018) of 3 with the Adam optimizer (Kingma and Ba, 2015) under a learning rate of 0.0003, until we see no improvement within 10 consecutive validation steps. All models were trained with the Marian NMT toolkit (Junczys-Dowmunt et al., 2018)[1]

3 Datasets and Cleaning

3.1 Corpora

All our models are trained in the constrained scenario – even more specifically, we only use data provided for the news translation task for these specific language pairs. This consists of 3.3M parallel sentences from the CCAligned corpus (El-Kishky et al., 2020), along with monolingual data in both languages. The details of the corpora used along with their sizes are shown in Table 1.

[1] https://github.com/marian-nmt/marian

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 180–186
November 10–11, 2021.

Corpus	Lines (M)
Parallel	3.36
+ deduplication and filtering	2.03
Monolingual	
Bn NewsCrawl	10.1
Bn CommonCrawl	49.6
Hi NewsCrawl	46.1
Hi CommonCrawl	202

Table 1: Bn and Hi corpora used in our submissions.

3.2 Cleaning

Since the CCAligned corpus is built from web crawls and is known to be very noisy (Caswell et al., 2021), we focused on cleaning the parallel data before training translation models. Our main approaches are rule-based and heuristic cleaning methods, along with language identification and language model filters. Our final systems used the following cleaning methods for the parallel corpus:

De-duplication Duplicate sentence pairs – around 17.3% of the corpus – were removed.

Splitting multi-language sentences We observed large chunks of the corpus where the sentences on the Bengali side also had their English translations attached in the same line. Some rough punctuation and script-based heuristics were used to remove the English segments from these lines. The roughness of these heuristics also affected a large number of other lines, mostly noisy ones containing non-lexical information, but we observed no degradation of quality due to this inaccuracy. We also found some such sentences on the Hindi side, but they were less frequent and removal showed no improvement in quality, so we did not split Hindi sentences in this way for our final models.

Language ID filtering We used publicly available FastText language identification models (Joulin et al., 2016, 2017)[2] to filter out lines in wrong languages. We get the top 3 predictions for each line, throw out lines where the right language does not appear in the top 3 for one or both sides, sort by the language prediction probabilities, and based on manual inspection, arrive at minimum threshold probabilities of 0.6 for Bengali lines and 0.4 for Hindi lines, above which lines are retained.

[2] https://fasttext.cc/docs/en/language-identification.html

Language model filtering We used KenLM (Heafield, 2011) to train separate trigram language models for Bengali and Hindi, on all provided Extended CommonCrawl monolingual data, and used these to score the parallel data. We retain sentences with $\log_{10}$ probabilities greater than -4.

4 Training with Synthetic Data

In each language direction, we trained 4 models with different seeds. We then ensembled these 4 models to back-translate (Sennrich et al., 2016) all the provided monolingual data. We used this translated data in many different ways as described in the remainder of this section.

Tagged back-translation Following Caswell et al. (2019), we prefixed a special <__BT__> token to all back-translated news monolingual data, combined the data with the clean parallel data, and trained new models.

Two-step training We first trained models on all the back-translated data only, then once that converged, continued training on the clean parallel data. Since the amount of monolingual data far exceeds the amount of parallel data, this training regime gave us better results than mixing parallel and back-translated data at the same time. The latter method would also involve finding the right amount of back-translated data to sample/select, since using it all would overwhelm the parallel training data.

Forward translation We also trained models on parallel data along with all the back-translations and all forward translations, i.e. instead of strictly keeping target monolingual data on the target side and synthetic back-translated data on the source side, we used both directions of translated data.

5 Fine-tuning to the Target Domain

5.1 Fine-tuning on retrieved sentences

Unlike many of the other language pairs in the news translation task, the Bengali-Hindi pair does not include any known in-domain training corpora. The training data is aligned from documents obtained through untargeted web crawling (El-Kishky et al., 2020), and thus contains out-of-domain and noisy text. On the other hand, the target domain, reflected in the validation and test sets, consists of Wikipedia content[3].

[3] Despite it being part of the 'news translation' task

To adapt our models to the target domain, we retrieved sentences from the training corpora which are similar to the **source** side of validation and test sets based on different similarity measures, and then fine-tuned the models on these subsets of data. The remainder of this section describes the different methods to retrieve the relevant subsets of data. The number of sentence pairs retrieved by each of these methods which are then used for fine-tuning is shown in Table 2.

Retrieval	Source	Lines (K)	
		Bn	Hi
1 bigram overlap	dev	448	891
2 bigram overlap	dev	243	597
3 bigram overlap	dev	158	445
1 bigram overlap	dev, test	487	932
2 bigram overlap	dev, test	273	639
3 bigram overlap	dev, test	183	479
LM threshold -2.5	dev	50	175
LM threshold -2.0	dev, test	12	13
TF-IDF	dev, test	5.6	27.9
TF-IDF cluster	dev, test	20	20

Table 2: Number of training sentence pairs retrieved for fine-tuning by different methods.

Based on vocabulary overlap The simplest method is to retrieve any sentence pairs whose source texts have 1, 2, or 3 non-punctuation bigrams which occur on the source side of the validation and test sets. Due to the large mismatch between training corpus and target domain, this method retrieves a surprisingly small proportion of the training corpus, as shown in Table 2.

Based on language model scoring We trained n-gram language models on the validation and test set or validation set data only, scored the parallel data with these language models, then kept sentences scoring above a certain threshold. Even though the small size of the validation data means that the language model is probably not very good, we still see some improvements by fine-tuning on data retrieved this way.

Based on TF-IDF similarity We first adapted the document aligner[4] from ParaCrawl (Bañón et al., 2020) to work at sentence level. This tool uses the translation of a source text (Uszkoreit et al., 2010) to match potential target text using cosine similarity of TF-IDF-weighted word frequency vectors. In this case, we match the source side of our validation and test sets with the parallel text to find potential "matches". This method retrieves too few matches with only the validation set, but we got a few thousand sentence pairs (Table 2) from a combination of validation and test sets.

Following Chen et al. (2020b), we also developed a variant where we first cluster each source sentence with another X sentences in the validation and test sets based on n-gram TF-IDF vector cosine similarity, then treat the cluster as a single query and compare it against each source sentence in the parallel training data. We always picked the top 20K resulting pairs. Through manual inspection, we found that the resulting corpus is very reasonable when we cluster the whole validation and test sets as one query, making the fine-tuning essentially a test domain adaptation process.

5.2 Fine-tuning on the validation set

Since the validation data is the only domain-specific data we had, similar to Chen et al. (2020a), we fine-tuned all our final models on a portion of the validation set (we used 95% of the data instead of 75%) until it stopped improving on the rest of the validation set. This was done as a final additional step after the other kinds of fine-tuning described previously.

6 Ensembles and Decoding Parameters

6.1 Ensembles

As shown in Table 3, our primary submissions consist of ensembles of multiple models trained and fine-tuned in different ways. Due to the component models not being very high-quality, we observed that this type of ensemble produces more robust translations than simple ensembles of models trained identically with different seeds.

6.2 Optimal decoding hyperparameters

Using an initial ensemble of 4 models, we swept a wide range of values of beam size and length normalization hyperparameters to decode the validation set. We find that optimizing these can result in an improvement of up to 0.5 BLEU on the validation set. We obtained the best scores with a beam size of 16, and a length normalization parameter of 1.3 for Bn→Hi and 0.7 for Hi→Bn, and used these values to decode the test set.

[4] https://github.com/bitextor/bitextor/tree/master/document-aligner

	Model	Bn→Hi		Hi→Bn	
		BLEU	ChrF	BLEU	ChrF
(1)	Single model baseline – Parallel data	19.56	0.4638	10.70	0.4378
(2)	Ensemble – Parallel data	20.37	0.4733	11.47	0.4482
(3)	Parallel + back-translated data	18.62	0.4577	9.78	0.4360
(4)	Parallel + backward + forward translations	20.16	0.4697	11.78	0.4503
(5)	Continue training on (3) with parallel data	21.26	0.4784	12.29	0.4587
(6)	Continue training on (4) with parallel data	20.97	0.4767	12.02	0.4470
(7)	Tagged BT (NewsCrawl only) + parallel data	20.61	0.4753	12.13	0.4541
	(5) fine-tuned on:				
(8)	1 bigram overlap, dev	21.55	0.4816	12.26	0.4573
(9)	2 bigram overlap, dev	21.49	0.4806	12.31	0.4587
(10)	3 bigram overlap, dev	21.35	0.4803	12.44	0.4600
(11)	LM threshold -2.5, dev	21.30	0.4794	12.29	0.4590
(12)	1 bigram overlap, dev+test	21.45	0.4814	12.29	0.4599
(13)	2 bigram overlap, dev+test	21.52	0.4812	12.21	0.4568
(14)	3 bigram overlap, dev+test	21.38	0.4794	12.26	0.4594
(15)	LM threshold -2.0, dev+test	21.29	0.4792	12.24	0.4563
(16)	TF-IDF, dev+test	21.32	0.4788	12.32	0.4601
(17)	(6) fine-tuned on TF-IDF cluster, dev+test	20.26	0.4710	12.02	0.4470

Table 3: Validation set BLEU and ChrF scores for our models.

Submitted ensembles	Bn→Hi		Hi→Bn	
	BLEU	ChrF	BLEU	ChrF
(8)+(9)+(10)+(11)	21.75	0.4895	–	
(6)+(7)+(8)+(9)+(10)+(11)+(16)+(17)	–		12.55	0.4536

Table 4: Test set BLEU and ChrF scores for our primary submissions. Model numbers refer to models from Table 3, but note that all models were fine-tuned on the validation set before ensembling.

6.3 Sentence splitting

In the source texts of the test set, we observed many instances of more than one sentence in one line. Since our models are trained on single sentences, we chose to run a sentence splitter on the test source, translate, and rejoin the translated sentences. For this purpose, we used the Moses sentence splitter (Koehn et al., 2007)[5] for Bengali text, and the IndicNLP sentence splitter (Kunchukuttan, 2020) for Hindi.

6.4 Numeral transliteration

Due to the fact that numerals in the Latin script are often used in Bengali and Hindi text, which is reflected by the web crawled training data, our models tend to generate a mix of Latin and Bengali/Hindi numerals, sometimes even in the same sentence. To ensure consistency, we transliterated all Bengali or Hindi numerals in our test outputs to their Latin script counterparts (it is equally feasible to convert Latin numerals to the target language). While this may not help in terms of automatic metrics (we lose 0.3-0.5 BLEU after this step), we believe human evaluators would prefer consistency in this regard.

7 Results

Table 3 shows BLEU[6] and ChrF[7] scored using sacreBLEU (Post, 2018) on the validation sets. We see that fine-tuning on the retrieved subsets of data consistently results in quality gains. We tried many different ensembles and, upon visual inspection, found that models fine-tuned on data retrieved on the basis of similarity to validation and test sets were not necessarily better than those from validation sets only.

[5] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/ems/support/split-sentences.perl

[6] signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1

[7] signature: chrF2+numchars.6+space.false+version.1.5.1

Ave.	Ave. z	System
82.1	0.202	GTCOM
79.1	0.163	Online-B
77.5	0.080	TRANSSION
78.0	0.076	MS-EgDC
78.0	**0.054**	**UEdin**
76.1	-0.015	Online-Y
75.7	-0.080	HuaweiTSC
75.7	-0.107	Online-A
70.8	-0.373	Online-G

(a) bn→hi

Ave.	Ave. z	System
95.0	0.245	HuaweiTSC
94.8	0.236	Online-A
94.5	0.233	GTCOM
94.6	**0.214**	**UEdin**
92.3	0.080	Online-Y
92.0	0.045	TRANSSION
91.3	0.029	Online-B
90.9	-0.008	MS-EgDC
73.5	-1.100	Online-G

(b) hi→bn

□ constrained ▨ unconstrained

Table 5: Human evaluation results. Our submissions are in bold. Systems within a cluster are considered tied.

Table 4 reports the automatic scores of our final submitted systems on the test sets. As shown in Table 5, according to human evaluation conducted by the task organizers, our systems rank at the top (tied) among all the constrained submissions for both translation directions.

8 Unsuccessful Attempts

In this section, we document some methods that we tried to use, but which did not work at all or did not result in better systems.

Dual conditional cross-entropy filtering Our initial cleaning effort was to use dual conditional cross-entropy (Junczys-Dowmunt, 2018) to self-filter the parallel data, which yielded no useful results. We also randomly split the data into two halves, trained translation models on each half, to score and filter the other half of the data – this method did not work either. We conclude that these methods are not suitable in this scenario where we do not have any clean data, however small, to train the initial cleaning model.

Copied monolingual data We attempted to synthesize training data by copying (Currey et al., 2017) and transliterating[8] monolingual data in the target language to source. In this way, we obtained pseudo parallel data that could potentially improve the decoder side of a translation model without harming the encoder much.

Transfer learning We also explored utilizing dataset from another language in the form of model pre-training. Following Aji et al. (2020), we initialize our Bengali↔Hindi model weights, excluding the embeddings, from our English↔German submission to WMT21 (Chen et al., 2021).

These methods above did not increase BLEU, except that transliterated monolingual data brought a tiny improvement. Model pre-training achieved the convergence faster, but did not achieve better final BLEU. Consequently, we did not carry out any further experiments with these methods.

[8] https://github.com/indic-transliteration/indic_transliteration_py

Acknowledgements

This work used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk) funded by the University of Edinburgh and EPSRC (EP/P020267/1).

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), 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).

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Gov-

ernment is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Nikolay Bogoychev, Kenneth Heafield, Alham Fikri Aji, and Marcin Junczys-Dowmunt. 2018. Accelerating asynchronous stochastic gradient descent for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2991–2996.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, D. Esch, Nasanbayar Ulzii-Orshikh, A. Tapo, Nishant Subramani, A. Sokolov, Claytone Sikasote, Monang Setyawan, S. Sarin, Sokhar Samb, B. Sagot, Clara Rivera, Annette Rios Gonzales, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, Toan Q. Nguyen, Mathias Muller, A. Muller, S. Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, M. Jenny, Orhan Firat, Bonaventure F. P. Dossou, S. Dlamini, N. D. Silva, Sakine cCabuk Balli, Stella Rose Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, P. Baljekar, Israel Abebe Azime, A. Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. Quality at a glance: An audit of web-crawled multilingual datasets. *ArXiv*, abs/2103.12028.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020a. Facebook AI's WMT20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.

Pinzhen Chen, Nikolay Bogoychev, and Ulrich Germann. 2020b. Character mapping and ad-hoc adaptation: Edinburgh's IWSLT 2020 open domain translation system. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 122–129, Online. Association for Computational Linguistics.

Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. The University of Edinburgh's English-German and English-Hausa submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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, Copenhagen, Denmark. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, M. Douze, H. Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *ArXiv*, abs/1612.03651.

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*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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, 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.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1101–1109, Beijing, China. Coling 2010 Organizing Committee.

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*, volume 30. Curran Associates, Inc.

The Volctrans GLAT System: Non-autoregressive Translation Meets WMT21

Lihua Qian[*†], **Yi Zhou**[*†], **Zaixiang Zheng**[*†], **Yaoming Zhu**[†], **Zehui Lin**[†], **Jiangtao Feng**[†], **Shanbo Cheng**[†], **Lei Li**[‡], **Mingxuan Wang**[†] **and Hao Zhou**[†]
[†]Bytedance AI Lab [‡]University of California Santa Barbara
{qianlihua,zhouyi.naive,zhengzaixiang,zhuyaoming,linzehui}@bytedance.com
{fengjiangtao,chengshanbo,wangmingxuan.89,zhouhao.nlp}@bytedance.com
lilei@cs.ucsb.edu

Abstract

This paper describes the Volctrans' submission to the WMT21 news translation shared task for German→English translation. We build a parallel (*i.e.*, non-autoregressive) translation system using the Glancing Transformer (Qian et al., 2020), which enables fast and accurate parallel decoding in contrast to the currently prevailing autoregressive models. To the best of our knowledge, this is the first parallel translation system that can be scaled to such a practical scenario like WMT competition. More importantly, our parallel translation system achieves the best BLEU score (35.0) on German→English translation task, outperforming all strong autoregressive counterparts.

1 Introduction

In recent years' WMT competitions, most teams develop their translation systems based on autoregressive models, such as Transformer (Vaswani et al., 2017). Although autoregressive models (AT) achieve strong results, it is also worth exploring other alternative machine translation paradigm. Therefore, we build our systems with non-autoregressive translation (NAT) models (Gu et al., 2018). Unlike the left-to-right decoding in the autoregressive models, the NAT models employ the more efficient parallel decoding. Specifically, our system employs single-pass parallel decoding, which generates all the tokens in parallel at one time, thus can accelerate decoding speed.

In this paper, we would like to present the best practice we explored in this year's competition for our parallel translation system, aiming at achieving top results while preserving decoding efficiency.

System Overview. To achieve this, we improve the parallel translation system in several aspects, including better model architectures, various data exploitation methods, mutli-stage training strategy, and inference with effective reranking techniques. For model architectures (§2), we build the parallel translation system based on the Glancing Transformer (GLAT, Qian et al., 2020). Besides, our system employs dynamic linear combination of layers (DLCL, Wang et al., 2019) for training deep models. For data exploitation (§3), we first filter data with multiple strategies. After filtering, we use the Transformer (Vaswani et al., 2017) to synthesize various distilled data. For training (§4), the NAT models employ multi-stage training to better exploit the distilled data. At inference phase (§5), the system generates the final results by reranking candidate hypothesis from multiple parallel generation models.

With the proposed techniques, our parallel translation system surpasses autoregressive models, and achieves the highest BLEU score (35.0) in the German→English translation task. Such results show that parallel translation system not only has great decoding efficiency, but also could achieve better performance compared to the autoregresssive counterparts.

2 Backbone Model Architecture

As depicted in Figure 1, our submitted system employs GLAT (Qian et al., 2020) as our backbone model architecture, and includes an auxiliary decoder in GLAT for achieving better translation performance. GLAT is a method for training non-autoregressive models rather than a model architecture, which adaptively samples target tokens in training. Although the target token sampling in GLAT helps training, it also introduces a gap between training and inference. To close the gap, we introduce the auxiliary decoder that shares the same encoder with the GLAT decoder, which is only used for training in a multi-tasking fashion.

[*]Equal contributions.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 187–196
November 10–11, 2021.

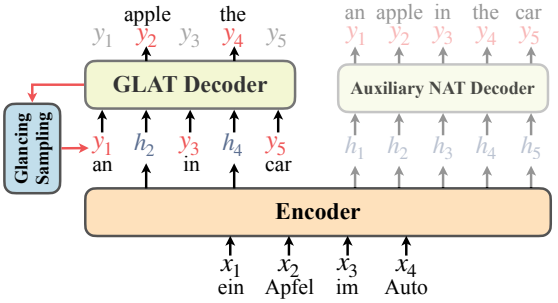


Figure 1: Illustration of our backbone model architecture: Glancing Transformer with an auxiliary decoder.

Besides, we train models with three architecture settings to increase model diversity.

2.1 Glancing Transformer

GLAT has three components: the encoder, the decoder, and the length predictor. The architecture of GLAT is built upon the Transformer (Vaswani et al., 2017). The encoder is the same as that of Transformer, and the decoder is different from the Transformer decoder in the attention mask. Transformer employs attention mask in self-attention layer to prevent decoder representations attending to subsequent positions. Since GLAT generates sentences in parallel, the decoder of GLAT has no attention mask and uses global context in decoding. The details of the length predictor is described in Section 2.3.

To reduce the difficulty of training deep models, we also employ dynamic linear combination of layers (DLCL, Wang et al., 2019) in the architecture. With DLCL, the input of each layer is the linear combination of outputs from all the previous layers.

Given the source input $X = \{x_1, x_2, ..., x_N\}$ and the target output $Y = \{y_1, y_2, ..., y_T\}$, we use the glancing language model (Qian et al., 2020) in training. The model performs two decoding during training. In the first decoding, the model generates the sentence $\hat{Y}$ in parallel. Then, the model randomly selects a subset of tokens $\mathbb{GS}(Y, \hat{Y})$ in the target sentence Y:

$$\mathbb{GS}(Y, \hat{Y}) = \text{Random}(Y, S(Y, \hat{Y})) \quad (1)$$

where $\text{Random}(Y, S)$ means randomly sample S tokens in Y. And the sampling number $S(Y, \hat{Y})$ is computed by $S(Y, \hat{Y}) = \alpha \cdot d(Y, \hat{Y})$. $d(Y, \hat{Y})$ is the Hamming distance between the first decoding result $\hat{Y}$ and the target sentence Y, and α is a hyper-parameter for controlling the sampling number more flexibly.

In the second decoding, the model replaces part of the original decoder input representations with the embeddings of tokens in $\mathbb{GS}(Y, \hat{Y})$. Specifically, the token y_i is used to replace the input representation at position i. With the replaced decoder inputs, the model learns to predict the remaining words and compute the training loss:

$$\mathcal{L}_{\text{glm}} = \sum_{y_t \in \overline{\mathbb{GS}(Y,\hat{Y})}} \log p(y_t | \mathbb{GS}(Y, \hat{Y}), X) \quad (2)$$

where $\overline{\mathbb{GS}(Y, \hat{Y})}$ is the subset of tokens in Y that are not selected. In training, the model starts from learning to generate sentence fragments and gradually learning the parallel generation of the whole sequence.

2.2 Auxiliary Decoder

Although the sampled target words in GLAT training help the model learn target word interdependencies, they also introduce a gap between training and inference as the model cannot obtain target word inputs in inference. Therefore, we add an auxiliary non-autoregressive decoder to close the gap. The auxiliary decoder shares the same encoder with the GLAT decoder and directly learns to predict the whole sequence in parallel. With the auxiliary decoder, we compute the loss for predicting the whole sequence:

$$\mathcal{L}_{\text{aux}} = \sum_{t=1}^{T} \log P_{\text{aux}}(y_t | X) \quad (3)$$

where P_{aux} is the output probability of the auxiliary decoder. We jointly train the two decoders and the training loss of model is:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{glm}} + \lambda \mathcal{L}_{\text{aux}} \quad (4)$$

Note that the auxiliary decoder is only used in training and has no additional cost in inference.

2.3 Length Prediction

To enable parallel generation, the model predicts the target length before decoding. We use the average of encoder hidden states H_{avg} as the representation to predict the length of target sentence. The probability of the target length is computed by:

$$P_{\text{len}} = \text{softmax}(H_{\text{avg}}^{\top} E_{\text{len}}) \quad (5)$$

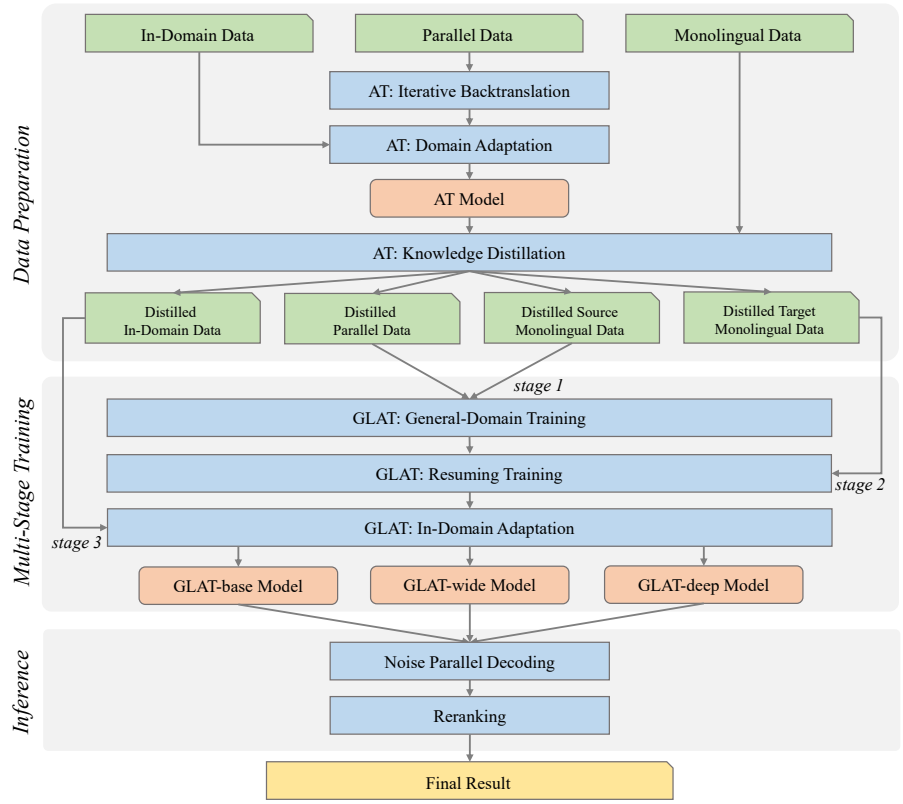


Figure 2: Overview of Volctrans GLAT System. Each grey block denotes a part of the system, the details can be found in Section 3: Data Preparation, Section 4: Multi-Stage Training, and Section 5: Inference.

where E_{len} is the embeddings of length. Instead of directly predicting the target length, the implemented model predicts the length difference between input and output, which is easier to learn. We use cross entropy loss for optimizing P_{len} and train the length predictor with the generation module jointly.

2.4 Model Variants

As shown in Figure 2, in order to increase the diversity of models, we use three model architecture settings for GLAT. The details of the three GLAT architecture variants are:

- GLAT-base: Following Wu et al. (2020);Sun et al. (2019), we increase the number of encoder layers and use 16 encoder layers for GLAT-base. For decoders, we use 6 layers for the original decoder and 2 layers for the auxiliary decoder. As for other model hyperparameters, we use the 1024 hidden dimension and 16 attention heads, which are the same as the setting of Transformer-big.
- GLAT-deep: We further increase the number of encoder layers to 32 for GLAT-deep. To keep the number of model parameters on the same scale, we decrease the hidden dimension to 768.
- GLAT-wide: Following previous work (Wu et al., 2020), we also expand the dimension of the feed-forward inner layer to construct GLAT-wide. We set the feed-forward dimension to 12288 and the encoder layer number to 12.

3 Data Preparation

In this section, we will describe our best practice of distilled data construction by employing AT models. As illustrated in *data preparation* in Figure 2, we will first depict the general procedure of data filtering and preprocessing of the provided raw data, followed by the training details of the AT models. Finally, we will describe how we produced distilled data given the trained AT models. The resulting distilled data will be used for training our GLAT system.

3.1 Data Filtering and Preprocessing

Data quality matters in machine translation systems. To obtain high-quality data, we employ rule-based heuristics, language detection, word alignment and

similarity-based retrieval to filter the provided parallel and monolingual corpora.

Rule-based Data Filtering

Based on experiences and WMT reports in previous years, we first preprocess raw data based on rules:

- Data deduplication.
- Delete parallel data with the same source and target.
- Remove special tokens and unprintable tokens.
- Remove HTML tags and inline URLs.
- Remove words or characters that repeat more than 5 times.
- Delete sentences that are too long (more than 200 words) or too short (less than 5 words), as well as the parallel data whose length-ratios of source and target sentences are out of balance.

Parallel Data Filtering

After completing the rule-based filtering, we further filtered parallel data via language detection and its parallelism. The filtering process consists of three stages:

1. Coarse-grained filtering: We filter parallel corpus according to the results and ratio of language detection. We use the pycld3[1] library to filter German→English sentence pairs with a language likelihood greater than 0.8 and a language ratio greater than 60%.
2. Word alignment learning: We use *fast align* (Dyer et al., 2013)[2] to automatically learn German→English word alignment on the coarsely filtered corpus.
3. Fine-grained filtering: We filter the sentences with an align score greater than five on all parallel corpora and sort them through the vocabulary learned by fast align.

Note that the amount of data in different corpora is not balanced. We split the data into the paracrawl group and the non-paracrawl group. We filter out about 10% of the data in the non-paracrawl group and 20% of the data in the paracrawl group.

Monolingual Data Filtering

For monolingual data, we first use the *pycld3* library to filter the data of low scores, similar to the coarse-grained filtering of parallel data.

Considering that monolingual data is too large, we searched for some of the most relevant sentences in our distilled data through sentence retrieval. We sample news domain sentences from the previous years' dev set and newscrawl corpus, and train a sentence BERT (Reimers and Gurevych, 2019)[3] to retrieve the sentences on the monolingual corpus. In detail, for each sampled news sentence, we calculate the inner product of sentence embedding between it and some random monolingual sentences (as the entire corpus is too large), where the sentence embedding is calculated with the sentence BERT model. We retrieved the top 8000 sentences for each news sample according to the inner product of sentence embedding. Finally, we deduplicate the retrieved sentences to obtain the final monolingual data.

	German (De)	English (En)
parallel data	75M	
monolingual data	86M	105M

Table 1: Statistics of the training data after preprocessing and filtering.

Data Preprocessing

Once we obtained filtered data, we preprocess them through the following steps:

1. Normalization: we use Moses tokenizer to normalize the punctuation.
2. Tokenization: we use Moses tokenizer to tokenize all datasets.
3. Truecasing: we use Moses truecaser to learn and apply truecasing on all datasets.
4. Subword segmentation: we use our proposed VOLT (Xu et al., 2021), which learns vocabularies via optimal transport, to split tokens into subwords, resulting in a joint vocabulary of a size of 12k subwords.

We summarize the statistics of the final datasets in Table 1.

3.2 Training of AT Systems

In this section, we describe our AT systems, which served to distill data for GLAT training. Overall, we first train a pair of German→English and English→German AT systems purely using parallel data. We then exploit source and target monolingual data to create synthetic parallel data to further improve the AT models. Besides, we leverage the

[1] https://pypi.org/project/pycld3/
[2] https://github.com/clab/fast_align
[3] https://github.com/UKPLab/sentence-transformers

testsets from previous years to fine-tune the AT models for in-domain adaptation.

Hyperparameters. The AT models are Transformer models with 12 layers of encoder and decoder. We use the implementations in Fairseq (Ott et al., 2019). All models are trained with Adam optimizer (Kingma and Ba, 2014). We use the inverse sqrt learning rate scheduler with 4000 warm-up steps and set the maximum learning rate to $5 \cdot 10^{-4}$. The betas are (0.9, 0.98). We use multiple GPUs during training, resulting in an approximate total effective batch size of 128k tokens. During training, we employ label smoothing (Szegedy et al., 2016) of 0.1 and set dropout rate (Srivastava et al., 2014) to 0.3.

Iterative Back Translation

Zhang et al. (2018) proposed an iterative joint training method for better usage of monolingual data from the source language (i.e., German) and target language (i.e., English). In each iteration, the German→English model generates forward synthetic data from the German monolingual data, and the English→German model generates backward synthetic data from the English monolingual data. Then, the German→English and English→German models are trained with the new forward and backward synthetic data to improve both models' performance, in which the target-side data are assumed to be the authentic ones from the monolingual corpus. In the next iteration, the German→English and English→German models can generate synthetic data with better quality, and their performance can be further improved . We jointly train the German→English and English→German models for 3 iterations.

In-domain Finetuning

We fine-tune the trained model on the previous years' testsets to obtain in-domain knowledge, which is a widely used technique in previous years' WMT (Li et al., 2019). Specifically, we use WMT19 German→English testset as in-domain data. We set the learning rate to 1e-4 without a learning rate scheduler and the max tokens per batch as 4096. We then fine-tune the model for 30 steps[4].

[4]Since the size of the in-domain data is small, fine-tuning with more steps will overfit the data.

	De-En	En-De
baseline	39.34	35.10
iterative BT	43.56	36.85
in-domain FT	44.00	38.30
forward translation	44.05	39.50
final training	44.15	39.70

Table 2: BLEU scores of AT models on `newstest20` with respect to different training stages.

Forward Translation

Bogoychev and Sennrich (2019) observed that on the sentences that are originally in the source language, which is the case of the test sets of this year's WMT, the forward translation could bring significantly more improvement than back-translation. We thus use the finetuned model, obtained by the aforementioned in-domain finetuning, to translate source monolingual corpus to obtain forward translation data. We then apply these forward translation data to finetune our AT models.

Finally, we combine all the parallel data, back-translation data, and forward translation data to further finetune our AT models. Table 2 shows the performance of the AT models with respect with each training stage. The resulting AT models are ready for constructing distilled data for GLAT training.

3.3 Constructing Distilled Data for GLAT

One of the widely known difficulties of training NAT models is the multi-modality problem (Gu et al., 2018). In the raw training data, the target tokens have strong correlations across different positions, which is hard to capture by NAT models due to the conditional independence assumption. A key ingredient in the training recipe for most of the NAT models is constructing training data via sequence-level knowledge distillation (Kim and Rush, 2016), where the target-side of the training data is replaced by the forward translation of AT models.

Note that previous work did not leverage existing large-scale monolingual data in training GLAT models, either from source or target language. In this work, we applied sequence-level knowledge distillation to parallel data and monolingual data from both source and target languages.

- Parallel data and source monolingual data distillation (119M sentences). We directly use

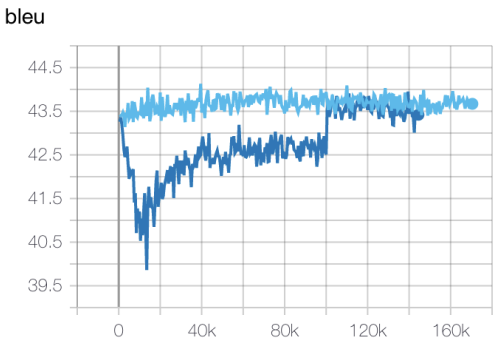


Figure 3: Learning curves of different finetuning strategies, reported on `newstest20`, De→En. The light blue curve denotes training with inverse square root scheduler where the peak learning rate equals $5 \cdot 10^{-4}$, and the initial sampling ratio λ is set to 0.5, the dark blue curve denotes training with a constant learning rate of $1e-4$ and $\lambda = 0.1$.

German→English AT model to obtain the forward translations of the German sentences.

- Monolingual target data distillation (39M sentences). The way to exploiting target monolingual data is not as evident as using the monolingual source data since the purpose of knowledge distillation is to construct a pseudo-parallel dataset where synthetic ones replace the actual target sentences. To this end, we propose a cycle distilling technique. We use the backward English→German AT model to back-translate the monolingual target data, resulting in a translated source dataset. We then used the German→English AT model to get the round-trip forward translation of the translated source dataset, obtaining the cycle distilled data. We will refer to this as *cycle KD data*.

4 Multi-Stage Training

We train our parallel translation system in a multi-stage way (See *Multi-Stage Training* in Figure 2). In the first stage, the model uses the distilled parallel and source monolingual data for training. In the second stage, we train the model with the target monolingual data (aka. *cycle KD data*). After training the model on large-scale distilled data until convergence, we finetune the model on small-scale in-domain data.

4.1 General-Domain Training

All models are trained with Adam optimizer with decoupled weight decay (Kingma and Ba, 2014; Loshchilov and Hutter, 2017). We use the inverse sqrt learning rate scheduler with 4000 warm-up steps and set the maximum learning rate to $5 \cdot 10^{-4}$. The adam betas are $(0.9, 0.999)$.

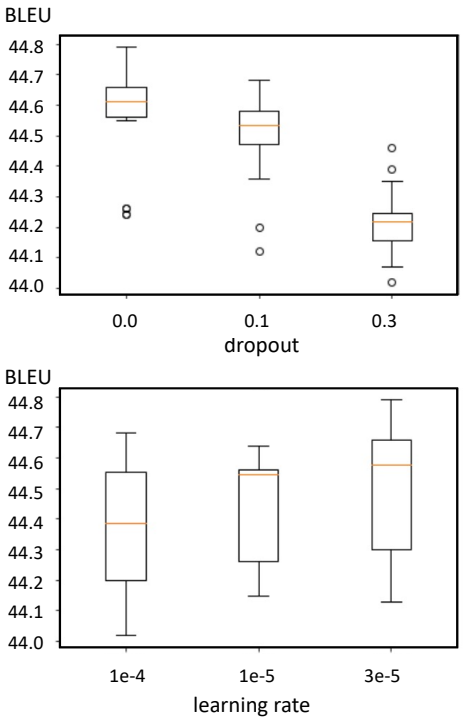


Figure 4: BLEU score versus the dropout and learning rate, reported on `newstest20`, De→En.

4.2 Resuming Training

We often have to load a pre-trained checkpoint and continuously train the model on a new dataset. The loaded checkpoint serves as a good initialization, and the parameters may change significantly in this process.

We found that it is not easy to apply the techniques from auto-regressive translation to GLAT directly. Preliminary experiments show that if we employ the techniques illustrated in (Qian et al., 2020) during the finetuning stage, the BLEU score will degrade dramatically and then increase slowly until convergence. The number of total update steps required for convergence is similar to training from scratch on a new dataset. There are mainly two concerns. Firstly, GLAT employs the inverse square root learning rate scheduler. The learning rate will increase to $5 \cdot 10^{-4}$ linearly and decay exponentially until the training process is over (the learning rate is close to $1e-4$). During the finetuning stage, a constant learning rate no larger than $1e-4$ will stabilize the training process. Secondly, the ini-

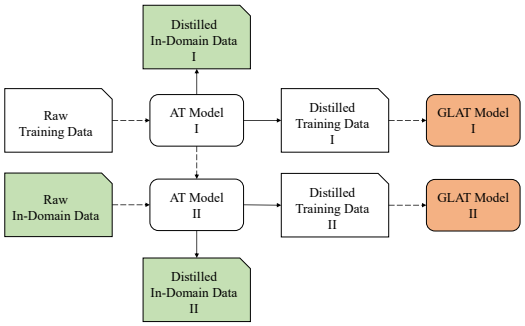


Figure 5: Various pipelines for domain adaptation.

In-Domain Data	GLAT-I	GLAT-II
-	0.00	+2.20
Raw	+1.51	+2.21
Distilled I	+0.30	+1.99
Distilled II	+1.56	+2.31

Table 3: Results of different adaptation pipelines. GLAT-I and GLAT-II are models trained with distilled training data generated by AT Model I and AT Model II in Figure 5, respectively. After training, we use the in-domain data Distilled I and Distilled II for fine-tuning.

tial sampling ratio $\lambda = 0.5$ in (Qian et al., 2020) can be too large for finetuning since the model can already do a good job in the translation task. A large sampling ratio may cause the model to suffer from "exposure bias"(Zhang et al., 2019): the gap between training (where some target words are provided) and validation (where no target words are provided). Figure 3 illustrates the comparison between two different finetuning strategies.

4.3 In-Domain Adaptation

When finetuning the model on small-scale in-domain data, which is widely used for domain adaptation (Meng et al., 2020), the parameters of the model do not change significantly.

For domain adaptation, we perform grid search on four group of hyper-parameters: learning rate($1e-5$, $3e-5$, $1e-4$), dropout(0.0, 0.1, 0.3), sampling rate λ (0.3, 0.1), and max number of tokens per batch (2000, 4000, 8000). For each combination, we conduct two experiments to reduce the variance. Experimental results (Figure 4) show that the learning rate and dropout rate are the most significant factors. Interestingly, when dropout is set to 0, the performance is surprisingly great, which indicates the effectiveness of over-fitting on an in-domain dataset.

feature groups	feature number
GLAT score	3
AT 16e6d	3
AT 12e12d	3
Self BLEU	1
Self Chrf	1

Table 4: Selected Features.

Model	BLEU	Self-R	AT-R
GLAT-base (w/o AUX)	42.28	42.54	42.90
+ CTC	41.04	-	-
+ AUX	43.1	43.11	43.52

Table 5: Results of different architectures, reported on `newstest20`, De→En.

There are several feasible pipelines for domain adaptation due to the interaction between auto-regressive and non-autoregressive models. Figure 5 illustrates these pipelines, and the key points are listed as follows:

- Should we finetune the auto-regressive model on the in-domain dataset (AT Model I→AT Model II)?
- Should we use the original in-domain dataset for GLAT's model adaptation or the in-domain dataset distilled by AT model I, or the in-domain dataset distilled by AT Model II?

Table 3 shows the results of different pipelines. Experiments show that making domain adaptation on the autoregressive model can boost the performance of the non-autoregressive model. It is also beneficial to further finetune the non-autoregressive model on the distilled in-domain dataset.

5 Inference

In this section, we introduce two approaches for GLAT's inference: Noisy parallel decoding (NPD) and Reranking (See *Inference* in Figure 2). NPD is easy to integrate into a single model and improve the performance; Reranking can help push the performance to the limit: generating as many candidates as possible and ranking them with as many features as possible.

5.1 Noisy Parallel Decoding

A simple yet efficient inference approach is noisy parallel decoding (NPD) (Gu et al., 2018). We

	GLAT-base			GLAT-deep			GLAT-wide		
Model	**BLEU**	**Self-R**	**AT-R**	**BLEU**	**Self-R**	**AT-R**	**BLEU**	**Self-R**	**AT-R**
baseline	43.10	43.11	43.52	42.44	43.89	43.14	43.38	43.49	43.81
+ cycle KD	43.40	43.24	43.77	42.86	43.51	43.73	43.51	43.49	43.79
+ adaptation	43.76	43.67	44.00	43.00	43.69	43.82	43.76	43.91	43.94
+ reranker					44.64*				

Table 6: Final results, reported on `newstest20`, De→En. * denotes the submitted system (BLEU=35.0 on `newstest21`, De→En). The baseline is GLAT w/ AUX.

first predict m target length candidates (in Table 5, $m = 5$), then generate output sequences with argmax decoding for each target length candidate. Then we use a model to rank these sequences and identify the best overall output as the final output. If the model for ranking and the one for generation is the same model (GLAT), we call it *Self-Reranking*; if the ranking model is AT, we call it *AT-Reranking*.

5.2 Reranking

We use kbmira[5] to re-rank hypotheses. We first train GLAT model variants of different settings, each of which produces a set of candidates via the various search algorithm in Section 2.4. For each source sentence, every model outputs 7 hypothesis candidates and a total of 252 translations are collected for re-ranking. Then we compute 44 features for each hypothesis, out of which 11 features are finally used. The selected features are listed in Table 4. The kbmira algorithm takes these features to select the best hypothesis from these candidates. Note that the kbmira algorithm is optimized on `newstest19` and validated on `newstest20` to select the best feature combination. Instead of enumerating all the possible combinations (2^{44}), we incrementally add feature groups to kbmira algorithm for fast search.

It is considered as an ablation study to predefined features. After selecting the best feature combination, we further search better kbmira weights to achieve higher BLEU scores on `newstest20`.

6 Experiment

For our parallel translation system, we train three GLAT variants with the distilled data, and get the final outputs by reranking candidate hypothesis obtained from multiple GLAT models.

6.1 Hyperparameters

We implement our models with Fairseq (Ott et al., 2019). Our experiments are carried out on 4 machines with 8 NVIDIA V100 GPUs, each of which has 32 GB memory. The number of tokens per batch is set to $256k$. The dropout rate is set to 0.3 for the first $100k$ steps. We reduce the dropout to 0.1 after $100k$ steps, which can contribute to an improvement of about 1 BLEU score (Figure 3). The hyper-parameter λ for balancing L_{glm} and L_{aux} is set to 1.

6.2 Results

Our models are trained on the distilled parallel data and the distilled source monolingual data firstly. We experiment with various utilization of raw data, but the results show that the usage of raw data has no positive effect. The results of different architectures can be found in Table 5. Self-R and AT-R denote self-reranking and reranking with an autoregressive model, respectively. Experimental results show that the auxiliary decoder (AUX) effectively improves the performance by about 0.6 BLEU scores. For GLAT-base + CTC (Graves et al., 2006), we first set the max output length to twice the source input length and remove the blanks and repeated tokens after generation. We find CTC does not improve the performance and requires about twice the training time for convergence.

Based on GLAT with AUX, we employ three technologies to improve further: continuously training on the cycle KD data, domain adaptation, and reranking with various features. Table 6 shows the final results of our submitted system. Training on the distilled target monolingual data can further improve the performance by about 0.3 BLEU scores. Since the domain adaptation has already been em-

[5] https://github.com/moses-smt/mosesdecoder

ployed in the AT model's training process, the cycle KD data has already contained information of the in-domain data. However, the domain adaptation on GLAT can still gain a slight improvement of about 0.2. Moreover, an additional reranker with more diverse features can boost the performance by about 0.6.

7 Conclusion

In this paper, we introduced our system submitted to the WMT2021 shared news translation task on German→English. We build a parallel translation system based on the Glancing Transformer (Qian et al., 2020). Knowledge distillation, domain adaptation, reranking have proven effective in our system. Our constrained parallel translation system gets first place in the German→English translation task with a 35.0 BLEU score.

References

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

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.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, et al. 2020. Wechat neural machine translation systems for wmt20. *arXiv preprint arXiv:2010.00247*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2020. Glancing transformer for non-autoregressive neural machine translation. *ACL 2021*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

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.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

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.

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.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 1810–1822.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020. The volctrans machine translation system for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 305–312.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of ACL 2021*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

NVIDIA NeMo's Neural Machine Translation Systems for English ↔ German and English ↔ Russian News and Biomedical Tasks at WMT21

Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams, Oleksii Kuchaiev
NVIDIA
Santa Clara, CA
{sandeepsub, ohrinchuk, vadams, okuchaiev}@nvidia.com

Abstract

This paper provides an overview of NVIDIA NeMo's neural machine translation systems for the constrained data track of the WMT21 News and Biomedical Shared Translation Tasks. Our news task submissions for English ↔ German (En ↔ De) and English ↔ Russian (En ↔ Ru) are built on top of a baseline transformer-based sequence-to-sequence model (Vaswani et al., 2017). Specifically, we use a combination of 1) checkpoint averaging 2) model scaling 3) data augmentation with backtranslation and knowledge distillation from right-to-left factorized models 4) finetuning on test sets from previous years 5) model ensembling 6) shallow fusion decoding with transformer language models and 7) noisy channel re-ranking. Additionally, our biomedical task submission for English ↔ Russian uses a biomedically biased vocabulary and is trained from scratch on news task data, medically relevant text curated from the news task dataset, and biomedical data provided by the shared task. Our news system achieves a sacreBLEU score of 39.5 on the WMT'20 En → De test set outperforming the best submission from last year's task of 38.8. Our biomedical task Ru → En and En → Ru systems reach BLEU scores of 43.8 and 40.3 respectively on the WMT'20 Biomedical Task Test set, outperforming the previous year's best submissions.

1 Introduction

We take part in the WMT'21 News Shared Task for English ↔ German, English ↔ Russian, and the Biomedical Shared Task for English ↔ Russian. Our systems are implemented in the NVIDIA NeMo[1] framework (Kuchaiev et al., 2019). They build on baseline sequence-to-sequence transformer models (Vaswani et al., 2017) in the following ways: 1) Checkpoint averaging, 2) Model scaling up to 1B parameters, 3) Data augmentation with large-scale backtranslation (Edunov et al., 2018) of monolingual Newscrawl data and sequence-level knowledge distillation from a right-to-left factorized model (Zhang et al., 2019b), 4) Finetuning models on in-domain news data from WMT test sets made available in previous years, 5) Ensembling models trained on different subsets of the overall data 6) Shallow fusion decoding with transformer language models (Gulcehre et al., 2015) 7) Noisy channel re-ranking of beam search candidate hypotheses (Yee et al., 2019).

Overall, we find each of these components results in a small improvement in BLEU scores with backtranslation results being mixed depending on the language direction and whether the test data contains translationese inputs. Using a combination of these techniques, we achieve 39.5 sacreBLEU scores on the En → De WMT'20 test set, outperforming the best BLEU scores from last year's competition of 38.77.

Training our En ↔ Ru biomedical task submission from scratch using a biomedical vocabulary and similar model improvements to those used for our news task submission, we report a sacreBLEU score of 40.3 on En → Ru and 43.8 on Ru → Enh on the WMT'20 Biomedical Shared Task test dataset. This improves over the best submissions from last year's competition[2] of 39.6 and 43.3 on En → Ru and Ru → En respectively.

2 Datasets

We participated in the constrained data track at this year's news and biomedical competitions and used all the parallel corpora provided by the WMT Shared Tasks for both En ↔ De and En ↔ Ru. We used the provided English, German, and Russian monolingual Newscrawl data for backtranslation and training our autoregressive transformer language models. We filter out monolingual

[1] https://github.com/NVIDIA/NeMo

[2] We compare against all En ↔ Ru Biomedical submissions, not just the ones marked as the final submission.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 197–204
November 10–11, 2021.

Newscrawl data only based on minimum and maximum length criteria, but perform more aggressive filtering of our parallel data described in Section 2.1.

2.1 Parallel Corpus Filtering

We use a combination of the following data filtering steps for all parallel corpora (including pseudo parallel corpora generated via backtranslation and distillation) except for the Biomedical Shared Task provided data.

- **Language ID Filtering** - We use the fastText (Joulin et al., 2016) language ID classifier[3] to remove training examples that aren't in the appropriate language.
- **Length and Ratio Filtering** - We filter out examples where a sentence in either language is longer than 250 tokens before BPE tokenization and where the length ratio between source and target sentences exceeds 1.3.
- **Bicleaner** - Bitexts that were assigned a Bicleaner (Ramírez-Sánchez et al., 2020) score of < 0.6 were removed.

On the news shared task, we keep 60M parallel sentences for En $\leftrightarrow$ De and 26M sentences for En $\leftrightarrow$ Ru after filtering.

2.2 Biomedical Task Data

Our parallel biomedical domain data included a mix of all the En $\leftrightarrow$ Ru parallel training data given by shared task organizers and biomedically relevant examples selected from the provided En $\leftrightarrow$ Ru news task data.

We trained two biomedical domain binary classifiers, one for English and one for Russian. The classifiers were composed of two task-specific fully connected layers on top of pre-trained BERT Base (Devlin et al., 2018) or RuBERT Base (Kuratov and Arkhipov, 2019) for English and Russian respectively. The positive examples were sourced from the WMT'20 Biomedical Shared Task train set. The negative examples were randomly sampled from the parallel En $\leftrightarrow$ Ru news data given for the WMT'21 news task. An equal amount of 45K examples were used for both the positive and negative classes.

We ran our English biomedical domain classifiers on the English half of all approximately 26M parallel En $\leftrightarrow$ Ru WMT'21 news training data. We saved all sentences with predicted biomedical domain probabilities over 50%, collecting around 560k examples. We then ran our Russian classifier on the Russian counterparts to the 560k predicted in domain English sentences. We averaged the classifier scores from the English and Russian domain classifiers and used this average score as our final selection criteria. We set a cut-off threshold of .90 resulting in 208K parallel examples classified from the news domain data. We combined this with the 46k parallel biomedical examples provided for the task, resulting in a total of 256,037 parallel training examples.

2.3 Data Pre-processing and Post-processing

We normalize punctuation[4] and tokenize[5] examples with the Moses toolkit. For En $\leftrightarrow$ De, we train a shared BPE tokenizer with a vocab of 32k tokens using the YouTokenToMe[6] library. For En $\leftrightarrow$ Ru, we train language-specific BPE tokenizers with a vocab of 16k tokens each. For the En $\leftrightarrow$ Ru Biomedical translation task, we learn a separate BPE tokenizer solely on our Biomedical Task Data described in 2.2. We use BPE-dropout (Provilkov et al., 2019) of 0.1 for both language pairs and tasks. We post-process En $\rightarrow$ De model generated translations to replace quotes with their German equivalents - „ and “.

3 System overview

Our systems build on the Transformer sequence-to-sequence architecture (Vaswani et al., 2017). In the subsequent subsections, we discuss model scaling, checkpoint averaging, data augmentation with backtranslation and right-to-left distillation, model finetuning, ensembling, shallow fusion decoding with LMs, and noisy channel re-ranking.

3.1 Model Configurations

We experiment with three different model configurations - Large, XLarge, and XXLarge. The Large configuration corresponds to the "Transformer Large" variant from Vaswani et al. (2017)

[3] https://fasttext.cc/docs/en/language-identification.html

[4] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl

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

[6] https://github.com/VKCOM/YouTokenToMe

and the XLarge and XXLarge scale that base configuration along depth and width. The exact specifications are in Table 1. Following Kasai et al. (2020), we keep the number decoder layers fixed at 6 and scale only the depth of the encoder to 24 layers for the "XLarge" configuration. For stable optimization of deep transformers, we use the "pre-LN" transformer block (Xiong et al., 2020). When scaling to 1 billion parameters (XXLarge), we only increase hidden and feedforward dimensions of the model.

	Large	XLarge	XXLarge
Hidden Dim	1,024	1,024	1,536
Feedforward Dim	4,096	4,096	6,144
Attention Heads	16	16	24
Encoder Layers	6	24	24
Decoder Layers	6	6	6
Pre-LN	✗	✓	✓
Parameters	240M	500M	1B

Table 1: Model Configurations

3.2 Checkpoint Averaging

Over the course of training, we save the top-k checkpoints that obtain the best sacreBLEU scores on a validation set. The final model parameters are obtained by averaging the parameters corresponding to these checkpoints.

$$\theta_{avg} = \frac{1}{k}\sum_{i=1}^{k}\theta_i$$

θ_{avg} are the model parameters after checkpoint averaging and $\theta_1 \ldots \theta_k$ are the individual checkpoints being averaged. Empirically, we didn't observe a difference between averaging the last k checkpoints versus the top-k checkpoints. The former is however more common and implemented in libraries such as fairseq (Ott et al., 2019).

3.3 Data Augmentation with Backtranslation & Right-to-left model distillation

We follow Edunov et al. (2018) in backtranslating monolingual Newscrawl data with noise introduced via topk sampling (k=500). For En $\leftrightarrow$ De, we backtranslate ~250M sentences and filter translations based on the process described in Section 2.1. We observed fairly significant drops in BLEU score when using backtranslated data for En $\leftrightarrow$ Ru and did not apply any data augmentation for this language pair. We use the XLarge model configuration trained only on the News Task provided parallel corpus to generate translations.

We also train an XLarge model for En $\rightarrow$ De and De $\rightarrow$ En on the News Task provided parallel data where the output sequence is factorized from right-to-left. Translations of the training dataset with topk sampling (k=500) using these models are generated and added to the overall training set.

When adding only backtranslated text or data generated from right-to-left factorized models, we use a 2:1 ratio of parallel to pseudo-parallel (model generated) data. When training with a combination of both, we use a 6:3:1 ratio of parallel, right-to-left generated, and backtranslated data. We skew data sampling in this way since training on right-to-left generated data showed better performance on recent WMT test sets as opposed to backtranslation which did better on old test sets that contained translationese inputs (see Tables 2 and 3).

3.4 Mixed Domain Training

For the biomedical task submission, we experiment with different mixed domain training approaches (Zhang et al., 2019a). We train on the concatenated combination of news task and biomedical task data-up-sampling the proportion of biomedical data to make up 30% or 50% of the data-parallel examples seen during training. We also train models on concatenated data with no up-sampling and with purely news task data. The base models trained on exclusively news task data still use the biomedical vocabulary tokenizer.

3.5 Model Finetuning

For our news task submission, we finetuned models on an in-domain parallel corpus consisting of WMT provided test datasets from past years (WMT'08 - WMT'19 for En $\leftrightarrow$ De comprising ~32k examples) for both En $\leftrightarrow$ De and En $\leftrightarrow$ Ru.

We finetuned our biomedical task base models on the 250k parallel sentences obtained via the process described in Section 2.2. Models are finetuned for 1-2 epochs using a fixed tuned learning rate and the top-k checkpoints on a validation dataset (newstest2020 for the News Shared Task) are averaged.

3.6 Ensembling

Given k different models for a particular language direction trained with the same tokenizer, we ensemble them at inference by averaging their probability distributions over the next token.

$$P(y_t|y_{<t}, x; \theta_1 \dots \theta_k) = \frac{1}{k} \sum_{i=1}^{k} P(y_t|y_{<t}, x; \theta_i)$$

Where $P(y_t|y_{<t}, x)$ is the probability distribution over the target token y_t given all previously generated target tokens $y_{<t}$ and the input sequence x. $\theta_1 \dots \theta_k$ are the k different models being ensembled.

Beam search scores are computed using these averaged probabilities at each time step. In practice, we ensemble models trained on different subsets of the available data.

For En $\leftrightarrow$ De, we ensemble a total of 6 models trained on different subsets of the data. Example: News Task provided bitext only, the addition of backtranslated and/or data from right-to-left factorized models and finetuned models.

For En $\leftrightarrow$ Ru, we ensemble a total of 3 identical XLarge models trained with different random seeds on the main parallel corpus.

For the En $\leftrightarrow$ Ru biomedical task, we ensemble 4 finetuned models whose base configurations were trained with different mixed domain sampling ratios. Specifically, each translation direction includes an ensemble of models initially trained on mixed domain data with 50% up-sampling of biomedical data, concatenated biomedical and news data with no up sampling, exclusively news data, and exclusively news data with right-to-left distillation.

3.7 Shallow Fusion Decoding with Language Models

Aside from backtranslation, another way to leverage large amounts of monolingual data is via training language models. We train language-specific 16-layer transformer language models at the sentence level, which is architecturally similar to Radford et al. (2019). They are trained on Newscrawl and use the same tokenizers as our NMT systems.

When generating translations, we decode jointly with our NMT system $\theta_{s\to t}$ and a target side language moel θ_t (Gulcehre et al., 2015). The score of a partially decoded sequence on the beam $\mathcal{S}(y_{1\dots n})$ of length n is given by the following recurrence

$$\mathcal{S}(y_{1\dots n}|x; \theta_{s\to t}, \theta_t) = \mathcal{S}(y_{1\dots n-1}|x; \theta_{s\to t}, \theta_t) + \log P(y_n|y_{<n}, x; \theta_{s\to t}) + \lambda_{sf} \log P(y_n|y_{<n}; \theta_t)$$

where the empty sequence has a score of 0. We tuned the LM importance coefficient λ_{sf} on a validation dataset and found a value between 0.05 - 0.1 to work well in practice.

3.8 Noisy Channel Re-ranking

We re-rank the beam search candidates produced by our ensemble model generated with or without shallow fusion using a neural noisy channel model (Yee et al., 2019). The noisy channel model computes the score of any translation $\mathcal{S}(y_i|x)$ on the beam based on a forward (source-to-target) model, a reverse (target-to-source), and a target language model. The best translation after re-ranking is given by

$$\arg\max_i \mathcal{S}(y_i|x) = \log P(y_i|x; \theta^{ens}_{s\to t}) + \lambda_{ncr}\big(\log P(x|y_i; \theta_{t\to s}) + \log P(y_i; \theta_t)\big)$$

Forward log probabilities are given by an ensemble of source-to-target models $\theta^{ens}_{s\to t}$. We experimented with using an ensemble of target-to-source translation models to compute $\log P(x|y_i)$ but didn't observe any empirical benefits and so all reported results use only a single reverse model $\theta_{t\to s}$ for noisy channel re-ranking. We generate 15 candidates via beam search and tune λ_{ncr} on a validation dataset and found a value between 0.5 - 0.7 to work well in practice.

3.9 Training & Optimization

All En $\leftrightarrow$ De models were trained for up to 450k updates using the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.98$ and Inverse Square Root Annealing (Vaswani et al., 2017) with 30k warm-up steps and a maximum learning rate of 4e-4. En $\leftrightarrow$ Ru models were trained for up to 150k updates with 7k warmup steps. We use label smoothing of 0.1 and a dropout of 0.1 on intermediate activations including attention scores to regularize our models.

The "Large" models were trained on NVIDIA DGX-1 machines with 8 32G V100 GPUs. We use a batch size of 16k tokens per GPU for an effective batch size of 128k tokens. The "XLarge" models were trained on 64 GPUs split across 4 NVIDIA DGX-2 nodes with 16 32G V100 GPUs each. These models use an effective batch size of 256k tokens. Finally, our "XXLarge" models were trained on 256 GPUs across 16 DGX-2 nodes with an effective batch size of 512k tokens.

	En → De News Task Model	WMT'14	WMT'18	WMT'19	WMT'20	Avg Δ
(1)	Transformer-Large	29.9	46.6	41.1	31.5	0
(2)	(1) + Checkpoint Averaging	30.7	48.3	43.5	33.5	1.4
(3)	(2) + Transformer-XLarge	32.2	48.7	43.3	34.7	2.1
(4)	(3) + Backtranslation	34.9	49.2	40.5	34.6	2.2
(5)	(3) + R2L Distllation	32.4	49.1	43.4	37.2*	2.9
(6)	(3) + Backtranslation + R2L Distllation	34.3	50.1	42.9	37.4*	3.6
(7)	(5) + Shallow Fuison Decoding	32.8	49.0	43.4	37.6*	3.1
(8)	(6) + Transformer-XXLarge	35.5	50.0	41.8	37.5*	3.6
(9)	(6) + Finetuning (WMT'08-19)	-	-	-	37.6*	-
(10)	(8) + (9) + Ensembling	34.4	50.7	44.2	38.9*	4.4
(11)	(10) + Noisy Channel Re-ranking	**36.0**	**51.6**	**44.3**	**39.5***	5.2

Table 2: Model ablations for En → De. All reported scores are obtained from sacreBLEU. WMT'20 scores with a * apply post-processing to replace punctuations as reported in Section 2.3. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

	De → En News Task Model	WMT'14	WMT'18	WMT'19	WMT'20	Avg Δ
(1)	Transformer-Large	35.5	45.0	40.5	37.5	0
(2)	(1) + Checkpoint Averaging	36.5	46.1	41.6	38.3	0.7
(3)	(2) + Transformer-XLarge	37.7	47.8	41.9	37.6	1.3
(4)	(3) + Backtranslation	40.3	50.4	40.5	37.7	2.3
(5)	(3) + R2L Distllation	37.5	47.8	42.3	39.7	1.9
(6)	(3) + Backtranslation + R2L Distllation	39.3	49.6	41.8	39.4	2.7
(7)	(6) + Finetuning (WMT'08-19)	-	-	-	41.1	-
(8)	(7) + Ensembling	39.5	49.9	**43.3**	41.9	3.7
(9)	(8) + Noisy Channel Re-ranking	**40.1**	**50.6**	42.8	**42.0**	4.0

Table 3: Model ablations for De → En. All reported scores are obtained from sacreBLEU. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

	En → Ru News Task Model	WMT'17	WMT'18	WMT'19	WMT'20	Avg Δ
(1)	Transformer-Large	35.4	30.8	32.0	22.3	0
(2)	(1) + Transformer-XLarge + Ckpt Avg	36.8	32.2	33.2	23.2	1.2
(3)	(2) + Finetuning (WMT'14-16)	38.0	33.1	35.1	24.3	2.5
(4)	(3) + Ensemble (x3)	**38.6**	33.5	35.3	**24.8**	2.9
(5)	(4) + Shallow Fusion	**38.6**	**33.7**	**35.7**	24.7	3.0
(6)	Oracle BLEU with beam size 4	-	-	39.9	-	-

Table 4: Model ablations for En → Ru. All reported scores are obtained from sacreBLEU. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

4 News Task Submission

In this Section, we present results for our News Shared Task submission. Tables 2 and 3 contain ablations for En ↔ De and while Tables 4 and 5 has ablations for En ↔ Ru.

Each of the components we describe improves BLEU scores except for backtranslation and scaling our models to 1B params. Both show mixed results on En ↔ De - scores improve significantly on WMT'14 and WMT'18 test sets when adding backtranslated data (possibly because these test sets contain translationese inputs) but hurts or does not improve performance on WMT'19 and WMT'20 test sets. Our 1B parameter model does significantly better on WMT'14, but worse on WMT'19 and is comparable to the 500M parameter model on WMT'18 and WMT'20. We found optimization

	Ru → En News Task Model	WMT'17	WMT'18	WMT'19	WMT'20	Avg Δ
(1)	Transformer-Large	37.6	33.0	37.7	36.6	0
(2)	(1) + Transformer-XLarge + Ckpt Avg	38.7	34.3	38.2	37.2	0.9
(3)	(2) + Finetuning (WMT'14-16)	40.7	35.4	40.5	37.1	2.2
(4)	(3) + Ensemble (x3)	40.7	35.5	**41**	**37.7**	2.5
(5)	(4) + Shallow Fusion	**40.9**	**35.9**	40.8	37.5	2.6
(6)	Oracle BLEU with beam size 4	-	-	46.4	-	-

Table 5: Model ablations for Ru → En. All reported scores are obtained from sacreBLEU. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

	En → Ru Biomedical Task Model	WMT'20 Bio	Δ
(1)	Transformer-Large News Task Model	32.2	0
(2)	Transformer-XLarge News Task Model	33.8	1.6
(3)	Transformer-XLarge + Biomed Vocab w/ News Data	33.9	1.7
(4)	Transformer-XLarge + Biomed Vocab w/ News + R2L Distillation Data	34.2	2.0
(5)	Transformer-XLarge + Biomed Vocab w/ News + 30% Biomed Data	36.7	4.5
(6)	Transformer-XLarge + Biomed Vocab w/ News + 50% Biomed Data	36.8	4.6
(7)	Transformer-XLarge + Biomed Vocab w/ News + Biomed Data	37.4	5.2
(8)	(2) + Biomed Data Finetuning	37.8	5.6
(9)	(3) + Biomed Data Finetuning	38.5	6.3
(10)	(4) + Biomed Data Finetuning	38.2	6.0
(11)	(6) + Biomed Data Finetuning	37.4	5.2
(12)	(7) + Biomed Data Finetuning	38.5	6.3
(13)	(9) (10) (11) (12) Ensemble	39.9	7.7
(14)	(13) + Shallow Fusion	40.0	7.8
(15)	(13) + Noisy Channel Re-ranking	**40.3**	8.1

Table 6: Model iterations for the Biomedical Shared Task En → Ru. All reported scores are checkpoint averaged and are obtained from sacreBLEU. Δ computes the improvement over the Transformer-Large baseline on the WMT'20 Biomedical Shared Task test set. Model 15 is our selected best submission and model 14 is our alternate submission.

with Adam to be unstable and used AdamW with a weight decay of 0.01 instead. Our final En → De model achieves a BLEU score of 39.5 on the WMT'20 test set, which improves over the submission with the best BLEU score from last year's competition of 38.8. We however do not do as well on De → En, with a final BLEU score of 42, compared to last year's best submission of 43.8.

Backtranslation significantly hurt performance in initial experiments on En ↔ Ru so we exclude it from our ensemble. The impact of ensembling, finetuning, and shallow fusion are fairly similar to En ↔ De. Additionally, we also report an "Oracle BLEU" score in Tables 4 and 5 where we compute BLEU scores by cheating and picking the translation on our beam that has the highest sentence BLEU score with respect to the reference. This is a useful indicator of how much there is to gain by re-ranking the beam search candidates.

5 Biomedical translation task submission

We present our Biomedical Shared Task submission in this section. Building on lessons learned from our news task ablation studies, we opted to use the Transformer-XLarge architecture, and average all of the intermediate model checkpoints which helps reduce model variance as a consequence of finetuning. Tables 6 and 7 show our results as we iterated on improving our models.

We trained our BPE tokenizer on biomedical data to mitigate character-level segmentation of words unique to the biomedical domain. We found this had a minimal effect on model performance. This could be because the majority of our parallel biomedical data was selected from news task training data, and thus biomedical words were already adequately accounted for by the news task model's tokenizer. We found that up-sampling in-domain

	Ru → En Biomedical Task Model	WMT'20 Bio	Δ
(1)	Transformer-Large News Task Model	38.7	0
(2)	Transformer-XLarge News Task Model	39.8	1.1
(3)	Transformer-XLarge + Biomed Vocab w/ News Data	39.8	1.1
(4)	Transformer-XLarge + Biomed Vocab w/ News + R2L Distillation Data	39.2	0.5
(5)	Transformer-XLarge + Biomed Vocab w/ News + 30% Biomed Data	37.6	-1.1
(6)	Transformer-XLarge + Biomed Vocab w/ News + 50% Biomed Data	38.4	-0.3
(7)	Transformer-XLarge + Biomed Vocab w/ News + Biomed Data	41.5	2.8
(8)	(2) + Biomed Data Finetuning	42.3	3.6
(9)	(3) + Biomed Data Finetuning	42.6	3.9
(10)	(4) + Biomed Data Finetuning	41.7	3.0
(11)	(6) + Biomed Data Finetuning	39.6	0.9
(12)	(7) + Biomed Data Finetuning	41.8	3.1
(13)	(9) (12) Ensemble	42.8	4.1
(14)	(9) (10) (11) (12) Ensemble	**43.8**	5.1
(15)	(14) + Shallow Fusion	43.7	5.0
(16)	(14) + Noisy Channel Re-ranking	42.1	3.4

Table 7: Model iterations for the Biomedical Shared Task Ru → En. All reported scores are checkpoint averaged and are obtained from sacreBLEU. Δ computes the improvement over the Transformer-Large baseline on the WMT'20 Biomedical Shared Task test set. Model 14 is our selected best submission.

biomedical data hurt performance compared to concatenating out-of-domain and in-domain data with no up-sampling. For the En → Ru direction, including any biomedical domain data during initial model training showed improvements over training on exclusively news task data. Up-sampling in-domain data for the Ru → En direction hurt performance compared to our news task model baselines.

Unsurprisingly, finetuning base models on biomedical domain data improved BLEU scores for all models. In-domain finetuning helped models initially trained on news task data overcome performance gaps between themselves and models that had seen a higher amount of biomedical data during training. Neither shallow fusion nor noisy channel re-ranking improved model performance after ensembling for the Ru → En direction. Both techniques individually improved En → Ru performance but failed to do so in combination.

Ensembling our models led to an additional performance boost and allowed us to reach our maximum En → Ru BLEU score of 40.3 and Ru → En BLEU score of 43.8. These scores show a 0.9 and 0.5 improvement over last year's best score of 39.4 and 43.3 (Bawden et al., 2020) respectively.

6 Conclusion

We present Neural Machine Translation Systems for the En ↔ De News Task and En ↔ Ru News and Biomedical shared tasks implemented in the NeMo framework (Kuchaiev et al., 2019). Our systems build on the Transformer sequence-to-sequence model to include backtranslated text and data from right-to-left factorized models, ensembling, finetuning, mining biomedically relevant data using domain classifiers, shallow fusion with LMs, and noisy channel re-ranking. These achieve competitive performance to submissions from previous years.

7 Author Contributions

Sandeep Subramanian: Sandeep implemented and experimented with model scaling, backtranslation, distillation with right-to-left factorized models, model ensembling, and noisy channel re-ranking. He also ran all of the En ↔ De News Shared Task experiments, the right-to-left factorized models for the En ↔ Ru Biomedical task, and wrote parts of the paper.

Oleksii Hrinchuk: Oleksii H implemented the shallow fusion approach and helped with writing backtranslation scripts. He also ran all of the En ↔ Ru News Shared Task experiments and trained the language models used in the Biomedical experiments.

Virginia Adams: Virginia implemented and experimented with the warm-start biomegatron en-

coder, biomedical baselines, classifiers to extract biomedically relevant monolingual and parallel corpora, mixed domain, and finetuning of News models. She ran all of the En ↔ Ru Biomedical experiments.

Oleksii Kuchaiev: Oleksii K advised and managed the project.

8 Acknowledgements

The authors would like to thank Mike Chrzanowski, Ryan Prenger, Eric Harper, Micha Livne, Abhinav Khattar, Anton Peganov, Mohammad Shoeybi, Somshubra Majumdar and Fei Jia for many useful discussions over the course of this project.

References

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Iñigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez De Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 Biomedical Translation Shared Task: Basque, Italian and Russian as New Additional Languages. In *5th Conference on Machine Translation*, Online, Unknown Region.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

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*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A Smith. 2020. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. *arXiv preprint arXiv:2006.10369*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Banón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298.

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*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.

Kyra Yee, Nathan Ng, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv preprint arXiv:1908.05731*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019a. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019b. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.

Facebook AI's WMT21 News Translation Task Submission

Chau Tran Shruti Bhosale James Cross Philipp Koehn Sergey Edunov Angela Fan
Facebook AI
chau,shru,jcross,pkoehn,edunov,angelafan@fb.com

Abstract

We describe Facebook's multilingual model submission to the WMT2021 shared task on news translation. We participate in 14 language directions: English to and from Czech, German, Hausa, Icelandic, Japanese, Russian, and Chinese. To develop systems covering all these directions, we focus on *multilingual* models. We utilize data from all available sources — WMT, large-scale data mining, and in-domain backtranslation — to create high quality bilingual and multilingual baselines. Subsequently, we investigate strategies for scaling multilingual model size, such that one system has sufficient capacity for high quality representations of all eight languages. Our final submission is an ensemble of dense and sparse Mixture-of-Expert multilingual translation models, followed by finetuning on in-domain news data and noisy channel reranking. Compared to previous year's winning submissions, our multilingual system improved the translation quality on all language directions, with an average improvement of 2.0 BLEU. In the WMT2021 task, our system ranks first in 10 directions based on automatic evaluation.

1 Introduction

We participate in the WMT2021 shared task on news translation and submit a multilingual translation system. In recent years, multilingual translation has gained significant interest as an alternative to developing separate, specialized systems for different language directions (Firat et al., 2016; Tan et al., 2019; Aharoni et al., 2019; Zhang et al., 2020; Tang et al., 2020; Arivazhagan et al., 2019). Multilingual systems have great potential for simplicity and consolidation, making them attractive options for the development and maintenance of commercial translation technologies. From a research standpoint, studies of transfer learning between related languages and developing methods that incorporate low-resource languages are strong motivators for grouping languages together in one system (Dabre et al., 2019; Fan et al., 2021).

Despite such motivations, existing multilingual translation systems have been unable to show that the translation quality of multilingual systems surpasses that of bilingual. Several works compare to bilingual baselines, but these baselines do not incorporate standard techniques used across the field — such as backtranslation or dense model scaling. Further, multilingual translation systems are often developed on non-standard training datasets and use different evaluation datasets. These factors make it difficult to assess the performance of multilingual translation, particularly when compared to the most competitive bilingual models.

In this work, our aim is to demonstrate against the winning WMT2020 models and our bilingual WMT2021 systems that multilingual translation models have stronger performance than bilingual ones. We focus on 14 language directions: English to and from Czech, German, Hausa, Icelandic, Japanese, Russian, and Chinese. We create an unconstrained system that utilizes both WMT distributed and publicly available training data, apply large-scale backtranslation, and explore dense and mixture-of-expert architectures. We compare the impact of various techniques on bilingual and multilingual systems, demonstrating where multilingual systems have an advantage. Our final multilingual submission improves the translation quality on average +2.0 compared to the WMT2020 winning models, and ranks first in 10 directions based on automatic evaluation on the WMT2021 leaderboard.

2 Data

We participate in translation of English to and from Czech (cs), German (de), Hausa (ha), Icelandic (is), Japanese (ja), Russian (ru), and Chinese (zh). We describe our bitext and monolingual data sources, including additional mined data created for Hausa, and our preprocessing pipeline.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 205–215

2.1 Bitext Data

For all directions, we use all available bitext data from the shared task . For language directions such as English to German or English to Russian, this provides millions of high-quality bitext. However, for low to mid resource languages, such as Hausa and Icelandic, we incorporate additional sources of data from freely available online sources such as ccMatrix (Schwenk et al., 2019), ccAligned (El-Kishky et al., 2020), and OPUS (Tiedemann, 2012). We utilize all available data sources to develop the best quality translation model possible.

For English-Hausa (and Hausa-English), we also mined extra parallel data from the provided monolingual data. We use LaBSE (Feng et al., 2020) to embed Hausa and English sentences into the same embedding space. We then use the margin function formulation (Artetxe and Schwenk, 2019) based on K-nearest neighbors (KNN) to score and rank pairs of sentences from the two languages. Using the mining strategy from Tran et al. (2020), we mined an additional one million pairs of parallel sentences for English-Hausa.

Data Processing. The majority of available bitext represents noisy alignment rather than the output of human translations. We apply several steps of preprocessing to filter noisy data. First, we apply language identification using `fasttext` (Joulin et al., 2017) and retain sentences predicted as the desired language[1]. We then normalize punctuation with `moses`. Subsequently, we removed sentences longer than 250 words and with a source/target length ratio exceeding 3.

2.2 Monolingual Data

Previous work (Ng et al., 2019) shows that using in-domain monolingual data provides the most quality improvement when used for large-scale backtranslation. For high resource languages such as English and German, there are sufficiently large quantities of in-domain data available in Newscrawl, and we do not utilize additional monolingual data. For the remaining languages, the data available in Newscrawl is not sufficient and we follow the strategy in Moore and Lewis (2010); Ng et al. (2019) to examine large quantities of general-domain monolingual data from Commoncrawl[2] and identify a subset that is most similar to the available in-domain news data. For each language, we train an n-gram language model (Heafield, 2011) on all available news-domain data (Newscrawl) and a n-gram language model on a similarly sized sample from general-domain data (Commoncrawl). For each sentence s in Commoncrawl, we compute word-normalized cross entropy scores $H_{\text{news}}(s)$ and $H_{\text{general}}(s)$ using in-domain language model and general-domain language model respectively. We retain sentences that meet the threshold $H_{\text{news}}(s) - H_{\text{general}}(s) > 0.01$. This selects around 5% of total number of sentences in the original Commoncrawl.

Language	Bitext	Monolingual
Czech	185M	140M
German	571M	237M
Hausa	1.7M	7M
Icelandic	28.2M	101M
Japanese	145.7M	218M
Russian	297M	163M
Chinese	166M	123M
English	—	430M

Table 1: **Amount of Data per Language**. The bitext data includes data distributed by the WMT Shared Task, the OPUS repository, ccMatrix, ccAligned, and newly mined data for Hausa. The monolingual data includes data distributed by the WMT Shared Task and CC100.

2.3 Vocabulary

To create our multilingual vocabulary, we first learn a multilingual subword tokenizer on our combined training data across all languages. We use SentencePiece (Kudo and Richardson, 2018), which learns subword units from untokenized text. We train our SPM model with temperature upsampling (with T=5) similar to Conneau et al. (2020), so that low-resource languages are represented. For bilingual models, we used vocabulary size of $32,000$, and for multilingual models, we used $128,000$. Subsequently, we convert the learned SPM units into our final vocabulary.

3 System Overview

We describe step-by-step how we created our final multilingual submission for WMT2021. We detail our bilingual and multilingual model architectures, as well as how we incorporate strategies such as backtranslation, news-domain finetuning, ensembling, and noisy channel reranking.

[1]Note: for Hausa, the language identification system was unreliable, so we did not utilize it.

[2]http://data.statmt.org/cc-100/

3.1 Baseline Bilingual Models

A pre-requisite to creating state-of-the-art multilingual translation systems is establishing strong, competitive bilingual baselines. Our goal is to apply the same set of techniques in data augmentation and modeling scaling to both bilingual and multilingual models, and demonstrate multilingual models have stronger translation quality.

To create baseline bilingual systems, we train a separate Transformer model (Vaswani et al., 2017) for each language direction. For every language pair except Hausa, we use the Transformer 12/12 configurations in Table 2. For Hausa-English (and English-Hausa), since the amount of bitext data is smaller, we use the Transformer-Base architecture similar to Vaswani et al. (2017). We train all our models using `fairseq` (Ott et al., 2019) on 32 Volta 32GB GPUs. We use learning rate of 0.001 with the Adam optimizer, batch size of 768,000 tokens[3], and tune the dropout rate for each language direction independently. For large models

3.2 Backtranslation

Backtranslation (Sennrich et al., 2015) is a widely used technique to improve the quality of machine translation systems using data augmentation. To perform backtranslation for a *forward* language direction (e.g. English to German), we use a system in the backward direction (e.g. German to English), to translate the target German monolingual data into the English source. We then use these *backtranslated* synthetic English to German sentence pairs in conjunction with the original parallel data to train an improved forward translation model.

We use all available filtered monolingual data we have for each language (up to 500 million sentences per language) for backtranslation. Using our baseline bilingual models (described in Section 3.1), we first finetune on in-domain news data (described in Section 3.5), and use an ensemble of 3 models with different seeds to generate backtranslation data using beam search. For Hausa-English and English-Hausa, we applied a round of iterative backtranslation (Hoang et al., 2018; Chen et al., 2019) as the quality improvement is significant.

3.3 Data Sharding and Sampling

Table 1 displays the amount of data for all languages after postprocessing. We divide the data into multiple shards, with each training epoch using one shard. We downsample data from both high resource directions and synthetic backtranslated data by dividing them into a greater number of shards than the real bitext data from low resource directions. We find that downsampling high resource languages works better than upsampling low resource languages, as upsampling contributes more strongly to overfitting.

[3]6000 tokens per GPU * 32 GPUs * 4 update frequency

	12/12	24/24	24/24 Wide
Layers	12	24	24
Emb. Size	1,024	1,024	2,048
FFN Size	4,096	8,192	16,384
Attn. Heads	16	16	32
Total Params.	480M	1.2B	4.7B

Table 2: **Dense Transformer Configurations**.

3.4 Model Architectures

We describe several model architectures that we compared using the final dataset with both bitext and backtranslated data.

Scaling Bilingual Models. Based on the baseline architectures described in Section 3.1, we further improve our bilingual models. The two main improvements are: adding backtranslated data, and adding deeper and wider Transformer configurations to take advantage of the increase in data.

Dense Multilingual Models. For the multilingual systems, we train two separate models: *Many to English*, or one system encompassing every language translated into English, and *English to Many*, or one for English into every language. The challenge of multilingual models is often one of capacity — given a fixed number of parameters, a model needs to learn representations of numerous languages rather than just one. To understand the needed capacity and optimal architectural configuration, we experiment with different Transformer architectures, ranging from 480M parameters to 4.7B parameters (see Table 2).

Sparsely Gated MoE Multilingual Models. In multilingual models, languages necessarily compete for capacity and must balance sharing parameters with specialization for different languages. A straightforward way to add capacity to neural architectures is to simply scale the model size in a *dense* manner: increasing the number of layers, the width of the layers, or the size of the hidden dimension.

However, this has a significant computational cost, as each forward pass activates all parameters — at the limit, models become incredibly slow to train and produce translations (Fan et al., 2021).

In this work, we instead focus on *sparse* model scaling, motivated by wanting to increase capacity without a proportional increase in computational cost. We train Sparsely Gated Mixture-of-Expert (MoE) models (Lepikhin et al., 2020) for *English to Many* and *Many to English*. These models aim to strike a balance between allowing high-resource directions to benefit from increased expert model capacity, while also allowing transfer to low-resource directions via shared model capacity. In each Sparsely Gated MoE layer, each token is routed to the top-k expert FFN blocks based on a learned gating function. Thus, only a subset of all the model's parameters is used per input sequence.

We use a Transformer architecture with the Feed Forward block in every alternate Transformer layer replaced with a Sparsely Gated Mixture-of-Experts layer with top-2 gating in the encoder and decoder. As in Lepikhin et al. (2020), we also add a gate loss term to balance expert assignment across tokens with a gate loss weight of 0.01. We use an expert capacity factor of 2.0. We use a learning rate of 0.001 with the Adam optimizer with 4000 warmup updates and a batch size of 1 Million tokens (MoE model with 64 experts) or 1.5 Million tokens (MoE model with 128 experts).

3.5 In-Domain Finetuning

Finetuning with domain-specific data is an effective method of improving translation quality for the desired domain, and thus we curated news-domain data for finetuning. For directions such as German and Russian, we finetune on evaluation datasets from previous years of WMT. For Hausa and Icelandic, as no previous data exists, we use mined data and filter to the subset identified as most likely news domain. Subsequently, we finetune our models on the in-domain data for a maximum of ten epochs, selecting the best model with validation loss on the `newstest2020` dev set. For our submission, we use the settings tuned on `newstest2020` and include `newstest2021` dev set in the final finetuning.

3.6 Checkpoint Averaging

To combat bias toward recent training data, it is common to average parameters across multiple checkpoints of a model (Vaswani et al., 2017). We apply this technique to all models and average the last five checkpoints. To address rapid overfitting during finetuning, we also average the finetuned model with the model after the initial training is complete and select this averaged set of parameters if it performs better on the validation data.

3.7 Noisy Channel Re-ranking

We apply noisy channel re-ranking to select the best candidate translations from n-best hypotheses generated with beam search. We follow Yee et al. (2019); Bhosale et al. (2020) and utilize scores from the direct model $P(tgt|src)$, channel model $P(src|tgt)$, and language model $P(tgt)$. To combine these scores for reranking, for every one of our n-best hypotheses, we calculate:

$$\log P(tgt|src) + \lambda_1 \log P(src|tgt) + \lambda_2 \log P(tgt)$$

The weights λ_1 and λ_2 are determined by tuning them with a random search over 1000 trials on a validation set and selecting the weights that give the best performance. In addition, we also tune a length penalty. The search bounds we use for the weights and the length penalty are [0,2].

Language Models. We trained Transformer-based language models for all languages on the same monolingual data as used for backtranslation. The exception is English, where we trained on the CC100 English data and RoBERTa training data (Conneau et al., 2020; Wenzek et al., 2019; Liu et al., 2019). For the high resource languages, the language models have 12 decoder layers and embedding dimension 4096. For Hausa and Icelandic, we trained smaller language models with 6 decoder layers to prevent overfitting.

3.8 Post-Processing

As a final step, we apply post-processing to the translation outputs for Czech, German, Icelandic, Japanese, and Chinese. For Czech, German, and Icelandic, we convert quotation marks to German double-quote style[4]. For Chinese and Japanese, we convert punctuation marks to the language-specific punctuation characters.

4 Experiments and Results

We conduct experiments to quantify the impact of each of the component in our system. All experiments are evaluated on `newstest20` (Barrault et al., 2020) using `SacreBLEU` (Post, 2018).

[4]https://en.wikipedia.org/wiki/Quotation_mark#German

	cs-en	de-en	ha-en	is-en	ja-en	ru-en	zh-en
Multilingual Vocab	28.3	**38.0**	28.3	34.5	21.1	**38.0**	**30.8**
Bilingual Vocab	**28.6**	36.8	**28.4**	**35.2**	**22.4**	37.0	29.6
	en-cs	**en-de**	**en-ha**	**en-is**	**en-ja**	**en-ru**	**en-zh**
Multilingual Vocab	33.2	39.4	23.1	**29.4**	**26.1**	25.7	42.4
Bilingual Vocab	**33.7**	**39.8**	**23.9**	**29.4**	**26.1**	**26.0**	**43.3**

Table 3: **Impact of Vocabulary on Bilingual Models**. We compare using a specialized bilingual vocabulary vs. a general multilingual vocabulary and its impact on performance of bilingual systems.

4.1 Creating State-of-the-Art Multilingual Translation Models

We investigate the effectiveness of multilinguality in translation. Compared to bilingual models, which can dedicate their capacity to specializing in specific source and target languages, multilingual systems must learn to effectively share available capacity across all languages while balancing languages of different resource levels. Despite rising research interest, previous WMT submissions have not demonstrated quality improvement of multilingual models over bilingual models. We discuss various choices and comparisons that build our state-of-the-art multilingual translation system. Overall, the best multilingual systems outperform the best bilingual ones in 11 out of 14 directions, with an average improvement of +0.8 BLEU.

4.1.1 Building a Multilingual Vocabulary.

Similar to how multilingual systems must share model capacity, multilingual translation models must also share *vocabulary capacity*. Instead of training specialized subword units for a specific language (often 32k), multilingual models group all languages together to learn a much smaller vocabulary set than 32k * number of languages. We first examine the impact of this multilingual vocabulary, by taking a bilingual system and training it with the multilingual vocabulary. This would indicate a performance difference coming not from architecture, but from the vocabulary itself. Table 3 indicates that across all directions, using a specialized bilingual vocabulary is usually superior, meaning multilingual systems must bridge the performance gap of a potentially subpar vocabulary. However, for some directions such as en-is and en-ja, no difference is observed.

4.1.2 Comparing Model Architectures.

Dense Transformer Models. Overall, we find that dense multilingual models are fairly competitive with dense bilingual models (see Table 4). Importantly, we find multilingual models benefit greatly from additional model capacity. In Table 5, we show comparable dense scaling applied to a bilingual model translating from English to German. While the multilingual model improves up to 1 BLEU point, the bilingual model only improves +0.3 BLEU, indicating diminishing return and possible overfitting in bilingual models. Scaling multilingual translation models has stronger potential for performance improvement.

Sparsely Gated Mixture of Expert Models. If multilingual models benefit from greater capacity, what is the best way to add that capacity? In Table 4, we compare the performance of Dense and MoE multilingual models while keeping the FLOPs per update approximately the same for fair comparison. Due to the conditional compute capacity of MoE layers, MoE models have a greater number of total parameters, but a comparable computational cost with the corresponding dense model.

For Many to English and English to Many, the MoE model with 64 experts per MoE layer gives an average boost of +0.7 BLEU on the dev set. To compare to scaling dense models, increasing dense model size from 12/12 to 24/24 does not correspond to significant improvement for Many to English. However, there is around +1 BLEU improvement in dense scaling on English to Many. We also see a slightly decline or no improvement in the performance of MoE models (MoE-64 12/12 vs MoE-128 24/24) when increasing model dimensionality and increasing the number of experts from 64 to 128. One possible hypothesis is that having 128 experts is largely unnecessary for only 7 languages. Compared to 64 experts, training convergence per expert is slower as each expert is exposed to fewer tokens during training on an average.

After finetuning on in-domain data, we observe a significant improvement in performance across the board. There is a larger improvement from finetuning in MoE models compared to the associated

	cs-en	de-en	ha-en	is-en	ja-en	ru-en	zh-en	Avg
Bilingual Dense 12/12	28.3	38.0	**28.3**	34.5	21.1	38.0	**30.8**	**31.3**
Dense 12/12	26.9	37.5	**28.3**	35.2	19.0	36.2	28.8	30.3
MoE-64 12/12	28.0	**38.9**	27.2	**37.3**	18.5	**39.1**	28.0	31.0
Dense 24/24	28.1	37.2	26.3	35.6	20.6	35.8	28.0	30.2
MoE-128 24/24	28.1	36.8	23.1	36.9	18.7	36.9	29.7	29.7
Dense 24/24 Wide	**29.0**	37.9	24.5	36.8	**21.2**	36.9	30.4	31.0
Bilingual Dense 12/12, BL-FT	30.4	42.8	30.3	35.5	24.6	39.5	36.2	34.2
Dense 12/12, ML-FT	30.3	42.4	32.7	37.5	23.9	39.5	34.2	34.4
MoE-64 12/12, ML-FT	31.6	43.5	33.4	38.8	25.7	39.8	36.0	35.5
Dense 24/24, ML-FT	31.8	43.4	36.0	38.8	25.6	40.3	36.3	36.0
MoE-128 24/24, ML-FT	31.9	43.6	34.9	**39.7**	26.5	40.4	**37.2**	36.3
Dense 24/24 Wide, ML-FT	**32.1**	**43.8**	**36.1**	39.4	**26.7**	**40.6**	36.9	**36.5**

	en-cs	en-de	en-ha	en-is	en-ja	en-ru	en-zh	Avg
Bilingual Dense 12/12	33.1	39.6	23.1	29.4	26.1	25.7	42.4	31.3
Dense 12/12	33.7	38.6	21.4	30.5	26.6	25.3	41.1	31.0
MoE-64 12/12	33.5	39.7	20.4	31.5	28.0	26.4	42.5	31.7
Dense 24/24	**34.0**	39.6	21.7	31.6	27.5	26.4	42.3	31.9
MoE-128 24/24	33.0	**40.2**	19.3	30.9	**28.8**	**26.6**	**42.8**	31.7
Dense 24/24 Wide	33.4	39.7	**23.4**	**32.0**	28.0	**26.6**	42.2	**32.2**
Bilingual Dense 12/12, BL-FT	35.7	39.5	23.3	29.4	27.7	26.0	43.0	32.1
Dense 12/12, ML-FT	35.0	39.1	22.9	30.5	26.9	25.6	41.5	31.6
MoE-64 12/12, ML-FT	35.9	40.4	24.1	29.6	28.8	26.4	43.0	32.6
Dense 24/24, ML-FT	35.8	40.1	24.1	31.6	28.7	**26.8**	42.5	32.8
MoE-128 24/24, ML-FT	36.4	**40.8**	**24.6**	31.2	**29.7**	**26.8**	**43.6**	**33.3**
Dense 24/24 Wide, ML-FT	**36.7**	40.6	**24.6**	**32.0**	29.3	26.7	43.0	**33.3**

Table 4: **Comparing Dense vs Sparsely Gated MoE Multilingual Models** before and after in-domain fine-tuning. *BL-FT* refers to finetuning a model on bilingual data, while *ML-FT* refers to finetuning a model on multilingual data, see Section 4.1.

	en-de
Bilingual 12/12	39.8
Bilingual 24/24	40.1
Bilingual 24/24 Wide	40.3
Bilingual 12/12 + FT	40.4
Bilingual 24/24 + FT	40.5
Bilingual 24/24 Wide + FT	40.4

Table 5: **Scaling Bilingual Models**.

dense baselines. Furthermore, the MoE model with 128 experts, which previously lagged behind the MoE model with 64 experts, now gives the best results for all but two directions. A possible hypothesis is that expert capacity in MoE models can retain specialized direction-specific finetuning better than dense models, where all language directions must share all model capacity while finetuning.

4.1.3 Effects of In-Domain Finetuning

Finetuning Improves Multilingual More than Bilingual. Table 6 compares the impact of finetuning across a variety of models. Multilingual systems benefit more from in-domain finetuning. As a result, the best multilingual system always outperforms the best bilingual system.

Multilingual Finetuning is better than Bilingual Finetuning. For multilingual models, there are two possible finetuning schemes (Tang et al., 2020). The multilingual model could be finetuned to specialize to the news domain in a multilingual fashion, concatenating the news data for all languages, *or* could be finetuned for each direction separately by training on bilingual news domain data. We compare *multilingual in-domain finetuning* with *bilingual in-domain finetuning* in Table 6. We find that multilingual finetuning is almost always better than bilingual finetuning, indicating that it is not necessary to take a multilingual system and specialize it to be bilingual via bilingual finetuning — a completely multilingual system still has the strongest performance.

4.1.4 Human Evaluation.

While a number of studies have been conducted on bilingual models to understand how BLEU correlates with human-perceived quality, few studies

	cs-en	de-en	ha-en	is-en	ja-en	ru-en	zh-en
Bilingual	28.3	38.0	28.3	34.5	21.1	38.0	30.8
Bilingual, BL-FT	30.4	42.8	30.3	35.5	24.6	39.5	36.2
Multilingual	29.0	37.9	24.5	36.8	21.2	36.9	30.4
Multilingual, BL-FT	31.8	43.3	31.9	37.0	26.5	**40.6**	36.8
Multilingual, ML-FT	**32.1**	**43.8**	**36.1**	**39.4**	**26.7**	**40.6**	**36.9**
	en-cs	**en-de**	**en-ha**	**en-is**	**en-ja**	**en-ru**	**en-zh**
Bilingual	33.1	39.6	23.1	29.4	26.1	25.7	42.4
Bilingual, BL-FT	35.7	39.5	23.3	29.4	27.7	26.0	**43.0**
Multilingual	33.4	39.7	23.4	32.0	28.0	26.6	42.2
Multilingual, BL-FT	36.1	40.3	24.2	30.1	28.7	**27.4**	**43.0**
Multilingual, ML-FT	**36.7**	**40.6**	**24.6**	**32.0**	**29.3**	26.7	**43.0**

Table 6: **Impact of Finetuning on Bilingual and Multilingual Models**. *BL-FT* refers to finetuning a multilingual model on bilingual data, while *ML-FT* refers to finetuning a multilingual model on multilingual data.

	cs-en	de-en	ha-en	is-en	ja-en
Bilingual	28.9	41.5	15.9	30.3	19.7
+ BT	28.3	38.0	28.3	34.5	21.1
Δ	-0.6	-3.5	+12.4	+4.2	+1.4
	en-cs	**en-de**	**en-ha**	**en-is**	**en-ja**
Bilingual	33.1	38.7	14.7	25.8	25.4
+ BT	33.2	39.4	23.1	29.4	26.1
Δ	+0.1	+0.7	+8.4	+3.6	+0.7

Table 7: **Impact of Large-scale Backtranslation in Bilingual Systems**.

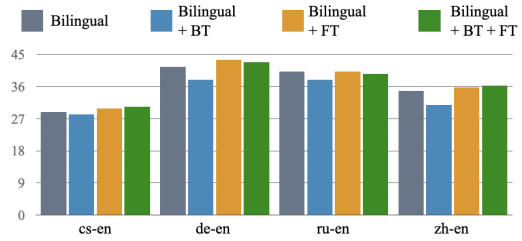


Figure 1: **Impact of In-Domain Finetuning after Backtranslation** on bilingual models.

	cs-en	de-en	ha-en	is-en	ja-en
Multilingual	27.7	37.6	16.5	34.2	20.8
+ BT	27.8	37.9	25.8	35.6	20.8
Δ	+0.1	+0.3	+9.3	+1.4	+0
	en-cs	**en-de**	**en-ha**	**en-is**	**en-ja**
Multilingual	33.7	39	10.0	27.0	26.9
+ BT	33.9	39.2	23.7	31.6	27.6
Δ	+0.2	0.2	+13.7	+4.6	+0.7

Table 8: **Impact of Large-scale Backtranslation in Multilingual Systems**.

have investigated multilingual ones. Given a bilingual system and a multilingual system with the same BLEU scores, we want to understand if there is anything intrinsically different in the multilingual system output that would impact human evaluation.

We study two directions: English to German and English to Russian. We ask human annotators who are fluent in source and native in target language to evaluate the translation quality between a bilingual system output and a multilingual system output. Both systems have similar BLEU scores, within decimal point difference. The translations are generated on the same English source sentence. We find no statistically significant difference between human evaluations of both systems, indicating that human evaluators have no innate preference for bilingual or multilingual systems.

4.2 Impact of Large-scale Backtranslation

Large-scale backtranslation has contributed to improvements in performance in machine translation models (Edunov et al., 2018), even when measured in human evaluation studies (Edunov et al., 2019; Bogoychev and Sennrich, 2019) — it is a component integrated into most modern translation systems. However, backtranslation also has downsides. Research has indicated that systems trained with large scale backtranslation data tend to overfit to the synthetically generated source sentences, producing lower quality translations when translating original source sentences (Marie et al., 2020). Further, backtranslation is fundamentally a form of data augmentation, which could have increasingly marginal effect when large-scale mined bitext is directly incorporated into training datasets. Beyond mining, multilingual translation can also be seen as an inherent form of data augmentation, as language directions can benefit from the training data

MMT	Model	cs-en	de-en	ha-en	is-en	ja-en	ru-en	zh-en	Avg	Incremental Δ
✗	**Bilingual**	28.9	41.5	15.9	30.3	19.7	40.2	34.8	30.2	—
✗	**+ Backtranslation**	28.3	38.0	28.3	34.5	21.1	38.0	30.8	31.3	+1.1
✗	**+ Finetuning**	30.4	42.8	30.3	35.5	24.6	39.5	36.2	34.2	+2.9
✓	**+ Multilingual**	32.1	43.8	36.1	39.4	26.7	40.6	36.9	36.5	+2.3
✓	**+ Ensemble**	32.3	44.5	37.2	39.9	27.2	40.9	37.8	37.1	+0.6
✓	**+ Reranking**	32.7	44.4	38.2	40.5	27.8	41.4	38.0	37.6	+0.5
✗	**WMT20 Winner**	29.9	43.8	—	—	26.6	39.2	36.9		
	Δ over WMT20	+2.8	+0.6	—	—	+1.2	+2.2	+1.1		

MMT	Model	en-cs	en-de	en-ha	en-is	en-ja	en-ru	en-zh	Avg	Incremental Δ
✗	**Bilingual**	33.1	38.7	14.7	25.8	25.4	25.8	40.0	29.1	—
✗	**+ Backtranslation**	33.1	39.6	23.1	29.4	26.1	25.7	42.4	31.3	+2.3
✗	**+ Finetuning**	35.7	39.5	23.3	29.4	27.7	26.0	43.0	32.1	+0.7
✓	**+ Multilingual**	36.4	40.8	24.6	31.2	29.7	26.8	43.6	33.3	+1.2
✓	**+ Ensemble**	36.8	41.1	25.0	32.5	29.7	26.9	43.6	33.7	+0.4
✓	**+ Reranking**	37.2	41.1	25.5	32.8	29.7	27.4	43.6	33.9	+0.2
✓	**+ Postprocessing**	39.8	42.6	25.5	34.5	29.8	28.8	48.2	35.6	+1.7
✗	**WMT20 Winner**	36.8	38.8	—	—	28.4	25.5	47.3		
	Δ over WMT20	+3.0	+3.8	—	—	+1.4	+3.3	+0.9		

Table 9: **Full Results of Submitted Models**. Starting with a bilingual baseline, we depict the incremental gain of different techniques across language pairs. Our final submission is a multilingual ensemble with noisy channel reranking, trained on all available data including backtranslation. On all language pairs, we observe improvement compared to the previous WMT20 winning models. The column *MMT* denotes if the model is multilingual. Note Hausa and Icelandic were not present in WMT20.

of other directions. Thus, we analyze further in this section the continued importance of backtranslation, even in multilingual systems.

Backtranslation in Bilingual Systems. First, we investigate if backtranslated data is still helpful, even after we augment the training dataset with mined and publicly available training data, beyond what is distributed in the WMT Shared Task. Our results in Table 7 show that backtranslation is helpful for 10 out of 14 directions, especially for low resource directions such as ha-en and is-en. However, for high resource directions such as de-en, ru-en, zh-en, bilingual systems trained with backtranslation had slightly lower validation BLEU compared to those trained without backtranslation.

Finetuning Corrects Overfitting to Translationese We further investigate the anomaly that high-resource directions can suffer from adding backtranslated data. Figure 1 shows that the minor BLEU degradation from adding backtranslation mostly disappears after applying in-domain finetuning. For zh-en and cs-en after in-domain finetuning, the system trained with backtranslation has stronger performance (+0.4 BLEU) compared to the system trained without backtranslation. Previous studies of this effect have indicated that backtranslation produces *translationese*, which has distinct qualities compared to original training data (Marie et al., 2020; Zhang and Toral, 2019; Graham et al., 2020). We hypothesize that in-domain finetuning, which trains the model on non-backtranslated data, can have a *corrective* effect that counteracts overfitting on translationese.

Backtranslation in Multilingual Systems. Table 8 summarizes the performance improvement from adding backtranslation to multilingual models in an ablation study. Overall, despite creating a fully unconstrained system with substantially greater training data and leveraging the data sharing potential of multilingual translation, we find that backtranslation still improves the performance. We believe this is influenced by the fact that backtranslation fully utilizes available monolingual data. While data mining techniques can identify potentially parallel sentences, it is naturally limited to identifying only a subset of the full monolingual data the algorithms utilize to mine.

4.3 Ablation on Components of Final Submission

Finally, we end by analyzing each aspect in our final submission and the cumulative effect. The effect of each component is shown in Table 9.

Bilingual Baselines. We find that our bilingual baselines have high BLEU scores, particularly for ru-en where our bilingual baseline is already stronger than the WMT20 winner. Overall, we observe that only en-ha and ha-en are significantly lower than 20 BLEU, indicating that curating a large amount of high quality bitext data is likely the most important basis of a strong system.

Backtranslation. Subsequently, we add backtranslated data. We observe that ha, is, and ja in particular observe large improvements in BLEU after adding backtranslated data, while other directions can actually slightly decrease in quality as a possible effect of translationese.

In-Domain Finetuning. We next evaluate the impact of in-domain finetuning and find an almost 3 BLEU improvement across directions for translation into English and 0.7 BLEU improvement for translation out of English. Across all language directions, finetuning is almost universally helpful.

Multilingual. Compared to bilingual models, multilingual models have stronger performance in every direction. Multilingual models benefit much more from scaling model size, as our largest architecture (MoE-128 24/24) has the best performance.

Ensembling. The effect of ensembling on average is fairly minor, but specific directions can see large improvements (such as +1 BLEU on zh-en).

Reranking. We then apply noisy channel reranking to the outputs of our final system. It is helpful across almost all directions, but does not have a huge effect on BLEU. On average, performance improves around 0.3 to 0.5 BLEU.

Postprocessing. Finally, we observe that postprocessing translated output to use standardized punctuation in each language is very important for BLEU scores when translating out of English. For example, Chinese in particular has a number of specific periods and double width punctuation characters, and properly using these produces almost +5 BLEU. However, we note that these techniques likely only improve BLEU score, and the effect on human evaluation is not well understood.

5 Conclusion

In this paper, we describe Facebook's multilingual model submission to the WMT2021 shared task on news translation. We employed techniques such as large scale backtranslation, bitext mining, large scale dense and sparse multilingual models, in-domain finetuning, ensembling, and noisy channel reranking. We provide extensive experiment results to quantify the impact of each technique, as well as how well they cumulatively stack to produce the final system. Our results demonstrate that multilingual translation can achieve state-of-the-art performance on both low resource and high resource languages, beating our strong bilingual baselines and previous years' winning submissions.

Acknowledgements

We'd like to thank Michael Auli and Halil Akin for helping getting this project started, help and advice along the way. We'd like to thank Holger Schwenk and Vishrav Chaudhary for their help in getting training data.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Shruti Bhosale, Kyra Yee, Sergey Edunov, and Michael Auli. 2020. Language models not just for pretraining: Fast online neural noisy channel modeling. *arXiv preprint arXiv:2011.07164*.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Peng-Jen Chen, Jiajun Shen, Matt Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook ai's wat19 myanmar-english translation task submission. *arXiv preprint arXiv:1910.06848*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage finetuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2019. On the evaluation of machine translation systems trained with back-translation. *arXiv preprint arXiv:1908.05204*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzman, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proc. of EMNLP*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

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.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81.

Kenneth Heafield. 2011. Kenlm: faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

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.

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*, pages 427–431. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997.

Robert Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. *CoRR*, abs/1804.08771.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2019. CCMatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Xu Tan, Jiale Chen, Di He, Yingce Xia, QIN Tao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 962–972.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. *Advances in Neural Information Processing Systems*, 33:2207–2219.

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.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81.

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'16)*, pages 3530–3534.

Tencent Translation System for the WMT21 News Translation Task

Longyue Wang* **Mu Li** **Fangxu Liu** **Shuming Shi** **Zhaopeng Tu**
Xing Wang **Shuangzhi Wu** **Jiali Zeng** **Wen Zhang**

Tencent AI Lab & Cloud Xiaowei

Abstract

This paper describes Tencent Translation systems for the WMT21 shared task. We participate in the news translation task on three language pairs: Chinese⇒English, English⇒Chinese and German⇒English. Our systems are built on various Transformer models with novel techniques adapted from our recent research work. First, we combine different data augmentation methods including back-translation, forward-translation and right-to-left training to enlarge the training data. We also apply language coverage bias, data rejuvenation and uncertainty-based sampling approaches to select content-relevant and high-quality data from large parallel and monolingual corpora. Expect for in-domain fine-tuning, we also propose a fine-grained "one model one domain" approach to model characteristics of different news genres at fine-tuning and decoding stages. Besides, we use greed-based ensemble algorithm and transductive ensemble method to further boost our systems. Based on our success in the last WMT, we continuously employed advanced techniques such as large batch training, data selection and data filtering. Finally, our constrained Chinese⇒English system achieves 33.4 case-sensitive BLEU score, which is the highest among all submissions. The German⇒English system is ranked at second place accordingly.

1 Introduction

In this year's news translation task, our translation team at Tencent AI Lab & Cloud Xiaowei participated in three shared tasks, including Chinese⇒English, English⇒Chinese and German⇒English. We used the same data strategies, model architectures and corresponding techniques for all tasks.

We hypothesized that different models have their own strengths and characteristics, and they can complement each other. Thus, we built various advanced NMT models which mainly differ in training data and model architectures. These models (i.e. DEEP, LARGE and LARGE-FFN) are empirically designed based on Transformer-Deep which has proven more effective than the Transformer-Big models (Li et al., 2019). In addition to the original multi-head self-attention, we also proposed a mixed attention strategy by combining relative position with the original one, which extends the self-attention to efficiently consider representations of the relative positions. We use a variation of relative position, the random attention (RAN) (Zeng et al., 2021). As a results, we combined these models at transductive fine-tuning stage.

In terms of data augmentation, we adapt back-translation (BT) (Sennrich et al., 2016a), forward-translation (FT) (Zhang and Zong, 2016) and right-to-left (R2L) (Zhang et al., 2019) techniques to generate large-scale synthetic training data. Different from the standard back-translation, we add noise to the synthetic source sentence in order to take advantage of large-scale monolingual text. In addition, we used tagged BT mechanism (i.e. add a special token to the synthetic source sentence) to help the model better distinguish the originality of data. All the parallel data and a large amount of monolingual data are used in corresponding data augmentation methods, and finally we combine them together to build strong baseline models.

To enhance the domain-specific knowledge, we introduced approaches at both data and model levels. First, we employed a hybrid data selection method (Wang et al.) to produce different fine-tuning datasets. More specifically, we apply language coverage bias (Wang et al., 2021a), data rejuvenation (Jiao et al., 2020) and uncertainty-based sampling (Jiao et al., 2021) to select content-

* Corresponding author: vinnylywang@tencent.com. The other authors are in alphabetical order of last name.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 216–224
November 10–11, 2021.

relevant and high-quality data from parallel and monolingual corpora. The news texts contain a number of sub-genres such as COVID-19 and government report. Thus, we fine-tuned a domain-specific model translate each sub-genre of text in the test set (i.e. "one domain one model").

We take advantage of the combination methods to further improve the translation quality. The "greedy search ensemble algorithm" (Li et al., 2019) is used to select the best combinations from single models. Furthermore, we propose an multi-model & multi-iteration transductive ensemble (m^2TE) method based on the translation results of the ensemble models. First, we divided models into two parts. Second, each part produced syntactic parallel testsets which is used to fine-tune another part of models. We repeated this procedure for N times.

This paper is structured as follows: Section 2 describes our advanced model architectures. We then present the data statistics and processing methods in Section 3. The methods and ablation study are detailed in Section 4 followed by final experimental results in Section 5. Finally, we conclude our work in Section 6.

2 Model Architecture

In this section, we mainly introduced three model architectures, which are empirically adapted from Transformer (Vaswani et al., 2017).

2.1 General Configurations

All models are implemented on top of the open-source toolkit Fairseq (Ott et al., 2019). Each single model is carried out on 8∼16 NVIDIA V100 GPUs each of which have 32 GB memory. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The gradient accumulation is used due to the high GPU memory consumption. We also employed large batching (Ott et al., 2018), which has significantly outperformed models with regular batch training. To speed up the training process, we conduct training with half precision floating point (FP16). We set max learning rate to 0.0007 and warmup-steps to 16000. All the dropout probabilities are set to 0.3. The detailed hyper-parameters of each model are summarized in Table 1.

2.2 Deep Model

Deep transformer has shown more effective performance than the TRANSFORMER-BIG models (Dou et al., 2018; Wang et al., 2019). We mainly modified the TRANSFORMER-BASE model by using a 40-layer encoder. To stabilize the training of deep model, we use the Pre-Norm strategy (Li et al., 2019), which is applied to the input of every sub-layer. The layer normalization was applied to the input of every sub-layer which the computation sequence could be expressed as: normalize → Transform → dropout → residual-add. The batch size is 5120 with 16 GPUs and "update-freq" is 1. We totally train models with 400K updates.

Module	DEEP	LARGE	LARGE-FFN
Encoder Layer	40	24	20
Attention Heads	8	16	16
Embedding Size	512	1024	1024
FFN Size	2048	4096	8192
Model Size	232M	514M	652M

Table 1: Hyper-parameters and model sizes of different models used in our systems.

2.3 Large Model

The large model is empirically designed based on TRANSFORMER-BIG models (Vaswani et al., 2017; Yang et al., 2020) with 24 encoder layers. More specifically, the batch size is 4096 with 8 GPUs and the "update-freq" is 4. We totally train models with 400K updates.

2.4 Large-FFN Model

We train Larger Transformers, the inner FFN dimension of which is twice as big as that of large Transformer. Specifically, in this setting, the FFN dimension is set to 8192. The number of encoder and decoder layers are 20 and 6 respectively. The number of head is 16. In addition to the original multi-head self-attention, we use a mixed attention strategy, where the random attention (Zeng et al., 2021) is combined with the original attention. In this way, the self-attention mechanism can efficiently consider representations of the relative positions, or distances between sequence elements. In training Large-FFN models, we set the batch size to 8192 toknes per GPU and the "update-freq" parameter is set to 8. The models are trained on 8 GPUs for about 3 days.

3 Data and Processing

3.1 Overview

Table 2 lists statistics of parallel and monolingual data we used in training our systems. The details

D.	L.	Parallel Data		Monolingual Data	
		# Sent.	# Word	# Sent.	# Word
In.	En	6.7M	128.2M	641.3M	13.1B
	Zh		116.0M	18.4M	466.0M
Out.	En	24.8M	613.8M	1.8B	35.5B
	Zh		550.3M	1.1B	28.4B
In.	En	58.5M	1.1B	641.3M	13.1B
	De		1.1B	353.8M	7.2B
Out.	En	3.4M	78.0M	1.8B	35.5B
	De		74.4M	417.0M	1.5B

Table 2: Data statistics of parallel and monolingual data. We combine sub-corpora according to in-domain (*In.*) and out-of-domain (*Out.*).

are as follows.

Chinese ⇔ English The bilingual data include all the available corpora provided by WMT2021: CCMT Corpus, News Commentary v16, ParaCrawl v7.1, Wiki Titles v3, UN Parallel Corpus V1.0 and WikiMatrix (except for Back-translated news). The monolingual English data consist of News crawl, News discussions, Common Crawl. The Chinese data consist of News crawl, News Commentary, Common Crawl and Extended Common Crawl.

English ⇒ German The bilingual data includes News Commentary v16, Europarl v10, ParaCrawl v7.1, Common Crawl, Wiki Titles v3, Tilde Rapid and WikiMatrix. For monolingual German data, we used News Crawl, News Commentary, Common Crawl and Extended Common Crawl. The monolingual English data are same as Chinese⇔English.

3.2 Pre-Processing

To process raw data, we applied a series of open-source/in-house scripts (Wang et al., 2014; Lu et al., 2014), including non-character filter, punctuation normalization, and tokenization/segmentation. The English and German languages are tokenized by Moses toolkit,[1] while the Chinese sentences are segmented by Jieba.[2] Furthermore, we generated subwords via BPE (Sennrich et al., 2016b) with 35K merge operations. The BPE models are trained on all the data in corresponding parallel and monolingual corpora instead of only parallel data. The vocabulary sizes of Chinese⇔English are 59100 and 48772, respectively. The vocabulary sizes of English⇒German are 41812 and 40948.

3.3 Filtering

To improve the quality of data, we filtered noisy sentences (pairs) according to their characteristics in terms of language identification, duplication, length, invalid string and traditional-simplified Chinese conversation. First, we filtered sentences whose language identification is invalid especially for English⇒German. Second, we removed similar sentences by comparing MD5 values of skelectons (i.e. removing stop words from sentences). About length, we filter out the sentences with length longer than 150 words. For more noisy corpora (e.g. ParaCrawl), we added hard filtering rules on special symbol, digital number, word length, punctuation number, HTML tags. Regarding bingling data, we further considered source-target ratio. For instance, the word ratio between the source and the target must not exceed 1:1.3 or 1.3:1. According to our observations, our method can significantly reduce noise issues including misalignment, translation error, illegal characters, over-translation and under-translation.

After filtering noisy training data, we used several data manipulation approaches to further improve the quality of the training data. We first followed Wang et al. (2021a) to identify the original languages of the bilingual sentence pairs, and explicitly distinguished between the source- and target-original training data using the bias-tagging strategy. We also identified the inactive training examples which contribute less to the model performance, rejuvenated them with self-training (Jiao et al., 2020). For the data augmentation with back-translation and forward-translation, we selected the most informative monolingual sentences by computing the uncertainty of monolingual sentences using the bilingual dictionary extracted from the parallel data (Jiao et al., 2021).

3.4 Evaluation

We regarded the WMT2019 test set as the validation set, and WMT2020 test set as the test set for all experiments. We ranked checkpoints according to either loss or BLEU on validation set. We used sacreBLEU score[3] as our evaluation metrics which is officially recommended. We also con-

[1] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer/tokenizer.perl.

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

[3] https://github.com/mjpost/sacrebleu.

#	Method	WMT20 Data			WMT21 Data		
		Data	*WMT19*	*WMT20*	*Data*	*WMT19*	*WMT20*
1	TRANSFORMER-DEEP	12.4M	30.1	30.2	31.5M	32.3	32.3
2	+ Forward-Translation	22.8M	33.1	31.8	49.9M	34.5	33.2
3	+ Back-Translation	32.4M	29.6	28.4	61.5M	32.4	32.5
4	+ Right-to-Left Training	24.8M	33.2	31.6	81.4M	34.4	33.1
5	+ 2 + 4	45.6M	33.9	32.2	99.8M	35.3	34.0
6	+ 2 + 3 + 4	58.0M	33.6	32.3	129.8M	35.5	34.3

Table 3: Effects of data augmentation methods on Chinese⇒English translation task. We used generally the same amount of monolingual data with the bilingual corpus. We used the DEEP model trained on the original bilingual data to construct the synthetic data, which is used together with the bilingual data to train the NMT models.

ducted post-processing such as *detokenizer.perl* on system output before sacreBLEU.

4 Method and Ablation Study

In this section, we conducted a comprehensive ablation study of the techniques used in this competition. We reported results on the Chinese⇒English task using the constrained data.

4.1 Data Augmentation

In this evaluation, we used three commonly-used data augmentation methods, namely *back-translation* (BT), *forward-translation* (FT) and *right-to-left training* (R2L), to exploit the useful monolingual data. All the synthetic parallel data is used together with the original parallel data to train NMT models.

Back-Translation This method first trains an intermediate target-to-source NMT system, which is used to translate monolingual target sentences into source language. Then the synthetic parallel corpus is used to train models together the bilingual data. In this work, we apply the noise back-translations method as introduced in Lample et al. (2018). When translating monolingual data we use an ensemble of two models to get better source translations. We follow Edunov et al. (2018) to add noise to the synthetic source data. Furthermore, we use a tag at the head of each synthetic source sentence as Caswell et al. (2019) does. To filter the pseudo corpus, we translate the synthetic source into target and calculate a Round-Trip BLEU score, the synthetic pairs are dropped if the BLEU score is lower than 30.

Forward-Translation This method is similar to BT but performs in a reverse manner. Recent studies showed that back-translation harms the translation performance, while forward-translation improves the performance (Edunov et al., 2020; Marie et al., 2020). Our preliminary experiments reconfirm their findings. Accordingly, we use forward-translation to construct the synthetic parallel data by translating the monolingual source sentences by the source-to-target NMT model, which is trained on the original bilingual data.

Right-to-Left Training The approach is proposed to address the error propagation problem in autoregressive generation task (Zhang et al., 2019). The main idea is to improve the agreement between translations generated by Right-to-Left (R2L) models and Left-to-Right (L2R) models. Following this work, we translate the source-side sentences in both parallel and monolingual corpora with both a R2L model and a L2R model, and use the translated pseudo corpus to improve the L2R model. For the right-to-left training, we trained another DEEP model on the bilingual data, whose target side is reversed. We drop the pseudo parallel data if the BLEU score lower than 15.

Experimental Results As shown in Table 3, we systematically investigated effects of 1) WMT20/WMT21 training data and 2) individual/combined data augmentation methods on Chinese⇒English translation task. For a comparison between the different training corpora of WMT20 and WMT21, we also reported results on the WMT20 training data ("WMT20 Data") used in last year (Wu et al., 2020b), and it consists of 12.4M sentence pairs after filtering. As seen, WMT21 extended around 19M sentence pairs, which improves the baseline model by +2.1 BLEU points. About data augmentation methods,

Source	Description	# Sent.
WMT	17-, 18-, 19-Dev/Test	7,466
	Source-Original	3,963
	Data Selection	1,000
	Data Augmentation	10,000
CWMT	08-, 09-, 11-Test	19,658
	Source-Original	8,036

Table 4: Statistics of data used for fine-tuning. "Source-Original" (SO) and "Data Selection" (DS) means respectively selecting source-original and domain-relevant examples from the whole WMT test sets. "Data Augmentation" indicates selecting data from the whole training corpus as extended data.

Finetune	W19	W20
None	35.3	34.0
WMT	44.3	35.5
+ CWMT	42.3	34.9
WMT (SO)	43.4	35.7
+ CWMT (SO)	43.3	35.1
+ DS	44.9	35.4
+ DA	42.5	35.8
+ ODOM	n/a	36.1

Table 5: Finetune results on the corresponding datasets.

we selected domain-relevant and high-quality sentences from all available monolingual data as listed in Table 2. To construct the new training data (i.e. combining authentic and synthetic data), we selected the same amount of monolingual data with the bilingual corpus. As seen, individually using FT and R2L can significantly improve the baseline model by around +1 BLEU point. About BT, we fount that it failed to outperform baseline in "WMT20 Data" while performs slightly better than baseline in "WMT21 Data". Finally, we trained the NMT models on the WMT21 training data augmented with the synthetic data generated by different data augmentation methods (up to 99.8M sentence pairs in total). We can further improve the performance by combining them together, demonstrating complementarity of different methods.

4.2 Fine-Tuning

We use in-domain finetune to further improve the model performance, which has proven effective on the WMT19∼20 news translation tasks (Sun et al., 2019; Meng et al., 2020; Li et al., 2020; Wu et al., 2020b). We construct different types of finetune data with the following approaches. Table 4 lists the statistics of data used for fine-tuning.

Previous Test Sets We follow the common practices to use WMT test sets in previous years as the finetune data. Specifically, we use WMT2017 development set, WMT2017 test set, WMT2018 and WMT19 test set.[4] Previous studies have shown that current NMT models suffer from the *language coverage bias* problem, which indicates the content-dependent differences between sentence pairs originating from the source and target languages, because the target-original data[5] can not improve translation performance (Wang et al., 2021a). Accordingly, we select the source-original examples (SO) from the test sets as the finetune data. Besides the WMT test sets, we also use the test sets from the CWMT competitions, which are available in the released data of WMT21 competition. In the CWMT testsets, each source sentence has four references, therefore we construct four sentence pair for each instance in the CWMT test sets.

In-Domain Training Data We employed data selection and data augmentation methods to select in-domain data from WMT/CWMT test sets and training corpus, respectively. More specifically, we employed BM25 algorithm to select relevant sentence pairs by regarding source-side of WMT20 test set as queries. As shown in Table 4, the "Data Selection" is a subset of WMT test sets. On the other hand, we extend the finetuning set by selecting in-domain data from the tranining corpus. We further use the RT and R2L approaches in Section 4.1 to augment the finetune data with the TRANSFORMER-DEEP model. Since the data augmentation approaches only require source-side sentences, we also construct the synthetic data for the WMT19 and WMT20 test sets.[6] We finetune the NMT model on the mixture of the additional synthetic corpus and the selected previous test sets.

One Domain One Model Li et al. (2020) argued that low-frequency words contain more domain information than high-frequency words, since low-

[4]In our final submission, we include WMT2019 and WMT2020 test sets in the fine-tune data.

[5]Target-original data are sentence pairs that are translated from the target language into the source language.

[6]In the final submission, we augment the WMT21 test set.

frequency words are mostly domain-specific nouns, etc., which may indicate the topic directly. Therefore, they adapt the TF-IDF algorithm to search and filter on the whole training set and then use them to train domain-specific models. We automatically to assigned domain labels to each source-side document in the test set. First, we used K-means clustering to obtain keywords of each document. Then, we proposed a rule-based method to classify each document in three categories: COVID-19, government report and other. In this experiment, we only focused on two specific domains and thus we trained two domain-specific models to translate COVID-19 and government report documents, respectively. The other documents are still dealt with a general-domain model.

Experimental Results As shown in Table 5, we investigated effects of different fine-tuning methods on Chinese⇒English translation task. As seen, source-original data is more effective than combining non-source-original one into finetuning dataset (35.5 vs. 35.7 BLEU). However, the CMWT dataset instead decrease the BLEU scores (-0.6 BLEU). The data reduction ("+DS") and expansion ("+DA") methods can not further improve the performance of baseline model (-0.3 and + 0.1 BLEU). Encouragingly, the "One Domain One Model" method can significantly improve the baseline model by +0.4 BLEU point.

4.3 Model Ensemble

Model ensemble is a widely used technique in previous WMT shared tasks, which can boost the performance by combining the predictions of several models at each decoding step (Li et al., 2019; Sun et al., 2019; Wang et al., 2018). In our work, we use two kinds of ensemble methods and finally the two are combined for further improvements.

Checkpoint Average For one model (same architecture and training data), we stored checkpoints according to their BLEU scores (instead of PPL or training time) on validation set. Then we combined top-L checkpoints (generate a final checkpoint) by averaging their weights to avoid stochasticity. To combine different models, we further ensembled the averaged checkpoint of each model. In our empirical experiments (Wang et al., 2020a), we find that this hybrid combination method outperforms solely combining checkpoints or models in terms of robustness and effectiveness.

Algorithm 1: Multi-Model & Multi-Iteration Transductive Ensemble

Input: Single Model M_n,
In-domain Seed D=$\{D_s, D_t\}$,
Ensemble N models E_N.
Output: New Model $M_n^{'}$

$t := 0$
while *not convergence* **do**
  Translate D_s with E_N and get $D_t^{E_N}$
  Train M_n on $D \cup D^{E_N}$ and get $M_n^{'}$,
  then $M_n = M_n^{'}$
  $t := t + 1$
end

Greedy Based Ensemble This method is proposed by Li et al. (2019), which adopts an easy operable greedy-base strategy to search for a better single model combinations on the development set. For more detail, please refer to the original paper. We also train single models with different hyper parameters to ensure the diversity. We refer to this method as Ensemble in the following.

Multi-Model & Multi-Iteration Transductive Ensemble Transductive ensemble (TE) is proposed by Wang et al. (2020b). The key idea is that source input sentences from the validation and test sets (in-domain seed) are firstly translated to the target language space with multiple different well-trained NMT models, which results in a pre-translated synthetic dataset. Then individual models are finetuned on the generated synthetic dataset. We propose an variation of TE, namely Multi-Model & Multi-Iteration TE (m^{2}TE) which is shown in Algorithm 1. The main difference from Iterative Transductive Ensemble (Wu et al., 2020b) is that E_N can be different groups of ensembled models (Deep, Large and Large-FFN models).

5 Final Results

In this section, we combined all the presented methods and techniques (detailed in Section 4) together and showed the final results in Table 6.

5.1 Chinese⇔English Translation Tasks

We train multiple single models in each settings. We found that the R2L method can significantly improve the baseline by about 1 BLEU score. It is surprising to find a gain of 2 BLEU improvement when combining all data augmentation meth-

System	Method	Zh⇒En		En⇒Zh		De⇒En	
		W19	*W20*	*W19*	*W20*	*W19*	*W20*
WMT2020 Competition Systems							
Meng et al.	*KD+Fine.+Ens.*	39.9	36.9	-	-	-	-
Li et al.	*XLM+Doc+Ens.+Fine.+Rerank*	-	-	40.5	49.1	-	-
Wu et al.	*KD+iteBT+Ens.*	-	-	-	-	43.8	43.5
Shi et al.	*KD+Ens.+Fine.+Rerank*	-	-	-	-	42.2	-
Wu et al.	*FT+R2L*	31.5	-	39.1	-	-	-
	FT+R2L+Fine.+Ens.	39.0	36.8	42.3	48.0	-	-
Our System	*BT+FT+R2L+Fine.+Ens.+Domain*	40.3	37.2	42.9	48.8	43.5	43.2

Table 6: Translation quality when combining all methods and techniques together.

ods. After we boost the in-domain corpus, we can further achieve 1∼2 more BLEU points on the different models, illustrating the effectiveness of fine-tuning. Specifically, we used corresponding development and test datasets and selected parallel data as in-domain corpus D. After training an NMT model M with the above methods, we finetune W on D with the same hyper parameters of training M. When testing on the WMT2020 test set, we achieve about 1.5 BLEU improvement. As the in-domain corpus is very limited, we propose a boosted finetune method by using the R2L training method to boost the finetune process. In our final submission, we add the WMT2020 test set to D, the batch size is set to 2048, the finetune finished after 3K training steps.

In our experiments, the ensemble models consists of 5 single models: 1 DEEP, 2 LARGE, 2 LARGER-FNN models. The simple ensembled model can outperform the best single model by 0.5∼2.0 BLEU scores. We then apply transductive ensemble to each group of models and the performance achieves 36.8 BLEU on Chinese⇒English task. Finally, we employed two fine-grained domain-specific models to translate COVID-19 and government report texts, respectively. This can further improve the model by +0.5 BLEU point. We also find that the single models that applied TE cannot bring further improvement to ensemble results. We do not apply re-ranking to this task, as we find that the improvement is insignificant.

5.2 German⇔English Translation Tasks

The baseline model are trained on bilingual data and R2L data. This boosts the BLEU score from 41.6 to 42.1. After adding BT and FT, we further improve the BLEU score by 1.3 BLEU scores.

For finetuning English⇒German models, we select the document whose source side is originally in German from all previous development and test dataset as in-domain corpus D. Single models are trained with the above methods are then fine-tune on D for one epoch with a fixed learning rate of 1e-4. In our final submission, the WMT2020 test set is added to D for better performance improvement. The fine-tuning can further achieve 0.93 BLEU improvement on the DEEP model.

In this task, the ensemble models consists of 3 single models: 1 DEEP, 1 LARGE, 1 LARGER-FNN models. The ensemble models outperform the best single model by 1.5 BLEU scores. Furthermorem we apply a rule-based post-processing procedure on punctuation and this can improve the BLEU score on development set by 0.5 point.

5.3 Official Results

The official automatic results (in terms of sacreBLEU) of our submissions for WMT 2021 are presented in Table 7. Among participated teams, our primary systems achieve the first and the second BLEU scores on Chinese⇔English and German⇔English, respectively. The experimental results demonstrates that our models can achieve the state-of-the-art performance.

In the future, we will integrate these useful techniques in the Tencent TranSmart (Huang et al., 2021), Mr. Translator (https://fanyi.qq.com), and Tencent Simultaneous Translation systems.

6 Conclusion

This paper presents the Tencent Translation systems for WMT2021 news translation tasks. We investigate various deep architectures to build strong baseline models. Then popular data augmentation

System	Zh-En	En-Zh	De-En
Best Official	33.4	36.9	35.0
Our System	33.4	36.5	34.9

Table 7: Official sacreBLEU scores of our submissions for WMT21 news task. The "Best Official" denotes the best performance among all participant teams.

methods such as BT, FT and R2L are combined to improve their performances. We demonstrate that in-domain fine-tuning and fine-grained domain modelling are effective to further improve domain-specific quality. Besides, our proposed greed-based ensemble algorithm and transductive ensemble method play key roles in our systems. Among participated teams, our primary systems achieve the first and the second BLEU scores on Zh⇒En and De⇒En, respectively. In the future, we will adopt useful methods to our advanced non-autoregressive translation models (Ding et al., 2021b,a) and investigate the effects of pre-training on NMT (Liu et al., 2021a,b).

It is worth mentioning that most advanced technologies reported in this paper are also adapted to our systems for biomedical translation task (Wang et al., 2021b), which achieve three 1st ranks in German/French/Spanish⇒English tasks.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *WMT*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2021a. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021b. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP*.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *ACL*.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. *arXiv preprint arXiv:2105.13072*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *EMNLP*.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael R Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *ACL*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *ICLR*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *WMT*.

Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2020. Sjtu-nict's supervised and unsupervised neural machine translation systems for the WMT20 news translation task. In *WMT*.

Xuebo Liu, Longyue Wang, Derek F Wong, Liang Ding, Lidia S Chao, Shuming Shi, and Zhaopeng Tu. 2021a. On the complementarity between pre-training and back-translation for neural machine translation. In *EMNLP*.

Xuebo Liu, Longyue Wang, Derek F Wong, Liang Ding, Lidia S Chao, Shuming Shi, and Zhaopeng Tu. 2021b. On the copying behaviors of pre-training for neural machine translation. In *ACL*.

Yi Lu, Longyue Wang, Derek F Wong, Lidia S Chao, and Yiming Wang. 2014. Domain adaptation for medical text translation using web resources. In *WMT*.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *ACL*.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. Wechat neural machine translation systems for WMT20. In *WMT*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. *NAACL*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *ACL*.

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

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao. 2020. Oppo's machine translation systems for WMT20. In *WMT*.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *WMT*.

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 *NIPS*.

Longyue Wang, Yi Lu, Derek F Wong, Lidia S Chao, Yiming Wang, and Francisco Oliveira. 2014. Combining domain adaptation approaches for medical text translation. In *WMT*.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020a. Tencent AI Lab machine translation systems for the WMT20 chat translation task. In *WMT*.

Longyue Wang, Derek F Wong, Lidia S Chao, Yi Lu, and Junwen Xing. A systematic comparison of data selection criteria for smt domain adaptation. *The Scientific World Journal*, 2014.

Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, and Chao Bian. 2018. Tencent neural machine translation systems for WMT18. In *WMT*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *ACL*.

Shuo Wang, Zhaopeng Tu, Zhixing Tan, Shuming Shi, Maosong Sun, and Yang Liu. 2021a. On the language coverage bias for neural machine translation. In *Findings of ACL*.

Xing Wang, Zhaopeng Tu, and Shuming Shi. 2021b. Tencent ai lab machine translation systems for the WMT21 biomedical translation task. In *WMT*.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020b. Transductive ensemble learning for neural machine translation. In *AAAI*.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020a. The volctrans machine translation system for WMT20. In *WMT*.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020b. Tencent neural machine translation systems for the WMT20 news translation task. In *WMT*.

Yilin Yang, Longyue Wang, Shuming Shi, Prasad Tadepalli, Stefan Lee, and Zhaopeng Tu. 2020. On the sub-layer functionalities of transformer decoder. In *EMNLP*.

Jiali Zeng, Shuangzhi Wu, Yongjing Yin, Yufan Jiang, and Mu Li. 2021. Recurrent attention for neural machine translation. In *EMNLP*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019. Regularizing neural machine translation by target-bidirectional agreement. In *AAAI*.

HW-TSC's Participation in the WMT 2021 News Translation Shared Task

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu,
Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang,
Lizhi Lei, Min Zhang, Hao Yang, Ying Qin,
Huawei Translation Service Center, Beijing, China
{weidaimeng,lizongyao,wuzhanglin2,yuzhengzhe,
chenxiaoyu35,shanghengchao,guojiaxin1,wangminghan,
leilizhi,zhangmin186,yanghao30,qinying}@huawei.com

Abstract

This paper presents the submission of Huawei Translate Services Center (HW-TSC) to the WMT 2021 News Translation Shared Task. We participate in 7 language pairs, including Zh/En, De/En, Ja/En, Ha/En, Is/En, Hi/Bn, and Xh/Zu in both directions under the constrained condition. We use Transformer architecture and obtain the best performance via multiple variants with larger parameter sizes. We perform detailed pre-processing and filtering on the provided large-scale bilingual and monolingual datasets. Several commonly used strategies are used to train our models, such as Back Translation, Forward Translation, Multilingual Translation, Ensemble Knowledge Distillation, etc. Our submission obtains competitive results in the final evaluation.

1 Introduction

This paper introduces our submission to the WMT 2021 News Translation Shared Task. We participate in seven language pairs including Chinese/English (Zh/En), German/English (De/En), Japanese/English (Ja/En), Hausa/English (Ha/En), Icelandic/English (Is/En), Hindi/Bengali (Hi/Bn), and Xhosa/Zulu (Xh/Zu) in both directions. We consider that the officially provided dataset has the acceptable size and quality and therefore only participate in the constrained evaluation. Our method is mainly based on previous works but with fine-grained data cleansing techniques and language-specific optimizations.

For each language pair, we perform multi-step data cleansing on the provided dataset and only keep a high-quality subset for training. At the same time, several strategies are tested in a pipeline, including Backward (Edunov et al., 2018) and Forward(Wu et al., 2019a) Translation, Multilingual Translation (Johnson et al., 2017), Right-to-Left Models (Liu et al., 2016), Iterative Joint Training (Zhang et al., 2018), Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019) , Fine-Tuning (Sun et al., 2019), Ensemble (Garmash and Monz, 2016), and PostProcess.

We combined all the techniques mentioned above and the overall training process is shown in Figure 1. Section 2 focuses on our data processing strategies while section 3 describes our training techniques, including model architecture and iterative training, etc. Section 4 explains our experiment settings and training processes and section 5 presents our experiment results.

2 Data

2.1 Data Source

For all language pairs, we follow the constrained data requirements and take full advantages of the bilingual and monolingual training data available. Table 1 lists the data sizes of each language pair before and after filtering.

2.2 Data Pre-processing

We use following operations to pre-process the data:

- Filter out repeated sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Convert XML escape characters.
- Normalize punctuations using Moses (Koehn et al., 2007).
- Delete html tags, non-UTF-8 characters, unicode characters and invisible characters.
- Filter out sentences with mismatched parentheses and quotation marks; sentences of which punctuation percentage exceeds 0.3; sentences with the character-to-word ratio greater than 12 or less than 1.5; sentences of which the source-to-target token ratio higher

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 225–231

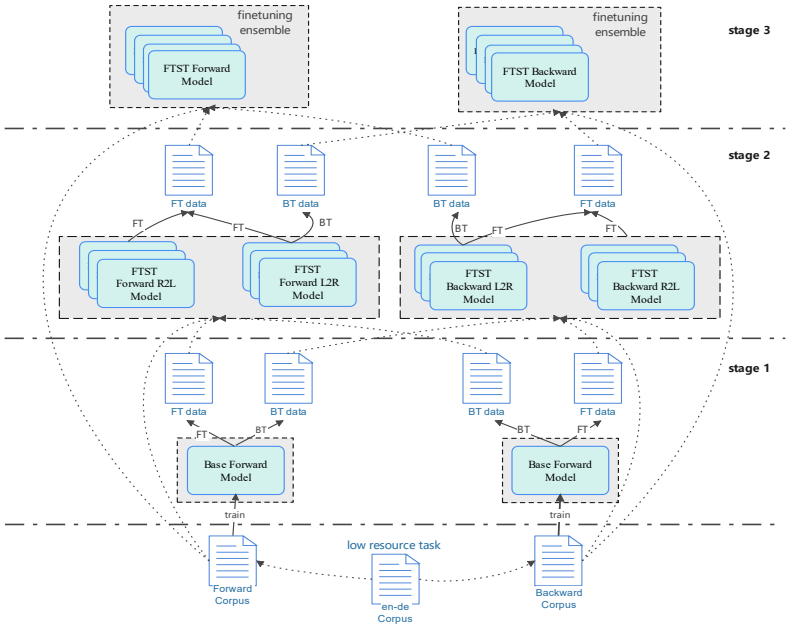


Figure 1: This figure shows the training process for the WMT 2021 News Translation Shared Task, which consists of three stages. In stage 1, one forward model and one backward model are trained. In stage 2, the synthetic data by FTST is used to train L2R and R2L models. In stage 3, the synthetic data by enhanced models are used to train models. Finally, model ensemble is used to boost the performance.

than 3 or lowers than 0.3; sentences with more than 120 tokens.

- Apply langid (Joulin et al., 2016b,a) to filter sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter sentence pairs with poor alignment.

We perform the additional steps to process Chinese data:

- Convert traditional Chinese characters to simplified ones.
- Convert fullwidth forms to halfwidth forms.

Data sizes before and after cleansing are listed in Table 1.

2.3 Data Selection

Since the news (in-domain) monolingual data in some tasks is not sufficient, it is necessary to obtain data from Common Crawl. We use Fasttext (Joulin et al., 2016a) to train a binary classification model to distinguish between in-domain and out-domain data.

3 System Overview

3.1 Model

Transformer (Vaswani et al., 2017) has been widely used for machine translation in recent years, which has achieved good performance even with the most primitive architecture without much modifications. Therefore, we choose to start from Transformer-Big and consider it as a baseline. Four variants of Transformer are also evaluated during the experiments, which are the model with wider FFN layers proposed in (Ng et al., 2019), and the deeper encoder version proposed in (Sun et al., 2019). Here, we use the following four variants:

- Deep 25-6 model: The number of the encoder layers is adjusted to 25 based on the transformer base model architecture and layer normalization is added. The other settings remain the same as the base model.
- Deep 35-6 model: The number of the encoder layers is adjusted to 36 based on the transformer base model architecture and layer normalization is added. The other settings remain the same as the base model.

language pairs	Raw bi data	Filter bi data	Used mono data
Zh/En	37.8M	16.5M	En: 150M, Zh:150M
De/En	95M	79M	En: 230M, De: 317M
Ja/En	18M	13.5M	En: 300M, Ja: 300M
Ha/En	0.73M	0.59M	En: 8M, Ha:8.65M
Is/En	5.69M	4.04M	En: 20M, Is: 18M
Hi/Bn	3.53M	3.4M	Bn: 59.3M, Hi: 45.8M

Table 1: Bilingual data sizes before and after filtering, and monolingual data used in tasks.

- Deep 35-6 big model: This model features 35-layer encoder, 6-layer decoder, 768 dimensions of word vector, 3076 dimensions of FFN, 16-head self-attention, and pre-norm.
- Deep 25-6 large Model: This model features 25-layer encoder, 6-layer decoder, 1024 dimensions of word vector, 4096 dimensions of FFN, 16-head self-attention, and pre-norm.

3.2 Data Augmentation

Back-translation (Edunov et al., 2018) is an effective way to boost translation quality by using monolingual sentences to generate synthetic training parallel data. As described in (Wu et al., 2019b), similar to back translation, the monolingual corpus in source language can also be used to generate forward translation text with a trained MT model, and the generated forward and backward translation data can both be merged with the authentic bilingual data. This strategy can increase the data size to a large extent.

We take full advantages of the officially provided monolingual data for data augmentation. In terms of back translation, we adopt top-k sampling for high-resource languages, and adopt beam search for low-resource languages. With regard to forward translation, we translate monolingual data using beam search. Through sampling, we ensure that the sizes of data generated by forward and back translation are relatively equal. In this paper, we refer to the combination of forward and sampling back translation as FTST.

3.3 Iterative Joint Training

Zhang et al. (2018) propose a new iterative joint training method, that is, using monolingual data from both source and target sides to train a source-to-target (forward) model and a target-to-source (backward) model at the same time. The two models generate synthetic data for each other. The advantage of such method is that both of the two models gain improvement after each iteration with the synthetic data provided by the other, and then can generate synthetic data with higher quality. Such training procedure is repeated after the two models converge.

3.4 Multilingual Translation

Johnson et al. (2017) propose a simple solution to use a single neural machine translation model to translate among multiple languages, and the model requires no change to the model architecture. Instead, the model introduces an artificial token at the beginning of the input sentence to specify the required target language. All languages use a shared vocabulary. There is no need to add more parameters. In low-resource tasks, we select a portion of the En-De bilingual data and conduct a joint training. The experiment shows that a multilingual model can improve the translation quality of low-resource languages to a large extent.

3.5 Right-to-Left Models

The approach of Right-to-Left is proposed by (Liu et al., 2016). The main idea is to integrate information of Right-to-Left (R2L) models to Left-to-Right (L2R) ones. Following this strategy, we translate the source sentences of the monolingual data with both R2L models and L2R models. In the Zh/En and De/En tasks, we use the R2L model to synthesize forward translation data using beam search and mix the synthetic data with the L2R synthetic data for iterative joint training.

3.6 Ensemble Knowledge Distillation

Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019) improves the performance of a student model by distilling knowledge from a group of trained teacher models. Comparing with some soft label distillation methods, the EKD for NMT is relatively straightforward, which can be implemented by training the student models on the combination of the original training set and the

translation from the ensembled teacher model on the training set. In our experiments, we ensemble models as the teacher model to translate the wmt21 test set, and use the translate results to further fine-tune models.

3.7 Fine-tuning

Previous works have demonstrated that fine-tuning a model with in-domain data, such as last year's test set, could effectively improve the performance of this year (Sun et al., 2019). We use the dev and test sets from previous years, coupled with data generated by ensemble knowledge distillation and noises added to the target side, to fine-tune models and achieve further improvements.

3.8 Ensemble

Model ensemble is a widely used technique in previous WMT workshops (Garmash and Monz, 2016), which can improve the performance by combining the predictions of several models at each decoding step. In our work, we ensemble models with different architectures to further improve system performances. For Zh/En and De/En, we experimented with a combination of the Deep 35-6 big model and the Deep 25-6 large model to ensemble. For all language pairs, we train multiple models to ensemble by shuffle the data.

4 Experiment Settings

4.1 Settings

We use the open-source fairseq (Ott et al., 2019) for training and sacreBLEU (Post, 2018) to measure system performances. The main parameters are as follows: Each model is trained using 8 GPUs. The size of each batch is set as 2048, parameter update frequency as 32, and learning rate as 5e-4 (Vaswani et al., 2017). The number of warmup steps is 4000, and model is saved every 1000 steps. The architectures we used are described in section 3.1. We adopt dropout, and the rate varies across different language pairs. Marian (Junczys-Dowmunt et al., 2018) is used for decoding during inference.

4.2 Training Process

We employ iterative training and phase-based data augmentation. Figure 1 shows our training process in details. The specific steps are as follows:

1) Process data using methods described in section 2.2. Train one forward model and one backward model.

System	en2zh	zh2en
baseline	39.1	26.5
FTST	45.1 (+6.0)	32.4 (+5.9)
in-domain FTST + R2L	46.2 (+1.1)	34.4 (+2.0)
finetuning	46.5 (+0.3)	34.8 (+0.4)
ensemble	46.7 (+0.2)	34.9 (+0.1)
wmt21 final submit	35.1	28.9

Table 2: The experimental result of Zh/En tasks

2) Generate back translation and forward translation data. Mix the data with parallel training data and train three forward L2R models and three backward models. At the same time, train three R2L models for generating R2L forward translation data, in order to improve the diversity of synthetic data.

3) Split monolingual data into several sets. Generate back translation and forward translation data using models trained in step 2. Mix sampled synthetic data with bilingual training data and train four forward models and four backward models.

4) Average the last five checkpoints of each model and fine-tune it. Ensemble models to produce the final system.

5 Results and analysis

5.1 Zh/En

We use methods described in Section 2.2 for data processing. Four model architectures mentioned in Section 3.1 are employed to increase system diversity. On the basis of bilingual baselines model, we use FTST data augmentation to further enhance model performance.

Table 2 lists the results of our submission on WMT 2020 News Task test set. Comparing with the baseline model, FTST leads to 6.0 BLEU increase on en2zh direction and 5.9 BLEU increase on the opposite direction. We conduct data distillation on source sentences from WMT 2017 and 2018 news test sets, mix the generated data with the original data, and add noises to the target side. We fine-tune the model using the mixed data and achieve 1.1 BLEU and 2.0 BLEU increases on en2zh and zh2en directions, respectively. We then conduct a second-round FTST data augmentation on the fine-tuned model. In this round, we adopt the R2L model. We conduct data distillation on source sentences from WMT 2017-2018 news test sets, mix

System	en2de	de2en
baseline	33.1	39.7
FTST	34.2 (+1.1)	40.8 (+1.1)
FTST + R2L	34.5 (+0.3)	41.1 (+0.3)
finetuning	38.2 (+3.7)	43.1 (+2.0)
ensemble	38.3 (+0.1)	43.4 (+0.3)
postprocess	39.7 (+1.4)	-
wmt21 final submit	29.8	34.7

Table 3: The experimental result of De/En tasks

System	en2ja	ja2en
baseline	36.4	21.4
iterative FTST	39.2 (+2.8)	23.1 (+2.7)
finetuning	42.9 (+3.7)	25.3 (+2.2)
ensemble	43.6 (+0.7)	26.0 (+0.7)
wmt21 final submit	45.4	26.5

Table 4: The experimental result of Ja/En tasks

the generated data with the WMT 2017-2018 test sets, and add noises to the target side. We fine-tune the model using the mixed data and achieve 0.3 BLEU and 0.4 BLEU increases on en2zh and zh2en directions, respectively. Finally, ensemble further leads to 0.2 BLEU increase on the en2zh direction and 0.1 BLEU increase on the opposite direction. When submitting the final results, we further fine-tune the model with WMT 2019 and 2020 test sets. Our models achieve 35.1 BLEU on the en2zh direction and 28.9 BLEU on the zh2en direction when measuring with the WMT 2021 News Task test set.

5.2 De/En

For the En-De task, we adopt the Deep 36-5 big model and Deep 25-6 large model, as described in section 3.1. We use Moses for English and German word segmentation. The training data are segmented by a shared SentencePiece model. The source and target side each has a vocabulary with 32K words. We process all data using filter methods described in section 2.2.

Table 3 lists the results of our submission on WMT 2020 News Task test set. Comparing with the baseline model, two rounds of FTST data augmentation contribute to 1.4 BLEU increase on each directions. We conduct data distillation on source sentences from WMT 2020 news test sets, mix the generated data with the WMT 2018 and WMT 2019 test sets after adding noises to the target side. We fine-tune the model using the mixed data and achieve 3.7 BLEU and 2.0 BLEU increases on en2de and de2en directions, respectively. Ensemble further leads to 0.1 BLEU increase on the en2de direction and 0.3 BLEU increase on the opposite direction. Ensemble does not have significant impact on this task. It should be noted that we find that the quotation marks generated by the en2de model does not comply with the German standard, so we add a correction script to the post-processing(just convert English quotation marks to German quotation marks), which surprisingly leads to 1.4 BLEU increase. When submitting the final results, we further fine-tune the model with WMT 2020 test set. Our submitted models achieve 29.8 BLEU on the en2de direction and 34.7 BLEU on the de2en direction when measuring with the WMT 2021 News Task test set.

5.3 Ja/En

For Ja/En task, we adopt the same settings as that for the Zh-En task. The dropout rate is set to 0.1. The training data are segmented by a shared SentencePiece model. The source and target side each has a vocabulary with 32K words. The size of parallel data after cleansing is 13.5M. We sampled 150M English monolingual data from News Crawl and 300M Japanese monolingual data from News Crawl and Common Crawl (150M from each source).

Table 4 lists the results of our submission on WMT 2020 News Task test set. Comparing with the baseline model, iterative FTST data augmentation contribute to 2.8 BLEU and 1.7 BLEU increases on the en2ja and ja2en directions respectively. We conduct data distillation on source sentences from WMT 2020 news test sets, mix the generated data with the WMT 2020 dev set after adding noises to the target side. We fine-tune the model using the mixed data and achieve 3.7 BLEU and 2.2 BLEU increases on en2ja and ja2en directions, respectively. We train four models on each direction and ensemble further leads to 0.9 BLEU increase on the en2ja direction and 1.0 BLEU increase on the opposite direction. When submitting the final results, we further fine-tune the model with WMT 2021 dev set. Our submitted models achieve 45.4 BLEU on the en2ja direction and 26.5 BLEU on the j2en direction when measuring with the WMT 2021 News Task test set.

System	en2ha	ha2en	en2is	is2en	hi2bn	bn2hi	xh2zu	zu2xh
baseline	2.8	7.7	18.3	25.1	7.4	18.0	2.1	6.2
multilingual (add en2de data)	14.9	18.9	20.2	28.0	9.2	18.3	7.3	8.1
iFTBT	19.7	23.2	23.5	32.4	10.4	19.4	9.3	9.2
wmt21 final submit	20.3	17.5	27.5	38.4	13.0	21.9	11.8	9.9

Table 5: The experimental result of low resource tasks. iBTFT indicates that multiple rounds of BTFT are used for data enhancement.

5.4 Low resource tasks

We use the same strategy to deal with low resource tasks (En-Ha, En-Is, Bn-Hi and Xh-Zu). We train a bilingual baseline model and a monolingual baseline model for each direction. Every multilingual model is trained with 10x bilingual data sampled from the training corpora and 50M En-De parallel data. For en2ha, en2is, hi2bn and xh2zu, we use en2de data for training. For other language directions, we use de2en data for training.

Table 5 lists the results of our submission on dev set. On the eight language directions, all multilingual models gain huge improvements when comparing with the bilingual baseline model. Particularly, En-Ha achieves the greatest improvements: 12.1 BLEU on en2ha direction and 11.20 on ha2en direction. Bn-Hi achieves the slightest improvements: 0.4 BLEU on bn2hi direction and 1.78 on hi2bn direction. The results demonstrate that the fewer the bilingual data, the greater impact a multilingual model has. In other extremely low-resource scenarios, the improvement gained by a multilingual model for En-Ha is greater than that for the Xh-Zu task. We think the reason lies in the differences of language similarities. On the basis of multilingual models, we conduct data augmentation as described in section 3.2. We adjust sampling ratios according to the monolingual data size of each languages. Our data augmentation strategy achieves improvements on all eight language directions, from 1.1 BLEU to 4.4 BLEU increase. When conduct the second-round FTST data augmentation, we only get a slight increase on the En-Ha task: 0.2 BLEU on en2ha direction and 0.9 on ha2en direction. We also leverage fine-tuning and ensemble techniques to further improve our model performances. Finally, we get the highest BLEU score on the xh2en direction and the second highest BLEU score on the en2ha direction.

6 Conclusion

This paper presents the submissions of HW-TSC to the WMT 2021 News Translation Task. For each direction in all pairs, we perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated. Our experiments show that in low-resource scenarios, multilingual model that utilizing data from other languages can improve system performance to a large extent. Data augmentation strategy is still effective for multilingual models. Our submissions finally achieves competitive results in the evaluation.

References

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

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.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

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.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 News Translation Task Submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

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

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

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.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019a. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019b. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

LISN @ WMT 2021

Jitao Xu
Univ. Paris-Saclay,
& CNRS, LISN

Pham Minh Quang
Univ. Paris-Saclay,
& CNRS, LISN
& Systran

Sadaf Abdul Rauf
Univ. Paris-Saclay,
& CNRS, LISN

François Yvon
Univ. Paris-Saclay,
& CNRS, LISN

{firstname.lastname}@limsi.fr

Abstract

This paper describes LISN's submissions to two shared tasks at WMT'21. For the biomedical translation task, we have developed resource-heavy systems for the English-French language pair, using both out-of-domain and in-domain corpora. The target genre for this task (scientific abstracts) corresponds to texts that often have a standardized structure. Our systems attempt to take this structure into account using a hierarchical system of sentence-level tags. Translation systems were also prepared for the News task for the French-German language pair. The challenge was to perform unsupervised adaptation to the target domain (financial news). For this, we explored the potential of retrieval-based strategies, where sentences that are similar to test instances are used to prime the decoder.

1 Introduction

This paper describes LISN's[1] submissions to the translation shared tasks at WMT'21, where we took part in two shared tasks. For the biomedical translation tasks, we have developed resource-heavy systems for the English-French language pair, using a diversity of out-of-domain and in-domain corpora, thus continuing the efforts reported in (Abdul Rauf et al., 2020). Like for previous years shared task, the target genre (scientific abstract) corresponds to texts that often have a standardized structure comprising typical subsections of one to five lines. Standard subsections report the OBJECTIVE, the METHOD, or the RESULTs of the study. Our systems for this year attempt to take this structure into account using sentence-level tags, with the hope to capture some of the document structure and the phraseology of the domain into account. These systems are documented in Section 2.

Translation systems were also prepared for the News task for the French-German language pair. The challenge this year was to perform unsupervised adaptation to the target domain (financial news), with no further detail regarding the test data. In particular, the organizers did not release any development data to tune systems. In this setting, we explored the potential of using a retrieval-based strategy, where sentences that are similar to the test instances are used to help the decoding. In this approach, introduced in (Bulte and Tezcan, 2019) and further explored in (Xu et al., 2020; Pham et al., 2020), translation is a two-step process: a retrieval phase, which identifies sentences that resemble the source test sentence in parallel corpora. These sentences and their translation are then used to *prime* the decoder: inserting relevant translations examples in the decoder's context should help to select the right translations, especially for words and terms from the test domain. These systems are described in Section 3.

2 MT for biomedical texts

In this section, we describe our participation to the biomedical task for WMT'21, in which we participated in both English to French and French to English directions. English-French is a reasonably resourced language pair with respect to biomedical parallel corpora, allowing us to train our Neural Machine Translation (NMT) systems (Vaswani et al., 2017) with in-domain corpora as well as large out-of-domain data that exists for this language pair. Like for last year (Abdul Rauf et al., 2020), our first goal is to make the best of all the available data, including supplementary in-domain monolingual data. Our corpora are described in Section 2.1.

For this year's participation, we also attempt to take the internal structure of biomedical abstracts into account. Many of these abstracts follow what is often refered to as the "IMRAD format", comprising the following subparts: INTRODUCTION,

[1] LISN [Laboratoire Inderdisciplinaire des Sciences du Numérique] is the new name of the laboratory formerly known as LIMSI.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 232–242

Parallel			
Corpus	**Wrds (M)**		**Sents.**
	English	**French**	
Ufal	89.5	100.3	2.72 M
Edp	0.04	0.04	2.44 K
Medline titles	5.97	6.43	0.63 M
Medline abstracts	1.23	1.44	0.06 M
Scielo	0.17	0.21	7.84 K
Cochrane-Reference	2.23	2.74	0.12 M
Cochrane-PE	0.43	0.53	20.5 K
Cochrane-GooglePE	0.63	0.77	30.3 K
Taus	20.1	23.2	0.88 M
Mlia	19.0	23.0	1.0M
IR Retrieved	13.2	14.7	3.6M
Development			
Medline 18	5.7K	6.9K	265
Medline 19	9.8K	12.4K	537
Test			
Medline 20	12.7K	16.2K	699
Monolingual			
Corpus	**English**	**French**	**Sent.**
Lissa_Fr	8.79	7.70	0.33 M
Med_Fr	16.3	16.2	0.06 M
IsTex_Fr	6.92	7.84	0.42M
Med_En	3.40	4.02	0.22M
Out Domain			
Corpus	**English**	**French**	**Sent.**
Out-of-domain	1139	1292	35M

Table 1: Data sources for the biomedical task

METHODS, RESULTS, and DISCUSSION (Sollaci and Pereira, 2004). This structure can be explicit in documents through dedicated headings or remain implicit. Our experiments aim to explore how to use this information in NMT and to measure the correlated impact. We notably expect that by informing the system with sub-document information, it will learn the typical style and phraseology of sentences occurring in each part.

For this purpose, we identified in our data all the abstracts that were conforming to this basic structure and worked to make this structure as explicit and standardized as possible. This notably implied to normalize the mains headings, as some variation was observed: for instance, ANALYSIS may be replaced with DISCUSSION, and additional subparts (OBJECTIVES, CONCLUSION) are also be observed. To incorporate the standard IMRaD format we mapped each subheading to the corresponding IMRaD subpart using a system of tags. Details regarding this process are given in Section 2.2.

All systems are based on the Transformer architecture of Vaswani et al. (2017). We were able to achieve appreciable gains both from back-translation and document structure processing. The results are discussed in Section 2.4.

2.1 Corpus and preprocessing

We trained our baseline systems on a collection of in domain biomedical texts as well as out-of-domain parallel corpus. Table 1 details the corpora used in training.

2.1.1 Parallel corpora

We gathered parallel and monolingual corpora available for English-French in the biomedical domain. The former included the biomedical texts provided by the WMT'20 organizers: Edp, Medline abstracts and titles (Jimeno Yepes et al., 2017), Scielo (Neves et al., 2016) and the Ufal Medical corpus[2] consisting of Cesta, Ecdc, Emea (OpenSubtitles), PatTR Medical and Subtitles. In addition, we used the Cochrane bilingual parallel corpus (Ive et al., 2016)[3], the Taus Corona Crisis corpus[4] and the Mlia Covid corpus.[5] We finally experimented with additional in-domain data selected using Information Retrieval (IR) techniques from general domain corpora including News-Commentary, Books and Wikipedia corpus obtained from the Open Parallel Corpus (OPUS) (Lison and Tiedemann, 2016). These were selected using the data selection scheme described in (Abdul-Rauf and Schwenk, 2009). Medline titles were used as queries to find relevant sentences. We used the `2-best` sentences returned from the IR pipeline as additional corpus.

Our out-of-domain corpora include the parallel data provided by the WMT14 campaign for French-English: Gigafr-en, Common Crawl, Europarl, News Commentary and the UN corpora.

For development purposes, we used Medline test sets of WMT'18 and 19, while Medline 20 was used as internal test data.[6]

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus

[3] https://github.com/fyvo/CochraneTranslations/

[4] https://md.taus.net/corona

[5] http://eval.covid19-mlia.eu/task3/

[6] These testsets were sentence-aligned with in-house

2.1.2 Monolingual sources

The back-translation of monolingual sources has often been effectively used to cater for parallel corpus shortage in the Biomedical domain in (Stojanovski et al., 2019; Peng et al., 2019). We also adopt this approach here.

Supplementary French data from three monolingual sources were collected from public archives: abstracts of medical papers published by Elsevier from the Lissa portal[7] and from the national ISTEX archive[8]; a collection of research articles collected from various sources[9] henceforth referred to as Med_Fr (Maniez, 2009). These documents were automatically translated into English with an NMT system trained on biomedical corpora, with a BLEU score of 33.6 on Medline20 testset.

The English side of Medline German and Spanish corpora is used as supplementary English data for back translation. Duplicate documents were removed based on the document id. For these, the internal structure of documents is often available and has been tagged as for the parallel data. These texts were then split into sentences[10] and translated into French using a NMT system trained on all Biomedical corpora with a BLEU score of 36.4 on Medline20 testset. All back-translated data is tagged using the proposal of Caswell et al. (2019).

Parallel and monolingual data are further processed using SentencePiece (Kudo and Richardson, 2018) tokenisation and detokenisation scheme to segment texts into subword units using a vocabulary of 32K subwords. These units were learned on all the in-domain corpora.

2.2 Sentence tagging: a three-level scheme

2.2.1 Tagging domains and corpora

As explained above, our training data is diverse, comprising in-domain parallel, out-of-domain parallel, and in-domain monolingual that is automatically back-translated. Some are made of lists of isolated sentences, while others retain the document information. Even within the in-domain data, some texts precisely match the genre of the testset (scientific abstracts) - this is the case for instance of Medline and to a lesser extent, Cochrane; while others are more remote (eg. the Ufal collection, or the Mlia corpus). In order to reflect this diversity, we designed a three-level sentence tagging scheme that is used for the experiments in Section 2.4.2. These tags appear as prefix of each source sentence.

The first level of tags distinguishes between out-of-domain data (<G>), and in-domain data (tagged <M>). The second level of tag aims to distinguish between data sources, hence the use of one dedicated tag for each corpus, except for the monolingual data, which is simply tagged with <BT>.

2.2.2 Tagging sections within documents

The third level of annotation is indented to enhance the translation context with information regarding the position of a sentence within the abstract. The structure of scientific abstracts in the medical domain often obey the IMRAD structure, and the third tag aims to include this structural information as an additional document-level context. Document level information is necessary to model long-range dependencies between words, phrases, or sentences, or document parts. For a translation system, the ability to model the context may notably improve certain translation decisions, e.g. a better or most consistent lexical choice (Kuang et al., 2018) or a better translation of anaphoric pronouns (Voita et al., 2018; Bawden et al., 2019). A recent review of these themes is in (Maruf et al., 2021).

For this purpose, we further pre-processed 6 corpora containing scientific abstracts. These corpora had different subheadings and structures as given below, which were mapped to a restricted set of section tags listed in Table 2:

1. Medline and Scielo: Abstracts and sub headings often without title. We identified a total of 189 subheadings including spelling variations. Examples include: Presenting Concerns of the Patient, Sources of Information, Novel finding, Study Selection etc.

2. Edp: Abstracts and sub headings mostly contain titles. 45 subheadings where found, such as: Case report, Observation, Subjects and Methods, Commentary, Pedagogical objectives etc.

3. Cochrane: only 10 different subheadings were found, including: Abs selection criteria, abs search strategy, abs data collection, summary title etc.

The identification and standardization of subheading information was a tedious process, involving a lot of rule-based processed to take the variability of sub-headings into account. In order to reconstruct fully parallel versions with subheadings, we also had to reinsert explicit headings in

tools and are shared at https://github.com/fyvo/WMT-Biomed-Test.

[7] https://www.lissa.fr/dc/#env=lissa

[8] https://www.istex.fr/

[9] https://crtt.univ-lyon2.fr/les-corpus-medicaux-du-crtt-613310.kjsp

[10] https://pypi.org/project/sentence-splitter/

Title	<H1>
Introduction	<INT>
Objectives	<OBJ>
Material and Methods	<MaM>
Results	<RES>
Conclusion	<CON>

Table 2: Standardized section heading tags

the source or the target files. Also note that this information was not available for all abstracts. After preprocessing files for which the full subheading information was available, we obtained the 6 fully-tagged corpora (see statistics in Table 3). A similar process was used for test sets (see Table 4).

Corpus	Lines	En words	Fr words
Medline	34836	742891	920811
Edp	1682	34167	37508
Scielo (wmt16)	7088	163275	199829
Cochrane-Reference	123598	2741426	3308485
Cochrane-GooglePE	30866	685490	828436
Cochrane-PE	20693	468691	568262

Table 3: Document-aligned training corpora

Testset	en-fr	fr-en
medline20	735	580
medline18	321	347
medline19	493	469

Table 4: Number of test sentences after alignment

Finally, we also introduced a third tag in all other documents as follows: sentences within an abstract where tagged as <ABS>, while all remaining sentences from other corpora where simply tagged as "unspecified subheading" (<US>).

2.3 Translation framework

Our translation systems mostly used the basic Transformer models, while a few contrastive systems used the large version (Vaswani et al., 2017). They all rely on Facebook's seq-2-seq library (fairseq) (Ott et al., 2019) with parameters settings borrowed from `transformer_wmt_de_en`.[11]. The ReLU activation function was used in all encoder and decoder layers. We optimize with Adam (Kingma and Ba, 2015), set up with a maximum learning rate of 0.0005 and an inverse square root decay schedule, as well as 4000 warmup updates. We share the decoder input and output embedding matrices. Models are trained with mixed precision and a batch size of 4096 tokens on 4 V100 GPUs for 300k updates. Systems were trained until convergence based on the BLEU score on the development sets. Evaluation was performed using SacreBleu (Post, 2018). Scores are chosen based on the best score on the development set (Medline 18, 19) and the corresponding scores for that checkpoint are reported on Medline 20 test set.

For fine-tuned systems, the process starts with models trained to convergence, based on BLEU score on dev sets. Training then resumes using a selected portion of the training corpus using the same parameters and criterion as for the base systems. In our results corresponding systems are post-fixed with `*-ft`.

2.4 Results

We present our results for the two directions in two tables, Table 5 and 6, differentiating the normal versus the *tag-based* systems. Base systems are given on the left, ($\Rightarrow$) identifies the derived (fine-tuned) systems.

2.4.1 Regular MT systems

Results for the untagged systems are reported in Table 5 and are denoted by `X*`, with `E*` and `F*` representing the English to French and French to English systems respectively.

We first built baseline systems. `X0` denotes the systems built using only the in-domain data provided by the organizers. `X1` are our baseline systems built using all in-domain parallel data. We see good improvement in both directions amounting to 4.2 and 4.8 BLEU points, which is obtained by adding around $1M$ sentences of additional Cochrane and Taus corpora to the already available $3.4M$ sentences from WMT'20. This hints at the relevance of the additional in-domain parallel corpora used.

We used the `X1` systems as strong in domain baselines to study the effect of adding back-translated in domain data. These appear as `X2` and `X3` in Table 5. Adding around 0.8M French to English and around 0.2M English to French back translated sentences did not help as much as we were expecting. We saw similar results last year and increased the amount of back translations this

[11] https://fairseq.readthedocs.io/en/latest/models.html

ID	Train	ID	Sentences	Medline 20	ID	Sentences	Medline 20
			EN-FR			**FR-EN**	
X0	WMT biomed data	**E0**	3.4M	31.6	**F0**	3.4M	28.8
X1	All biomed	**E1**	4.5M	35.8	**F0**	4.5M	33.6
Back translations of monolingual data							
X2	X1 + BT	**E2**	5.3M	34.8	**E2**	4.7M	33.5
X3	X1 + BT-tag	**E3**	5.3M	**36.6**	**F3**	4.7M	32.4
Out of domain fine-tuned with in domain							
X4	outdomain⇒biomed	**E4**	40.5M	32.3	**F4**[3]	41M	**35.8**

Table 5: Results for systems using in-domain and out-of-domain corpora. Superscripts ^{*n} denote runs submitted

ID	Train	Sentences	Medline 20		
			<SUBHEAD>	<ABS>	<US>
			EN-FR		
TE1	Out+In	41.7M	36.2	**36.3**	**36.3**
TE2[1]	TE1⇒ftbiomedplusbt	47.2M	**38.7**	38.5	38.6
TE3	TE2⇒ftCocMed	48.0M	38.2	**38.4**	38.3
Transformer Large					
TE4	Out+In	41.7M	36.1	36.2	**36.3**
TE5[2]	TE4⇒ftbiomedplusbt	47.2M	38.4	**38.5**	38.2
			FR-EN		
TF1	Out+In	40.9M	**32.1**	32.0	**32.1**
Mixed baseline finetuned with in-domain					
TF2[1]	TF1⇒ftbiomedplusbt	46.4M	**35.7**	35.2	35.2
TF3[2]	TF2⇒ftCocMed	48.8M	**35.3**	34.9	34.8

Table 6: Results for systems with sentences tagged with our 3 level tagging scheme. Test sets are decoded 3 times, where the third tag is varied from the more specific (<SUBHEAD>) to the more generic (<US>). Superscripts ^{*n} denote the runs submitted.

year. `X3` denote systems built using the tagging scheme proposed by Caswell et al. (2019), where back translations are prefixed with the <BT> tag on the source side.

Indicating that a training sentence is back-translated allows the model to separate the helpful and harmful signal. This proved particularly true for English into French where adding tag to back translations improved the BLEU score by 0.8 points; but it was not helpful in the reverse direction where the amount of back translated data was may be too small (0.2M lines). back-translations as compared to the baseline corpora.

Finally, systems were built by initialising the parameters from huge out-of-domain corpora and later fine tuned on in-domain corpora (`X4`), where in-domain sub words learned from all the Biomedical data are used to segment the out-of-domain data. The initial systems were trained for 4 epochs on general domain WMT14 EN-FR corpora. The FR-EN system (`F4`) is the best system in this direction, reaching a BLEU score of 35.8.

2.4.2 Tagged Systems

As our 3-level tagging scheme, described in Section 2.2, is adding information about the domain of each sentence, we specifically focused on larger systems by using all the available in- and out-of-

domain corpora.

Results are summarized in Table 6 with TE* representing the Tagged English to French systems and TF* representing the French to English systems. TE1 is the baseline system for EN-FR built with all the available in domain and out-domain data. TE4 is the corresponding baseline using a Large Transformer[12]. We then fine-tune these systems with all the in-domain data including the back translations, these are represented by TE2 and TE5 respectively. This gives an appreciable gain of 2.5 and 2.3 BLEU points for Transformer and Transformer large systems. As we saw no major difference in scores for Transformer versus Transformer large, so we continue the rest of experiments with the simple Transformer architecture. Fine-tuning further with just abstracts from Cochrane and Medline did not yield any further improvement.

French to English results display similar trends. The baseline (TF1) using all available (in domain + out-of-domain) data tagged with our 3 level scheme yielded a BLEU score of 32.1. Fine-tuning it further with all in-domain data (TF2) gives an improvement of 3.6 BLEU points which does not improve further when fine-tuning continues with just Cochrane and Medline abstracts (TF3).

To measure whether the model learned document domain and/or sentence origin information, we tested by tagging the test set with three different tags in the third position, using either the exact subheading, or abstract or UnSpecified for sentences for which the sub-section is unknown. Table 6 reports the scores for the three cases. Though the difference in scores for the three cases is minute, in-domain systems{TE2, TE3, TE5} and {TF2, TF3} achieve their best results when the test set is tagged with the subheading or the abstract tag, typical feature of the biomedical corpora. Conversely, for out-of-domain systems {TE1, TE4, TF1}, the best scores are always for the test set tagged with <US>. This strongly hints that the system is using the extra-information provided by the tag. These observations need to be confirmed using other metrics, as BLEU may not properly reflect these differences.

For English to French direction we got better scores with the tagged systems, with the best system (TE2 = 38.7) achieving 2.1 BLEU points more than the best un-tagged system (E3 = 36.6). This was however not the case for French-English where both tagged and un-tagged systems had more or less similar scores.

[12]hidden size of 1024 and a feed forward size of 4096. Rest of the parameters same as for other systems.

		EN-FR		
E2	base+bt	34.8		
E3	base+bt-tag	36.6		
		<SUBHEAD>	<ABS>	<US>
TE	Indomain+bt	**37.3**	37.0	37.0
		FR-EN		
F2	base+bt	33.5		
F3	base+bt-tag	32.4		
		<SUBHEAD>	<ABS>	<US>
TF	Indomain+bt	**34.4**	34.4	34.4

Table 7: Comparison of our 3 level tagged systems with the corresponding untagged systems. Systems {E2,F2} are built by adding back-translated data to the baseline. In systems {E3,E3}, the added back-translated data start with <BT> tag. Systems {TE, TF} use our 3-level tagging scheme for all sentences.

Systems in Tables 5 and 6 have different baselines, thus to establish a fair comparison we report numbers for comparable systems in Table 7. Systems {E2,E3,F2,E3} are copied from Table 5, whereas {TE and TF} are the corresponding systems using our tagging scheme on the sole biomedical data. We see here a clear gain for French-English when we use a 3-level tagging scheme (TF) compared to just adding the <BT> tag (F3); results for the reverse direction are more even and having one or three tags does not make a difference.

2.5 Conclusion

In this section, we have presented our work for the biomedical task. We notably have tried to incorporate document origin and structure information and improve strong baseline systems that were using a wealth of in-domain and out-of-domain data. . Overall, our systems for this year are significantly better than last year's, even though the benefits of adding document structures as tags need to be confirmed by more experiments and analyses.

3 News translation task: De ↔ Fr

In the 2021 News translation task, we focused on the German-French language pair in which the participants are asked to build MT systems for News in the financial domain. In this section, we discuss details of our approach and the rationale behind it.

3.1 Unsupervised adaptation

As the training and development data do not contain domain information, the supervised domain adaptation paradigm is not suitable here. However, non-parametric adaptation (Bapna and Firat, 2019), example-based guided machine translation (Zhang et al., 2018), unsupervised domain adaptation (Farajian et al., 2017) or priming NMT (Xu et al., 2020; Pham et al., 2020) have showed promising results for this problem. These approaches retrieve translation examples that are similar to the input source sentence, and use them to guide the inference and to reproduce existing translations or to locally adapt the pre-trained NMT system to the input sentence.

Even though all of the approaches mentioned above have merits of their own, we decided to focus on computationally cheaper methods such as (Bulte and Tezcan, 2019; Xu et al., 2020) where the retrieved instances provide an extra conditioning context for the decoder. Pham et al. (2020) further improved these techniques by proposing to simultaneously prime the source and the target side of the retrieved examples (see Section 3.4.1) and has been our main source of inspiration.

3.2 Data and preprocessing

We use all available parallel data for De $\leftrightarrow$ Fr, with the exception of the ParaCrawl data, for training. We also use monolingual data to improve translation quality. For both languages, we choose Newscrawl 2020. We additionally use Newscrawl 2018 and 2019 French data at inference time to explore the ability of our priming model to make use of extra data. Details are in Section 3.4.2. We use *newstest2019* as development set and test our models on *newstest2020*.

We filter out sentence pairs with invalid language tag using `fasttext` language id model[13] (Bojanowski et al., 2017). We use Moses tools to normalize punctuation, to remove non-printing characters and to tokenize into words. The final parallel data contains 5.6M sentences.[14] We use a shared source-target vocabulary built with 40K Byte Pair Encoding (BPE) units using the `subword-nmt` implementation (Sennrich et al., 2016b).[15]

[13] https://dl.fbaipublicfiles.com/fasttext/supervised-models/lid.176.bin

[14] https://github.com/moses-smt/mosesdecoder

[15] https://github.com/rsennrich/subword-nmt

3.3 Baseline systems

We build our Transformer-based (Vaswani et al., 2017) systems using `fairseq`[16] (Ott et al., 2019). Our baseline system is a large Transformer with a hidden size of 1024 and a feedforward size of 4096. We optimize with Adam (Kingma and Ba, 2015), set up with a maximum learning rate of 0.0007 and an inverse square root decay schedule, as well as 4000 warmup updates. We tie the encoder and decoder input embedding matrices with the decoder output embedding matrix and we apply layer normalization before each block. Models are trained with mixed precision and a batch size of 4096 tokens on 4 V100 GPUs for 300k updates.

3.4 Submitted systems

3.4.1 Boosting NMT by similar translations

Our approach comprises 2 steps: similar translation retrieval and inference where the priming example is processed in forced-decoding mode.

The retrieval of relevant examples for a given source sentence is based on their distance in some high-dimensional numerical representation space. These representations are computed using the encoder of the baseline system (see Section3.3) so as to keep our systems in the "constrained" track, as the use of large pre-trained models such as BERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019), etc., was only allowed in unconstrained submissions. More precisely, for each sentence, we average the contextualized embeddings output at the last layer of the encoder. From the training dataset, we create a datastore of pairs (K, V) in which the key K is the sentence embedding of some source sentence $\mathbf{f}$ and the value is the sentence pair $(\mathbf{f}, \mathbf{e})$ whose source sentence is $\mathbf{f}$. For each query, we retrieve k keys ($k = 10$ in all experiments).

The similarity between two sentences is the cosine similarity and the retrieval of the nearest neighbor(s) is performed using FAISS (Johnson et al., 2017). In order to search through a large datastore, we divide it into shards containing at most 500K data points; we conduct the k nearest neighbor search on each shard, gather all the retrieved keys from all shards into a list and reduce it to the k nearest keys. Given an input sentence and the list of its k nearest neighbours, we append m ($m \leq k$) retrieved source sentences to the input sentence and initialize the target side by the concatenation of the

[16] https://github.com/pytorch/fairseq

m corresponding target sentences. We use a special token to separate sentences.

During training, we train the NMT model with two types of examples (with and without retrieval): this means that each training sample will occur twice, once with and once without priming. The former examples have the following format:

$$\mathbf{f}_1 * \cdots * \mathbf{f}_m || \mathbf{f}$$

$$\mathbf{e}_1 * \cdots * \mathbf{e}_m || \mathbf{e}$$

while the latter is presented as the original data.

During inference, we use the same format as for the source-side, while we initialize the decoder with the prefix $\mathbf{e}_1 * \cdots * \mathbf{e}_m ||$. We therefore call this initialization "force-decoding". The special tokens, which serve as joiners between the retrieved sentences and the source/target sentence, are carefully chosen so that they never occur in the real text to avoid ambiguity. As discussed in Pham et al. (2020), it is possible to concatenate several similar sentences i.e. use $m > 1$; we however only report results with $m = 1$, since our preliminary experiments did not show superior results with $m > 1$.

3.4.2 Monolingual retrieval

Pham et al. (2020) suggested that monolingual texts in the target language can also be helpful to inform the inference. To make use of monolingual data, we create pseudo translation pairs with back-translation to generate the missing source language side. For this step, we leverage the baseline NMT system in Section 3.3 for one direction to back-translate the monolingual target text in the inverse direction. We use Newscrawl 2020 as monolingual resource for both directions. The monolingual French data contains approximately 10M sentences while the German data is much larger. We randomly extract 10M sentences from the German monolingual data as the pseudo corpus. The back-translated corpora are added to the real parallel corpora to create a larger datastore for retrieval.

3.5 Evaluation

3.5.1 Priming and Back-translation

We mainly evaluate our method on the De→Fr direction. Results on both Newstest2019 and 2020 are in Table 8. Our priming model is able to improve for 0.4 BLEU on newstest2019. However, the same improvement is not observed for newstest2020. As indicated in Pham et al. (2020), monolingual back-translated data could be directly applied during inference without any additional training. We thus search similar sentences on both original and synthetic data for the test sets. As shown in Table 8 (+ bt inference), searching on synthetic data directly improves our results by 0.6 BLEU point on newstest2019.

Model	newstest2019	newstest2020
baseline	35.7	32.8
+ bt	37.5	33.7
+ tag	37.5	34.3
priming	34.6	33.2
+ bt inference	35.2	33.2
priming + bt	37.4	33.9
+ tag	36.9	34.1
+ min sim 0.85	37.5	34.3

Table 8: BLEU scores of models for De→Fr direction. Our best submitted system obtained a BLEU score of 28.1 on newstest2021.

Even though priming model could benefit from back-translated data at inference time, training with synthetic data has proven to be effective in many previous works (Sennrich et al., 2016a; Edunov et al., 2018; Ng et al., 2019). Therefore, we also experiment by adding back-translated data to the original data and retrain a translation model. Results (+ bt) demonstrate that training with synthetic data clearly improves the performance on both test sets. Caswell et al. (2019) reports that using explicit tags to distinguish original from back-translated data provides further gains; however in our experiments, tagging BT data was not very helpful.

Our model using priming with synthetic data was not able to surpass the baseline model trained with additional back-translated data. One possible reason is that similar sentences retrieved with low similarity scores may be too noisy, and therefore decrease the overall performance. Filtering out noisy similar sentences (with a threshold of 0.85)[17] help to further improve the performance and makes it our best system (+ min sim 0.85). This setting was used for our primary submission for both directions.

We directly apply the best settings found for De→Fr to the reverse direction (Fr→De) and report the corresponding results in Table 9.

[17]Thresholding the minimum similarity score is the result of a trade-off: using a high threshold selects good sentences for priming, at the risk of leaving many examples without any priming data, while a low threshold retrieves more examples, many of which are of poor quality. Our preliminary experiments showed that that 0.85 was a reasonable value.

Model	newstest2019	newstest2020
baseline	27.7	27.2
+ bt	32.4	32.9
+ tag	30.9	31.0
priming + bt	29.8	29.3
+ tag	29.5	29.6
+ min sim 0.85	30.4	30.1

Table 9: BLEU scores of models for Fr→De. Our best submitted system obtained a BLEU score of 37.2 on newstest2021.

3.5.2 Priming and domain adaptation

In this section, we try to assess the relationship between domain adaptation (DA) and priming, and question our initial assumption that priming performs some kind of unsupervised adaptation. Our test set for this part contains 1000 lines extracted from the European Central Bank (ECB) corpus, also available from OPUS website.

As an alternative to priming, we first consider a simple unsupervised domain adaptation technique, where we retrieve $k = 10$ most similar sentences for each test sample, yielding a corpus of $10 \times k$ sentences that we use to fine-tune for two epochs the baseline systems. Again, filtering based on a similarity scores helps to accumulate a smaller number of sentences that are closer to the test domain.

We then try to combine priming and fine-tuning in the following manner: for each test sentence, we use the k nearest examples $(\mathbf{f}_1, \mathbf{e}_1) \ldots (\mathbf{f}_k, \mathbf{e}_k)$ to derive k *domain-adaptation examples with priming* as follows: the first primes $\mathbf{f}_2$ with $\mathbf{f}_1$, the second $\mathbf{f}_3$ with $\mathbf{f}_2$, and so on, until finally $\mathbf{f}_1$ is primed with $\mathbf{f}_k$ (the target part is built accordingly). This corpus is used for fine-tuning, and decoding proceeds as before (with $\mathbf{f}_1$ as prime).

These approaches (priming, unsupervised DA, and priming+DA) are compared in Table 10. We first see that using back-translated data is detrimental to the BLEU score of the baseline system, an effect that might be due to the difference between News texts and ECB domain. We also see that unsupervised adaptation with highly similar sentences yields a small gain. Priming alone achieves the same result as the baseline, but can also benefit somewhat from unsupervised DA. Our best results are obtained when we mix the two strategies, only keeping highly similar sentences.

Model	ECB
baseline	26.7
baseline + bt + tag	25.9
+ FT min sim 0.7	26.3
+ FT min sim 0.8	26.1
priming + bt + tag	25.9
+ FT	25.6
+ FT min sim 0.7	26.3
+ FT min sim 0.8	26.0
priming + bt + tag + min sim 0.7	26.3
+ FT min sim 0.7	26.5
priming + bt + tag + min sim 0.8	26.3
+ FT min sim 0.8	26.3

Table 10: BLEU scores for De→Fr on ECB.

3.6 Conclusion

In this section, we have reported our attempt to perform domain adaptation through priming, a technique which uses sentences that are similar to the test instances to provide additional context in training and decoding. In our experiments with the translation of News between French and German, we had little success with this technique, even when using massive amounts of back-translated data to search for relevant primes. This suggests that priming is not so useful for "open" domains such as News (Pham et al., 2020), and should better be used for standardized types of texts that occur in more specialized domains. We also tried to compare unsupervised DA and priming, showing that, in our context, the former was yielding better results that than the latter and also proposed a promising way to combine these two complementary techniques.

4 Conclusion and outlook

In this paper, we have described the systems prepared for this year's participation to WMT shared tasks. For the biomedical track, most of our efforts have been invested in the development of high resource systems, trying to take the structure of medical abstracts into account. In the News task, we have explored ways to perform unsupervised domain adaptation using retrieval based techniques and back-translated data.

Acknowledgements

This work was made possible thanks to the Saclay-IA and the Jean ZAY computing platforms. It was granted access to the HPC resources of IDRIS under the allocation 2021-[AD011011580R1, AD011011270R1, AD011011717] made by

GENCI. The first author is funded through a regional grant from the “Région Ile de France”.

References

Sadaf Abdul Rauf, José Carlos Rosales Núñez, Minh Quang Pham, and François Yvon. 2020. LIMSI @ WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 803–812, Online. Association for Computational Linguistics.

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 European Chapter of the ACL (EACL 2009)*, pages 16–23, Athens, Greece. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

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.

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

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.

Julia Ive, Aurélien Max, François Yvon, and Philippe Ravaud. 2016. Diagnosing high-quality statistical machine translation using traces of post-edition operations. In *International Conference on Language Resources and Evaluation - Workshop on Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem (MT Eval 2016 2016)*, page 8, Portorož, Slovenia.

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, 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.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 596–606, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical*

Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

François Maniez. 2009. L'adjectif dénominal en langue de spécialité: étude du domaine de la médecine. *Revue française de linguistique appliquée*, 14(2):117–130.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A survey on document-level neural machine translation: Methods and evaluation. *ACM Comput. Surv.*, 54(2).

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'16)*, pages 2942–2948, Portorož, Slovenia. European Language Resources Association (ELRA).

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Peng, Jianfeng Liu, Liangyou Li, and Qun Liu. 2019. Huawei's NMT systems for the WMT 2019 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, Florence, Italy. Association for Computational Linguistics.

Minh Quang Pham, Jitao Xu, Josep Crego, François Yvon, and Jean Senellart. 2020. Priming neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 516–527, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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.

Luciana B Sollaci and Mauricio G Pereira. 2004. The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. *Journal of the Medical Library Association : JMLA*, 92(3):364–367.

Dario Stojanovski, Viktor Hangya, Matthias Huck, and Alexander Fraser. 2019. The LMU munich unsupervised machine translation system for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 393–399, Florence, Italy. 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.

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.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. 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 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

WeChat Neural Machine Translation Systems for WMT21

Xianfeng Zeng[1]*, Yijin Liu[12]*, Ernan Li[1]*, Qiu Ran[1]*, Fandong Meng[1]*, Peng Li[1], Jinan Xu[2], and Jie Zhou[1]
[1] Pattern Recognition Center, WeChat AI, Tencent Inc, China
[2] Beijing Jiaotong University, Beijing, China
{xianfzeng,yijinliu,cardli,soulcaptran,fandongmeng,patrickpli,withtomzhou}@tencent.com
jaxu@bjtu.edu.cn

Abstract

This paper introduces WeChat AI's participation in WMT 2021 shared news translation task on English→Chinese, English→Japanese, Japanese→English and English→German. Our systems are based on the Transformer (Vaswani et al., 2017) with several novel and effective variants. In our experiments, we employ data filtering, large-scale synthetic data generation (i.e., back-translation, knowledge distillation, forward-translation, iterative in-domain knowledge transfer), advanced fine-tuning approaches, and boosted Self-BLEU based model ensemble. Our **constrained** systems achieve 36.9, 46.9, 27.8 and 31.3 case-sensitive BLEU scores on English→Chinese, English→Japanese, Japanese→English and English→German, respectively. The BLEU scores of English→Chinese, English→Japanese and Japanese→English are the highest among all submissions, and that of English→German is the highest among all constrained submissions.

1 Introduction

We participate in the WMT 2021 shared news translation task in three language pairs and four language directions, English→Chinese, English↔Japanese, and English→German. In this year's translation tasks, we mainly improve the final ensemble model's performance by increasing the diversity of both the model architecture and the synthetic data, as well as optimizing the ensemble searching algorithm.

Diversity is a metric we are particularly interested in this year. To quantify the diversity among different models, we compute Self-BLEU (Zhu et al., 2018) from the translations of the models on the valid set. To be precise, we use the translation of one model as the hypothesis and the translations of other models as references to calculate an average BLEU score. A higher Self-BLEU means this model is less diverse.

For model architectures (Vaswani et al., 2017; Meng and Zhang, 2019; Yan et al., 2020), we exploit several novel Transformer variants to strengthen model performance and diversity. Besides the Pre-Norm Transformer, the Post-Norm Transformer is also used as one of our baselines this year. We adopt some novel initialization methods (Huang et al., 2020) to alleviate the gradient vanishing problem of the Post-Norm Transformer. We combine the Average Attention Transformer (AAN) (Zhang et al., 2018) and Multi-Head-Attention (Vaswani et al., 2017) to derive a series of effective and diverse model variants. Furthermore, Talking-Heads Attention (Shazeer et al., 2020) is introduced to the Transformer and shows a significant diversity from all the other variants.

For the synthetic data generation, we exploit the large-scale back-translation (Sennrich et al., 2016a) method to leverage the target-side monolingual data and the sequence-level knowledge distillation (Kim and Rush, 2016) to leverage the source-side of bilingual data. To use the source-side monolingual data, we explore forward-translation by ensemble models to get general domain synthetic data. We also use iterative in-domain knowledge transfer (Meng et al., 2020) to generate in-domain data. Furthermore, several data augmentation methods are applied to improve the model robustness, including different token-level noise and dynamic top-p sampling.

For training strategies, we mainly focus on scheduled sampling based on decoding steps (Liu et al., 2021b), the confidence-aware scheduled sampling (Mihaylova and Martins, 2019; Duckworth et al., 2019; Liu et al., 2021a), the target denoising (Meng et al., 2020) method and the Graduated Label Smoothing (Wang et al., 2020) for in-domain finetuning.

For model ensemble, we select high-potential

* Equal contribution.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 243–254
November 10–11, 2021.

candidate models based on two indicators, namely model performance (BLEU scores on valid set) and model diversity (Self-BLEU scores among all other models). Furthermore, we propose a search algorithm based on the Self-BLEU scores between the candidate models with selected models. We observed that this novel method can achieve the same BLEU score as the brute force search while saving approximately 95% of search time.

This paper is structured as follows: Sec. 2 describes our novel model architectures. We present the details of our systems and training strategies in Sec. 3. Experimental settings and results are shown in Sec. 4. We conduct analytical experiments in Sec. 5. Finally, we conclude our work in Sec. 6.

2 Model Architectures

In this section, we describe the model architectures used in the four translation directions, including several different variants for the Transformer (Vaswani et al., 2017) .

2.1 Model Configurations

Deeper and wider architectures are used this year since they show strong capacity as the number of parameters increases. In our experiments, we use multiple model configurations with 20/25-layer encoders for deeper models and the hidden size is set to 1024 for all models. Compared to our WMT20 models (Meng et al., 2020), we also increase the decoder depth from 6 to 8 and 10 as we find that gives a certain improvement, but deeper depths give limited performance gains. For the wider models, we adopt 8/12/15 encoder layers and 1024/2048 for hidden size. The filter sizes of models are set from 8192 to 15000. Note that all the above model configurations are applied to the following variant models.

2.2 Transformer with Different Layer-Norm

The Transformer (Vaswani et al., 2017) with Pre-Norm (Xiong et al., 2020) is a widely used architecture in machine translation. It is also our baseline model as its performance and training stability is better than the Post-Norm counterpart.

Recent studies (Liu et al., 2020; Huang et al., 2020) show that the unstable training problem of Post-Norm Transformer can be mitigated by modifying initialization of the network and the successfully converged Post-Norm models generally outperform Pre-Norm counterparts. We adopt these initialization methods (Huang et al., 2020) to our training flows to stabilize the training of deep Post-Norm Transformer. Our experiments have shown that the Post-Norm model has a good diversity compared to the Pre-Norm Model and slightly outperform the Pre-Norm Model. We will further analyze the model diversity of different variants in Sec. 5.1.

2.3 Average Attention Transformer

We also use Average Attention Transformer (AAN) (Zhang et al., 2018) as we used last year to introduce more model diversity. In the Average Attention Transformer, a fast and straightforward average attention is utilized to replace the self-attention module in the decoder with almost no performance loss. The context representation g_i for each input embedding is as follows:

$$g_i = FFN(\frac{1}{i}\sum_{k=1}^{i} y_k) \quad (1)$$

where y_k is the input embedding for step k and i is the current time step. $FFN(\cdot)$ denotes the position-wise feed-forward network proposed by Vaswani et al. (2017).

In our preliminary experiments, we observe that the Self-BLEU (Zhu et al., 2018) scores between AAN and Transformer are lower than the scores between the Transformer with different configurations.

2.4 Weighted Attention Transformer

We further explore three weighting strategies to improve the modeling of history information from previous positions in AAN. Compared to the average weight across all positions, we try three methods including decreasing weights with position increasing, learnable weights and exponential weights. In our experiments, We observe exponential weights perform best among all these strategies. The exponential weights context representation g_i is calculated as follows:

$$c_i = (1-\alpha)y_i + \alpha \cdot c_{i-1} \quad (2)$$

$$g_i = FFN(c_i) \quad (3)$$

where α is a tuned parameter. In our previous experiments, we test different alpha, including 0.3, 0.5, and 0.7, on the valid set and we set the alpha to 0.7 in all subsequent experiments as it slightly outperform the others.

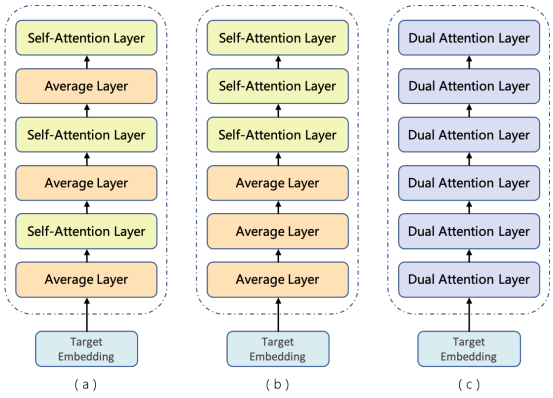


Figure 1: Mixed-AAN Transformers.

2.5 Mixed-AAN Transformers

Our preliminary experiments show that the decoder structure is strongly related to the model diversity in the Transformer. Therefore, we propose to stack different types of decoder layers to derive different Transformer variants. As shown in Figure 1, we mainly adopt three Mixed-AAN Transformer architectures: a) Alternately mixing the standard self-attention layer and the average attention layer, b) Continuously stacking several average attention layers on the bottom layers and then stacking self-attention layers for the rest layers. c) Stacking both the self-attention layer and average attention layer at each layer and using their average sum to form the final hidden states (named as 'dual attention layer').

In the experiments, Mixed-AAN not only performs better but also shows strong diversity compared to the vanilla Transformer. With four Mixed-AAN models, we reach a better ensemble result than the result with ten models which consist of deeper and wider standard Transformer. We will further analyze the effects of different architectures from performance, diversity, and model ensemble in Sec. 5.1

2.6 Talking-Heads Attention

In Multi-Head Attention, the different attention heads perform separate computations, which are then summed at the end. Talking-Heads Attention (Shazeer et al., 2020) is a new variation that inserts two additional learned linear projection weights, W_l and W_w, to transform the attention-logits and the attention scores respectively, moving information across attention heads. The calculation formula is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}W_l)W_wV \quad (4)$$

We adopt this method in both encoders and decoders to improve information interaction between attention heads. This approach shows the most remarkable diversity among all the above variants with only a slight performance loss.

3 System Overview

In this section, we describe our system used in the WMT 2021 news shared task. We depicts the overview of our NMT system in Figure 2, which can be divided into four parts, namely data filtering, large-scale synthetic data generation, in-domain finetuning, and ensemble. The synthetic data generation part further includes the generation of general domain and in-domain data. Next, we proceed to illustrate these four parts.

3.1 Data Filtering

We filter the bilingual training corpus with the following rules for most language pairs:

- Normalize punctuation with Moses scripts except Japanese data.
- Filter out the sentences longer than 100 words or exceed 40 characters in a single word.
- Filter out the duplicated sentence pairs.
- The word ratio between the source and the target words must not exceed 1:4 or 4:1.
- Filter out the sentences where the fast-text result does not match the origin language.
- Filter out the sentences that have invalid Unicode characters.

Besides these rules, we filter out sentence pairs in which Chinese sentence has English characters in En-Zh parallel data. The monolingual corpus is also filtered with the n-gram language model trained by the bilingual training data for each language. All the above rules are applied to synthetic parallel data.

3.2 General Domain Synthetic Data Generation

In this section, we describe our techniques for constructing general domain synthetic data. The general domain synthetic data is generated via large-scale back-translation, forward-translation and knowledge distillation to enhance the models' performance for all domains. Then, we exploit

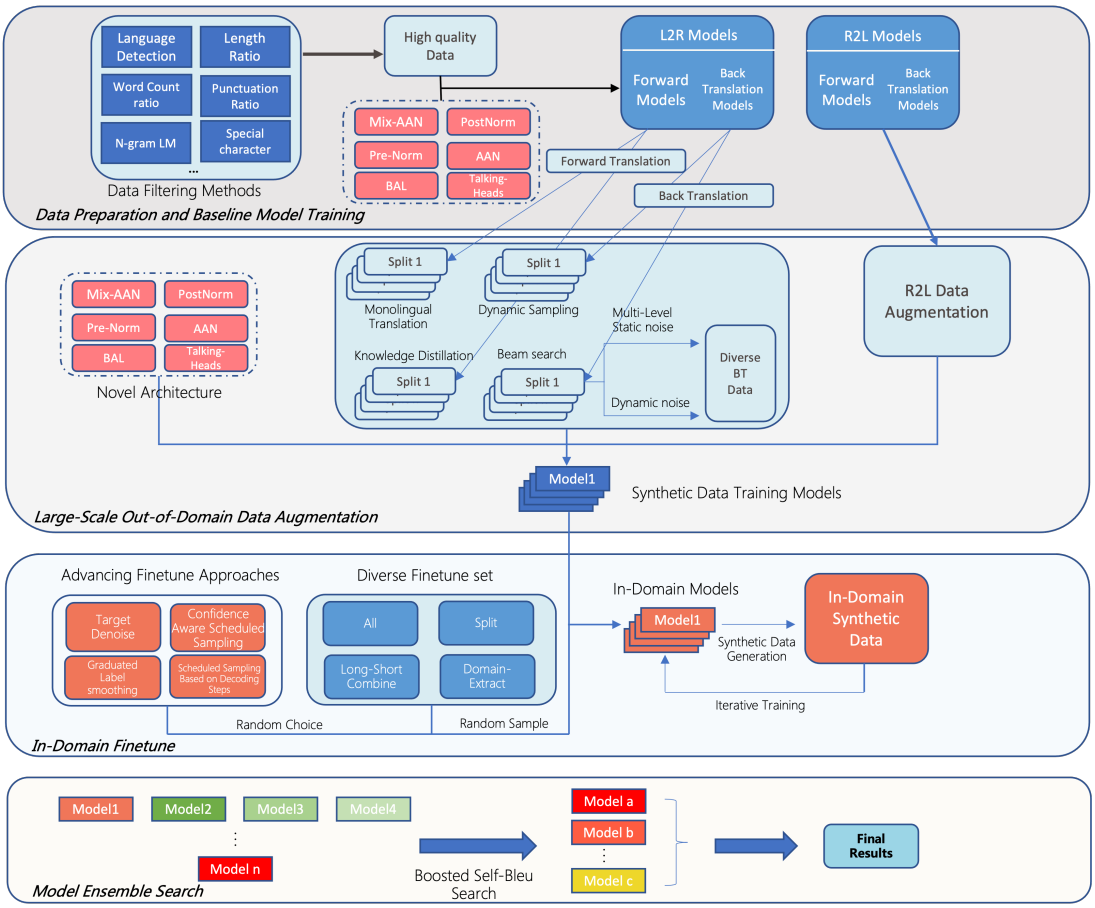


Figure 2: Architecture of our NMT system.

the iterative in-domain knowledge transfer (Meng et al., 2020) in Sec 3.3, which transfers in-domain knowledge to the vast source-side monolingual corpus, and builds our in-domain synthetic data. In the following sections, we elaborate the above techniques in detail.

3.2.1 Large-scale Back-Translation

Back-translation is the most commonly used data augmentation technique to incorporate the target side monolingual data into NMT (Hoang et al., 2018). Previous work (Edunov et al., 2018) has shown that different methods of generating pseudo corpus has a different influence on translation quality. Following these works, we attempt several generating strategies as follows:

- Beam Search: Generate target translation by beam search with beam size 5.
- Sampling Top-K: Select a word randomly from top-K (K is set to 10) words at each decoding step.
- Dynamic Sampling Top-p: Selected a word at each decoding step from the smallest set whose cumulative probability mass exceeds p and the p is dynamically changing from 0.9 to 0.95 during data generation.

Note that we also use Tagged Back-Translation (Caswell et al., 2019) in En→De and Right-to-Left (R2L) back-translation in En↔Ja, as we achieve a better BLEU score after using these methods.

3.2.2 Knowledge Distillation

Knowledge Distillation (KD) has proven to be a powerful technique for NMT (Kim and Rush, 2016; Wang et al., 2021) to transfer knowledge from the teacher model to student models. In particular, we first use the teacher models to generate synthetic corpus in the forward direction (i.e., En→Zh). Then, we use the generated corpus to train our student models.

Notably, Right-to-Left (R2L) knowledge distillation is a good complement to the Left-to-Right (L2R) way and can further improve model performance.

3.2.3 Forward-Translation

Using monolingual data from the source language to further enhance the performance and robustness of the model is also an effective approach. We use the ensemble model to generate high quality forward-translation data and obtain a stable improvement in En→Zh and En→De directions.

3.3 Iterative In-domain Knowledge Transfer

Since in-domain knowledge transfer (Meng et al., 2020) delivered a massive performance boost last year, we still use this technique in En↔Ja and En→De this year. It is not applied to En→Zh because no significant improvement is observed. We guess the reason is that the in-domain finetuning in the En→Zh direction does not bring a significant improvement compared to the other directions. And in-domain knowledge transfer is aiming at enhancing the effect of finetuning, so this does not have a noticeable effect in the English-Chinese direction.

We first use normal finetuning in Sec. 3.5 to equip our models with in-domain knowledge. Then, we ensemble these models to translate the source monolingual data into the target language. We use 4 models with different architectures and training data as our ensemble model. Next, we combine the source language sentences with the generated in-domain target language sentences as pseudo-parallel corpus. Afterwards, we retrain our models with both in-domain pseudo-parallel data and general domain synthetic data.

3.4 Data Augmentation

Once the pseudo-data is constructed, we further obtain diverse data by adding different noise. Compared to previous years' WMT competitions, we implement a multi-level static noise approach for our pseudo corpus:

- Token-level: Noise on every single subword after byte pair encoding.
- Word-level: Noise on every single word before byte pair encoding.
- Span-level: Noise on a continuous sequence of tokens before byte pair encoding.

The different granularities of noise make the data more diverse. The noise types are random replacement, random deletion and random permutation. We apply the three noise types in a parallel way for each sentence. The probability of enabling each of the three operations is 0.2.

Furthermore, an on-the-fly noise approach is applied to the synthetic data. By using on-the-fly noise, the model is trained with different noises in every epoch rather than all the same along this training stage.

3.5 In-domain Finetuning

A domain mismatch exists between the obtained system trained with large-scale general domain data and the target test set. To alleviate this mismatch, we finetune these convergent models on small scale in-domain data, which is widely used for domain adaption (Luong and Manning, 2015; Li et al., 2019). We take the previous test sets as in-domain data and extract documents that are originally created in the source language for each translation direction (Sun et al., 2019). We also explore several advanced finetuning approaches to strengthen the effects of domain adaption and ease the exposure bias issue, which is more serious under domain shift.

Target Denoising (Meng et al., 2020). In the training stage, the model never sees its own errors. Thus the model trained with teacher-forcing is prune to accumulated errors in testing (Ranzato et al., 2016). To mitigate this training-generation discrepancy, we add noisy perturbations into decoder inputs when finetuning. Thus the model becomes more robust to prediction errors by target denoising. Specifically, the finetuning data generator chooses 30% of sentence pairs to add noise, and keeps the remaining 70% of sentence pairs unchanged. For a chosen pair, we keep the source sentence unchanged, and replace the i-th token of the target sentence with (1) a random token of the current target sentence 15% of the time (2) the unchanged i-th token 85% of the time.

Graduated Label-smoothing (Wang et al., 2020). Finetuning on a small scale in-domain data can easily lead to the over-fitting phenomenon which is harmful to the model ensemble. It generally appears as the model over confidently outputting similar words. To further preventing overfitting of in-domain finetuning, we apply the Graduated Label-smoothing approach, which assigns a higher smoothing penalty for high-confidence predictions, during in-domain finetuning. Concretely, following the paper's setting, we set the smoothing penalty to 0.3 for tokens with confidence above 0.7,

zero for tokens with confidence below 0.3, and 0.1 for the remaining tokens.

Confidence-Aware Scheduled Sampling. Vanilla scheduled sampling (Zhang et al., 2019) simulates the inference scene by randomly replacing golden target input tokens with predicted ones during training. However, its critical schedule strategies are only based on training steps, ignoring the real-time model competence. To address this issue, we propose confidence-aware scheduled sampling (Liu et al., 2021a), which quantifies real-time model competence by the confidence of model predictions. At the t-th target token position, we calculate the model confidence $conf(t)$ as follow:

$$conf(t) = P(y_t|\mathbf{y}_{<\mathbf{t}}, \mathbf{X}, \theta) \tag{5}$$

Next, we design fine-grained schedule strategies based on the model competence. The fine-grained schedule strategy is conducted at all decoding steps simultaneously:

$$y_{t-1} = \begin{cases} y_{t-1} & if\ conf(t) \leq t_{golden} \\ \hat{y}_{t-1} & else \end{cases} \tag{6}$$

where t_{golden} is a threshold to measure whether $conf(t)$ is high enough ($e.g.$, 0.9) to sample the predicted token $\hat{y}_{t-1}$.

We further sample more noisy tokens at high-confidence token positions, which prevents scheduled sampling from degenerating into the teacher forcing mode.

$$y_{t-1} = \begin{cases} y_{t-1} & if\ conf(t) \leq t_{golden} \\ \hat{y}_{t-1} & if\ t_{golden} < conf(t) \leq t_{rand} \\ y_{rand} & if\ conf(t) > t_{rand} \end{cases} \tag{7}$$

where t_{rand} is a threshold to measure whether $conf(t)$ is high enough ($e.g.$, 0.95) to sample the random target token $\hat{y}_{rand}$.

Scheduled Sampling Based on Decoding Steps. We propose scheduled sampling methods based on decoding steps from the perspective of simulating the distribution of real translation errors (Liu et al., 2021b). Namely, we gradually increase the selection probability of predicted tokens with the growth of the index of decoded tokens. At the t-th decoding step, the probability of sampling golden tokens $g(t)$ is calculated as follow:

Algorithm 1 Boosted Self-BLEU based Ensemble

Input:
List of candidate models M = $\{m_0, ..., m_n\}$
Valid set BLEU for each model B = $\{b_i, ..., b_n\}$
Average Self-BLEU for each model S = $\{s_i, ..., s_n\}$
The number of models n
The number of ensemble models c

Output: Model combinations C

for $i \leftarrow 1\ to\ n$ **do**
  $score_i = (b_i - min(B)) \cdot weight + (max(S) - s_i)$
  $weight = \frac{(max(S) - min(S))}{(max(B) - min(B))}$
end for
Add the highest score model to candidates list C = { m_{top} }
while $|C| < c$ **do**
  $index = \arg\min_i \frac{1}{|M-C|} \sum_{i \in M-C, j \in C} BLEU(i, j)$
  Add m_{index} to candidate list C
end while
return C

- Linear Decay: $g(t) = \max(\epsilon, kt + b)$, where ϵ is the minimum value, and $k < 0$ and b is respectively the slope and offset of the decay.
- Exponential Decay: $g(t) = k^t$, where $k < 1$ is the radix to adjust the decay.
- Inverse Sigmoid Decay: $g(t) = \frac{k}{k + e^{\frac{t}{k}}}$, where e is the mathematical constant, and $k \geq 1$ is a hyperparameter to adjust the decay.

Following our preliminary conclusions (Liu et al., 2021b), we choose the exponential decay and set k to 0.99 by default.

3.6 Boosted Self-BLEU based Ensemble (BSBE)

After we get numerous finetuned models, we need to search for the best combination for ensemble model. Ordinary random or greedy search is over-simplified to search for a good model combination and enumerate over all combinations of candidate models is inefficient. The Self-BLEU based pruning strategy (Meng et al., 2020) we proposed in last year's competition achieve definite improvements over the ordinary ensemble.

However, diversity is not the only feature we need to consider but the performance in the valid

set is also an important metric. Therefore, we combine Self-BLEU and valid set BLEU together to derive a Boosted Self-BLEU-based Ensemble (BSBE) algorithm. Then, we apply a greedy search strategy in the top N ranked models to find the best ensemble models.

See algorithm 1 for the pseudo-code. The algorithm takes as input a list of n strong single models M, BLEU scores on valid set for each model B, average Self-BLEU scores for each model S, the number of models n and the number of ensemble models c. The algorithm return a list C consists of selected models. We calculate the weighted score for each model as line 2 in the pseudo-code. The weight calculated in line 3 is a factor to balance the scale of Self-BLUE and valid set BLEU. Then the list C initially contains the model m_{top} has a highest weighted score. Next, we iteratively re-compute the average Self-BLEU between the remaining models in $|M - C|$ and selected models in C, based on which we select the model has minimum Self-BLEU score into C.

In our experiments, we save around 95% searching time by using this novel method to achieve the same BLEU score of the Brute Force search. We will further analyze the effect of Boosted Self-BLEU based Ensemble in section 5.2.

4 Experiments And Results

4.1 Settings

The implementation of our models is based on Fairseq[1] for En→Zh and EN→De, and OpenNMT[2] for En↔Ja. All the single models are carried out on 8 NVIDIA V100 GPUs, each of which has 32 GB memory. We use the Adam optimizer with β_1 = 0.9, β_2 = 0.998. The gradient accumulation is used due to the high GPU memory consumption. The batch size is set to 8192 tokens per GPU and we set the "update-freq" parameter in Fairseq to 2. The learning rate is set to 0.0005 for Fairseq and 2.0 for OpenNMT. We use warmup step = 4000. We calculate sacreBLEU[3] score for all experiments which is officially recommended.

4.2 Dataset

The statistics of all training data is shown in Table 1. For each language pair, the bilingual data is the combination of all parallel data released by WMT21. For monolingual data, we select data from News Crawl, Common Crawl and Extended Common Crawl, it is then divided into several parts, each containing 50M sentences.

	En→Zh	En→De	En↔Ja
Bilingual Data	30.7M	74.8M	12.3M
Source Mono Data	200.5M	332.8M	210.8M
Target Mono Data	405.2M	237.9M	354.7M

Table 1: Statistics of all training data.

For general domain synthetic data, we use all the target monolingual data to generate back-translation data and a part of source monolingual data (about 80 to 100 million for different languages) to get forward translation data. For the in-domain pseudo-parallel data, we use the entire source monolingual data and bilingual data. All the test and valid data from previous years are used as in-domain data.

We use the methods described in Sec. 3.1 to filter bilingual and monolingual data.

4.3 Pre-processing and Post-processing

English and German sentences are segmented by Moses[4], while Japanese use Mecab[5] for segmentation. We segment the Chinese sentences with an in-house word segmentation tool. We apply punctuation normalization in English, German and Chinese data. Truecasing is applied to English↔Japanese and English→German. We use byte pair encoding BPE (Sennrich et al., 2016b) with 32K operations for all the languages.

For the post-processing, we apply de-truecaseing and de-tokenizing on the English and German translations with the scripts provided in Moses. For the Chinese translations, we transpose the punctuations to the Chinese format.

4.4 English→Chinese

The results of En→Zh on newstest2020 are shown in Table 2. For the En→Zh task, filtering out part of sentence pairs containing English characters in Chinese sentences shows a significant improvement in the valid set. After applying large-scale Back-Translation, we obtain +2.0 BLEU score on the baseline. We further gain +0.62 BLEU score after applying knowledge distillation and +0.24 BLEU from Forward-Translation. Surprisingly, we observe that adding more BT data from different

[1]https://github.com/pytorch/fairseq
[2]https://github.com/OpenNMT/OpenNMT-py
[3]https://github.com/mjpost/sacrebleu
[4]http://www.statmt.org/moses/
[5]https://github.com/taku910/mecab

SYSTEM	En→Zh	En→Ja	Ja→En	En→De
Baseline	44.53	35.78	19.71	33.28
+ Back Translation	46.52	36.12	20.82	35.28
+ Knowledge Distillation	47.14	36.66	21.63	36.38
+ Forward Translation	47.38	–	–	36.78
+ Mix BT	48.17	37.22	22.11	–
+ *Finetune*	49.81	42.54	25.91	39.21
+ *Advanced Finetune*	50.20	–	–	39.56
+ 1st In-domain Knowledge Transfer	–	40.32	24.49	39.23
+ *Finetune*	–	43.66	26.24	–
+ *Advanced Finetune*	–	–	–	39.87
+ 2nd In-domain Knowledge Transfer	–	43.69	25.89	–
+ *Finetune*	–	44.23	26.27	–
+ *Advanced Finetune*	–	44.42	26.38	–
+ Normal Ensemble	50.57	45.11	28.01	40.42
+ BSBE	**50.94** ⋆	**45.35** ⋆	**28.24** ⋆	40.59
+ Post-Process	–	–	–	**41.88** ⋆

Table 2: Case-sensitive BLEU scores (%) on the four directions *newstest2020*, where '⋆' denotes the submitted system. Mix BT means we use multiple parts of Back Translation data with different generation strategies. The *Advanced Finetune* methods outperform the normal *Finetune* and we report the best results in single model. BSBE outperforms Normal Ensemble in all four directions.

shards with different generation strategy can further boost the model performance to 48.17. The fine-tuned model achieves a 49.81 BLEU score, which demonstrates that the domain of the training corpus is apart from the test set domain. The *advanced* finetuning further brings about 0.41 BLEU score gains compared to normal finetune. Our best single model achieves a 50.22 BLEU score.

In preliminary experiments, we select the best performing models as our ensemble combinations obtaining +0.4 BLEU score. On top of that, even after searching hundreds of models, no better results are obtained. With BSBE strategies in Sec. 3.6, a better model combination with less number of models are quickly searched, and we finally achieve 50.94 BLEU score. Our WMT2021 English→Chinese submission achieves a SacreBLEU score of 36.9, which is the highest among all submissions and chrF score of 0.337.

4.5 English→Japanese

The results of En→Ja on newstest2020 are shown in Table 2. For the En→Ja task, we filter out the sentence pairs containing Japanese characters in the English side and vice versa. The Back-Translation and Knowledge Distillation improve the baseline from 35.78 to 36.66. Adding more BT data further brings in 0.56 improvements. The improvement by finetuning is much larger than other directions, which is 5.32 BLEU. We speculate that this is because there is less bilingual data for English and Japanese than for other languages, and the test results for Japanese are char level BLEU so this direction is more influenced by the in-domain finetuning. Two In-domain knowledge transfers improve BLEU score from 37.22 to 43.69. Normal finetune still provides 0.54 improvements after in-domain knowledge transfer. Then, we apply *advanced* finetuning methods to further get 0.19 BLEU improvements. Our final ensemble result outperforms baseline 9.57 BLEU.

4.6 Japanese→English

The Ja→En task follows the same training procedure as En→Ja. From Table 2, we can observe that Back-Translation can provide 1.11 BLEU improvements from baseline. Knowledge Distillation and more BT data can improve the BLEU score from 20.82 to 22.11. The finetuning improvement is 3.8 which is slightly less than the En→Ja direction but still larger than En→Zh and En→De. We also apply two-turn in-domain knowledge transfer and further boost the BLEU score to 25.89. After normal finetuning, the BLEU score achieves 26.27. The *advanced* finetuning methods provide a slight improvement on Ja→En. After ensemble, we achieve 28.24 BLEU in newstest2020.

MODEL	EN-ZH	EN-JA	JA-EN	EN-DE
Transformer	49.92	44.27	26.12	39.76
Transformer with Post-Norm	49.97	-	-	-
Average Attention Transformer	49.91	44.38	26.31	39.62
Weighted Attention Transformer	49.99	-	-	39.74
Average First Transformer ∗	50.14	**44.42**	26.37	**39.87**
Average Bottom Transformer ∗	50.10	44.36	**26.38**	39.77
Dual Attention Transformer ∗	**50.20**	-	-	**39.87**
Talking-Heads Attention	49.89	-	-	39.70

Table 3: Case-sensitive BLEU scores (%) on the four translation directions *newstest2020* for different architecture. The model with '∗' is the Mixed-AAN variants. The **bolded** scores correspond to the best single model scores in Table 2.

MODEL	Transformer	Post-Norm	AAN	Weighted	Avg-First	Self-First	Dual	TH
Transformer	100	78.12	76.02	75.08	74.47	74.02	73.51	72.63
Post-Norm	78.12	100	76.12	75.10	74.33	74.05	73.45	72.59
AAN	76.02	76.12	100	79.24	74.81	74.97	73.43	72.13
Weighted	75.08	75.10	79.24	100	74.72	74.93	73.55	72.21
Avg-First ∗	74.46	74.33	74.81	74.72	100	75.25	74.28	72.25
Avg-Bot ∗	74.02	74.05	74.97	74.93	75.25	100	74.21	72.33
Dual ∗	73.51	73.45	73.43	73.55	74.28	74.21	100	72.23
TH	**72.63**	**72.59**	**72.13**	**72.21**	**72.25**	**72.33**	**72.23**	100

Table 4: Self-BLEU scores (%) between different architectures. For simplicity, we refer to these models as Transformer (Pre-Norm Transformer), Post-Norm (Post-Norm Transformer), AAN (Average Attention Transformer), Weighted (Weighted Attention Transformer), Avg-First (Average First Transfromer), Avg-Bot (Average Bottom Transformer), Dual (Dual Attention Transformer), TH (Talking-Heads Attention). The model with '∗' is the Mix-AAN variants.

4.7 English→German

The results of En→De on newstest2020 are shown in Table 2. After adding back-translation, we improve the BLEU score from 33.28 to 35.28. Knowledge Distillation further boosts the BLEU score to 36.58. The finetuning further brings in 2.63 improvements. After injecting the in-domain knowledge into the monolingual corpus, we get another 0.31 BLEU gain. We apply a post-processing procedure on En→De. Specifically, we normalize the English quotations to German ones in German hypotheses, which brings in 1.3 BLEU improvements.

5 Analysis

To verify the effectiveness of our approach, we conduct analytical experiments on model variants, finetune methods, and ensemble strategies in this section.

5.1 Effects of Model Architecture

We conduct several experiments to validate the effectiveness of Transformer (Vaswani et al., 2017) variants we used and list results in Table 3. We also investigate the diversity of different variants and the impacts on the model ensemble. The results is listed in Table 4 and Table 5. Here we take En→Zh models as examples to conduct the diversity and ensemble experiments. The results in other directions show similar trends.

Performance. As shown in Table 3, AAN performs slightly worse than other variants in En→Zh but Mixed-AAN variants outperform normal Transformer. Weighted Attention Transformer provides noticeable improvement compare to AAN and sometimes better than vanilla Transformer.

Diversity. The Self-BLEU scores in Table 4 demonstrate the difference between two models, more different models generally have lower scores. As we can see, AAN and all the variants with AAN have an absolutely lower Self-BLEU score with the Transformer. The Talking-Heads Attention has the minimum scores among all the variants.

Ensemble. In our preliminary experiments, we observe that more diverse models can significantly help the model ensemble. The results are listed

MODELS	newstest2020
Deeper & Wider Transformer	50.31
Weighted & Mixed-AAN	50.44
Ensemble with all models above	**50.62**

Table 5: Ensemble results with different architectures. The first row is the ensemble results with 10 deeper and wider models searched from dozens of ones. The second row is the ensemble results with only 4 Weighted Attention Transformer and Mixed-AAN models.

ALGORITHM	BLEU	Number of Searches
Greedy	50.19	$2n$
Brute Force	50.44	$\sum_{i=1}^{n} C_n^i$
BSBE	**50.44**	$n+1$

Table 6: Results of different search algorithm. n is the total number of models used for the search. The number of searches is number that the methods need to translate the valid set. Our BSBE achieves comparable BLEU score as Brute Force search and significantly reduces the searching time.

in Table 5. We get a more robust ensemble model with only four models using our novel variants than searching from dozens of Deeper and Wider Transformer models. Even these four models are trained with the same training data. After we combine the four models with Deeper and Wider Transformer, we can further get a significant improvement.

Take En→Zh as an example, our final submission consist of 1 Average First Transformer, 1 Average Bottom Transformer, 1 Dual Attention Transformer, 1 Weighted Attention Transformer and 1 Transformer with Post-Norm.

5.2 Effects of Boosted Self-BLEU based Ensemble

To verify the superiority of our Boosted Self-BLEU based Ensemble (BSBE) method, we randomly select 10 models with different architecture and training data. For our submitted system, we search from over 500 models. We use a greedy search algorithm (Deng et al., 2018) as our baseline. The greedy search greedily selects the best performance model into candidate ensemble models. If the selected model provides a positive improvement, we keep it in the candidates. Otherwise, it is added to a temporary model list and still has a weak chance to be reused in the future. One model from the temporary list can be reused once, after which it is withdrawn definitely. We compare the results of greedy search, BSBE and Brute Force and list the ensemble model BLEU and the number of searches in Table 6. Note that n is the number of models, which is 10 here. For BSBE, we need to get the translation result of every model to calculate the Self-BLEU. After that, we only need to perform the inference process once.

5.3 Effects of Advanced Finetuning

In this section, we describe our experiments on advanced finetuning in the four translation directions. As shown in Table 7, all the advanced finetuning methods outperform normal finetuning. For En→Zh, Scheldule Sampling Based on Decoding Steps with Graduated Label Smoothing improves the model performance from 49.81 to 50.20. For En↔Ja, Target Denoising with Graduated Label Smoothing provides the highest BLEU gain, which are 0.19 and 0.11. For the En→De direction, Confidence-Aware Scheldule Sampling with Graduated Label Smoothing performs the best, improving from 39.21 to 39.42. These findings are in line with the conclusion of Wang and Sennrich (2020) that links exposure bias with domain shift.

6 Conclusion

We investigate various novel Transformer based architectures to build robust systems. Our systems are also built on several popular data augmentation methods such as back-translation, knowledge distillation and iterative in-domain knowledge transfer. We enhance our system with advanced finetuning approaches, i.e., target denoising, graduated label smoothing and confidence-aware scheduled sampling. A boosted Self-BLEU based model ensemble is also employed which plays a key role in our systems. Our constrained systems achieve 36.9, 46.9, 27.8 and 31.3 case-sensitive BLEU scores on English→Chinese, English→Japanese, Japanese→English and English→German, respectively. The BLEU scores of English→Chinese, English→Japanese and Japanese→English are the highest among all submissions, and that of English→German is the highest among all constrained submissions.

Acknowledgements

Yijin Liu and Jinan Xu have been supported by the National Key R&D Program of China (2020AAA0108001) and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors

FINETUNING APPROACH	EN-ZH	EN-JA	JA-EN	EN-DE
Normal	49.81	44.23	26.27	39.21
Graduated Label Smoothing	49.95	44.32	26.35	39.32
+ *Target Denoising*	50.09	**44.42**	**26.38**	39.34
+ *Confidence-Aware Scheldule Sampling*	50.17	44.35	26.33	**39.42**
+ *Scheldule Sampling Based on Decoding Steps*	**50.20**	44.36	26.33	39.40

Table 7: Case-sensitive BLEU scores (%) on the four translation directions *newstest2020* for different finetuning approaches. We report the highest score and bold the best result among different finetuning approaches.

would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. 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: Shared Task Papers*, pages 368–376, Belgium, Brussels. Association for Computational Linguistics.

Daniel Duckworth, Arvind Neelakantan, Ben Goodrich, Lukasz Kaiser, and Samy Bengio. 2019. Parallel scheduled sampling. *arXiv preprint arXiv:1906.04331*.

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*, pages 489–500, Brussels, Belgium. 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, Melbourne, Australia. Association for Computational Linguistics.

Xiao Shi Huang, Felipe Pérez, Jimmy Ba, and Maksims Volkovs. 2020. Improving transformer optimization through better initialization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021a. Confidence-aware scheduled sampling for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2327–2337.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021b. Scheduled sampling based on decoding steps for neural machine translation. In *Proceedings of EMNLP*.

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.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 239–247, Online. Association for Computational Linguistics.

Fandong Meng and Jinchao Zhang. 2019. DTMT: A novel deep transition architecture for neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 224–231. AAAI Press.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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.

Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. 2020. Talking-heads attention. *arXiv preprint arXiv:2003.02436*.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. 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. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466, Online.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR.

Jianhao Yan, Fandong Meng, and Jie Zhou. 2020. Multi-unit transformers for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1047–1059, Online. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

Small Model and In-Domain Data are All You Need

Hui Zeng
Independent Researcher
felix_zeng_ai@aliyun.com

Abstract

I participated in the WMT shared news translation task and focus on one high resource language pair: English and Chinese (two directions, Chinese to English and English to Chinese). The submitted systems (ZengHuiMT) focus on data cleaning, data selection, back translation and model ensemble. The techniques I used for data filtering and selection include filtering by rules, language model and word alignment. I used a base translation model trained on initial corpus to obtain the target versions of the WMT21 test sets, then I used language models to find out the monolingual data that is most similar to the target version of test set, such monolingual data was then used to do back translation. On the test set, my best submitted systems achieve 35.9 and 32.2 BLEU for English to Chinese and Chinese to English directions respectively, which are quite high for a small model.

1 Introduction

I participated in the WMT shared news translation task and focus on the English and Chinese language pair. This language pair is challenging due to the plentiful in-domain bitext training data and abundant monolingual data. High resource means fierce competition, many high-tech companies and universities chose this language pair also. My neural machine translation system is developed using base transformer (Vaswani et al., 2017) architecture and the toolkit I used is THUMT (Zhang et al., 2020). Rules and word aligning model are used to clean parallel data. Language model is used to clean monolingual data. I use a base transformer (Vaswani et al., 2017) architecture since I have only one GPU. The following techniques are used on model training: a. Increase the number of encoder layers to 12 to further improve the encoder's representation capability; b. Back translation (Sennrich et al., 2016) are applied to fully utilize the monolingual corpus. c. Shared vocabulary is used for better performance. d. Four different models using diversified data are trained for ensemble decoding.

2 Data Filtering and Selection

The parallel data is mainly from CCMT Corpus[1], and the monolingual data is collected from the internet. I did not use any other datasets since I think they are not highly related to this news translation task. To evaluate my model's performance, I merged the test set from WMT2017 to WMT2020 to build a big development set.

2.1 Monolingual Data Filtering Using Language Model

In terms of monolingual data, I collected more than 20 million Chinese sentences and more than 15 million English sentences from various websites.
The Chinese text are collected from the following websites:
http://www.chinanews.com/
https://cn.reuters.com/
http://news.ifeng.com/
http://people.com.cn/
https://www.sina.com.cn/
http://www.xinhuanet.com/
https://news.cctv.com/
https://www.qq.com/

[1]http://mteval.cipsc.org.cn:81/agreement/description

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 255–259

https://www.sohu.com/
The English text are collected from the following websites:
https://www.bbc.com/news
www.theguardian.com
https://www.telegraph.co.uk/
https://www.washingtonpost.com/
https://www.latimes.com/
https://www.smh.com.au/
https://www.brisbanetimes.com.au/
https://www.dailymail.co.uk/
https://www.rt.com/
https://time.com/
https://www.euronews.com/
https://www.msnbc.com/
https://www.cbsnews.com/
https://www.nytimes.com/
https://www.newsweek.com/
https://www.foxnews.com/
https://www.standard.co.uk/

The following rules are used for a simple cleaning:
•Remove duplicated sentences.
•Remove the sentences containing special characters.
•Remove the sentences containing html addresses or tags.

Afterwards, language models are used to filter the monolingual data. For English sentences, lm-scorer [2] is used to calculate a score for each sentence, which is the mean of tokens' probabilities. The pre-trained model used for English is GPT-2 (Radford et al., 2019). [3] For Chinese sentences, a pre-trained Chinese GPT-2 (Radford et al., 2019)[4] model is used to calculate a score for each sentence. Then, the English and Chinese sentences are filtered by their scores.

GPT-2 (Radford et al., 2019) is a large transformer-based language model with 1.5 billion parameters, trained on a dataset of 8 million web pages. GPT-2 (Radford et al., 2019) is trained with a simple objective: predict the next word, given all of the previous words within some text.

The threshold I used is determined based on my personal evaluation on the text. After calculating the scores for all the sentences, I sampled the sentences by their scores and perform a language quality check. I started from the extremely low scores and the extremely high scores, and then gradually move the scale from the two ends to the middle until I find that the language quality is up to my standard.

There are about 16 million Chinese sentences and 10 million English sentences left after filtering using language model.

2.2 Parallel Data Filtering Using Rules

For CCMT parallel Corpus and synthetic parallel corpus from back translation, I used the following rules to filter data.

a. Remove duplicated sentence pairs.
b. Remove the lines having identical source and target sentences.
c. Remove the sentence pairs containing special characters.
d. Remove the sentence pairs containing html addresses or tags.
e. Remove the sentence pairs with empty source or target side.

2.3 Parallel Data Filtering Using Word Alignment

In order to get word alignment results, fast_align (Dyer et al., 2013) is used on the CCMT Corpus filtered by rules, then extract-lex [5] is used to generate bilingual phrase tables. The phrase tables are then pruned according to probabilities. Afterwards, I use the pruned phrase table to measure the confidence of the sentence pairs being mutual translations. The confidence score is calculated like this: check each token of the target sentence to find if it has a counterpart in the source side, then perform this operation in the reverse direction, the final confidence score is calculated by summing up the two percentages from two respective directions and then getting the average.
Then the confidence score is used to remove bad sentence pairs. The sentence pairs with confidence scores below 0.6 are discarded. In this way, I finally got a high quality parallel CCMT Corpus.

3 System Description

This section illustrate how I train the model step by step.

[2]https://github.com/simonepri/lm-scorer
[3]https://openai.com/blog/better-language-models/
[4]https://huggingface.co/uer/gpt2-chinese-cluecorpussmall
[5]https://github.com/marian-nmt/extract-lex

3.1 Data pre-processing

For data preprocessing, I use the tokenizer developed on my own to process both Chinese and English. Chinese text (including punctuations and numbers) is split to single character level. I keep the upper and lower case letters of English as they are, since I believe they are also important features for the model. Numbers in English text are also split into single digits. I use byte pair encoding (BPE) (Sennrich et al., 2016) to create a shared vocabulary, so that the vocabulary size is reduced to 45467. I also wrote a post-processor to restore the Chinese and English text to normal form.

3.2 Normal Model Training

To evaluate my model's performance, I merged the test set from WMT2017 to WMT2020 into a big development set. First, I use the CCMT parallel Corpus filtered by rules and word aligning model to train base transformer (Vaswani et al., 2017) English to Chinese and Chinese to English translation models. Two sets of training parameters were used with only one difference: the number of encoder layers. The detailed parameters are as follows:

batch_size=15000,
max_length=384,
hidden_size=512,
filter_size=2048,
num_heads=8,
num_encoder_layers=6 or 12,
num_decoder_layers=6
max_relative_dis=16,
layer_preprocess="layer_norm",
eval_steps=2000,
warmup_steps=4000

Validation is performed every 2000 steps, the training is terminated if there is no gain in BLEU for 20 consecutive validations.

As shown in Table 1, using the same filtered CCMT Corpus, the BLEU scores of models with deeper encoder (12-layer-encoder, 6-layer-decoder) are slightly higher than that of the base version.

Back translation (Sennrich et al., 2016) is a useful data augmentation technique to boost model performance with target side monolingual data. The technique starts from training a target to source translation model using initial bilingual corpus, which is later used to translate the monolingual data in the target language back to source language. Then the synthetic back-translated corpus is concatenated with the original bilingual corpus to train the source to target translation model. After the source to target model is enhanced, the same method can be applied

Model + Corpus	**BLEU EN2 ZH**	**BLEU ZH to EN**
filtered CCMT Corpus base transformer 6-layer-encoder, 6-layer-decoder, base transformer	32.7	21.0
filtered CCMT Corpus base transformer 12-layer-encoder, 6-layer-decoder, base transformer	32.9	21.1
filtered CCMT Corpus half of the filtered monolingual data multiple rounds of back translations 12-layer-encoder, 6-layer-decoder, base transformer	35.3	24.5
filtered CCMT Corpus in-domain monolingual data extracted using test set multiple rounds of back translations 12-layer-encoder, 6-layer-decoder, base transformer best single model	38.3	28.0
filtered CCMT Corpus in-domain monolingual data extracted using test set multiple rounds of back translations 12-layer-encoder, 6-layer-decoder, base transformer ensemble of four models	39.5	29.1

Table 1: Different models and their BLEU scores
I merged the test set from WMT2017 to WMT2020 into a big development set.
The BLEU scores are calculated on this big development set.

again to train the back-translation system in the reversed direction.

I repeat this process using **half** of the filtered monolingual data for several iterations until the BLEU is not increasing.

3.3 Training on In-domain Data

BERT (Devlin et al., 2019) is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. Before feeding word sequences into BERT (Devlin et al., 2019), 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.

After the WMT2021 test set was released, I first translated the Chinese and English test sentences to target versions using the above models, then I generated feature representations for the target versions of the test sentences using pre-trained English BERT (Devlin et al., 2019)[6] and Chinese BERT (Devlin et al., 2019)[7] models.

The example representations are shown in Figure 1 and Figure 2.

```
[[[ 0.0464,  0.2214,  0.1195,  ..., -0.2340,  0.3114,  0.5087],
  [ 0.2481,  0.0501, -0.2061,  ..., -0.2382,  1.0769,  0.2516],
  [-0.0371,  0.2901,  1.1179,  ..., -0.2890,  0.7471,  0.4760],
  ...,
  [ 0.5898, -0.3815,  0.5829,  ...,  0.0953, -0.2390,  0.4471],
  [ 0.6843,  0.1767, -0.2153,  ..., -0.0880, -0.5834, -0.3697],
  [-0.0013,  0.3379,  0.2578,  ...,  0.1301,  0.4615,  0.0671]]]
```

Figure 1: The BERT representation of "I like this competition very much", the tensor shape is [1, 9 , 768]

```
[[[-0.1065,  0.3885,  1.0523,  ..., -0.2281,  0.0663, -0.5467],
  [-0.1091, -0.1201,  0.8952,  ..., -1.3898, -0.3197, -0.0227],
  [ 0.2057, -0.5159,  0.3208,  ..., -0.4561,  0.6920,  0.0200],
  ...,
  [ 0.8978, -0.2769,  0.6887,  ...,  0.6736, -0.2800,  0.1171],
  [-0.1827,  0.5523,  1.5507,  ..., -0.7618,  0.2912, -0.3564],
  [ 0.3490,  0.2635,  1.0002,  ..., -0.7596, -0.1226, -0.3443]]]
```

Figure 2: The BERT representation of "我非常喜欢这个竞赛。", the tensor shape is [1, 12 , 768]

I also generated feature representation for each sentence in the **other half** of the filtered monolingual data. These features are then used to calculate the cosine similarity scores between the target versions of test sentences generated by the previous trained models and the monolingual sentences that are not used in previous training.

Then, the similarity scores are used to find out monolingual sentences that are most similar to the WMT2021 test set.

For each test set sentence, hundreds of monolingual sentences are extracted. In order to determine a threshold score, I randomly sampled 100 test set sentences and their extracted counterparts. Then I checked their similarities and scores using my personal linguistic competences in these two languages. The determined threshold score was then used to automatically extract in-domain data.

Finally, I extracted around 550 thousand Chinese sentences and 420 thousand English sentences as in-domain monolingual data. These sentences are then divided into four equal portions. On the basis of the best models using back translation and the first half of monolingual data, I use four portions of in-domain English data and four portions of in-domain Chinese data to do back translation until the BLEU stops increasing. Therefore, I get four in-domain English to Chinese and four in-domain Chinese to English translation models. These models are then ensembled to build two most powerful models for each direction.

3.4 Results

The BLEU scores on the aforesaid big development set (I merged the test set from WMT2017 to WMT2020 to build a big development set) for each corpus plus model combination are shown in Table 1.

On the WMT 2021 test set, my best submitted systems achieve 35.9 and 32.2 BLEU for English to Chinese and Chinese to English directions respectively, which are even higher than most of the systems from famous high-tech companies.

4 Conclusion

This paper describes Hui Zeng's translation systems (ZengHuiMT) for the WMT2021 news translation shared task. The potential of small model plus in-domain data is explored. I am pleased to argue that, with high quality in-domain data, small model could achieve BLEU scores comparable to that of huge models.

[6]https://huggingface.co/distilbert-base-uncased

[7]https://huggingface.co/bert-base-chinese

Acknowledgments

Thanks to my wife who spend most of her time to take care of our two kids, so that I am able to participate in the contest and complete this paper.

References

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 Proceedings of the 31st International Conference on Neural Information Processing Systems,* pages 6000–6010.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, Yang Liu. 2020. THUMT: An Open Source Toolkit for Neural Machine Translation. *In Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track),* pages 116–122. 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 the 40th annual meeting on association for computational linguistics*, pages 311–318. 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. 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.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),* pages 4171–4186. Association for Computational Linguistics.

Radford, Alec and Wu, Jeff and Child, Rewon and Luan, David and Amodei, Dario and Sutskever, Ilya. 2019. Language Models are Unsupervised Multitask Learners.

The Mininglamp Machine Translation System for WMT21

Shiyu Zhao, Xiaopu Li, Minghui Wu, Jie Hao
Mininglamp Technology, Beijing, China
{zhaoshiyu, lixiaopu, wuminghui, haojie}@mininglamp.com

Abstract

This paper describes Mininglamp neural machine translation systems of the WMT2021 news translation tasks. We have participated in eight directions translation tasks for news text including Chinese↔English, Hausa↔English, German↔English and French↔German. Our fundamental system was based on Transformer architecture, with wider or smaller construction for different news translation tasks. We mainly utilized the method of back-translation, knowledge distillation and fine-tuning to boost single model, while the ensemble was used to combine single models. Our final submission has ranked first for the English→Hausa task.

1 Introduction

This paper describes the Mininglamp submissions to the WMT2021 news translation tasks for eight directions including four high-resource Chinese↔English, German↔English, two medium-resource French↔German and two low-resource Hausa↔English. Furthermore, all of our systems were built with constrained data sets.

For this participation, we experimented with some smaller or wider Transformer (Vaswani et al., 2017) architectures to reach a reliable baseline based on different resource scales, sampling or beam search in back-translation to generate more suitable pseudo bilingual sentences. Particularly in the low-resource tasks, Hausa↔English, the Transformer-Small neural machine translation was built for the baseline, we presented iterative between back-translation and fine-tuning pattern which significantly improve the BLEU score on the validation set, and it worked well on English→Hausa task. Due to time constraints, we did not experiment on Hausa→English task. This path could be an experiment in the future work.

As for the data augmentation aspect, we experimented with several back-translation methods (Sennrich et al., 2016a), including the beam search, unrestricted sampling and sampling-topK (Edunov et al., 2018), to leverage the target-side monolingual data. We also applied knowledge distillation (Freitag et al., 2017) to leverage the source-side monolingual data.

Our systems followed four main steps:1) data filtering and preprocessing, 2) back-translation to generate pseudo bilingual data, 3) knowledge distillation by monolingual data, 4) fine-tuning with in-domain.

It should be emphasized that we used Marian[1] (Junczys-Dowmunt et al., 2018) to implement only for Hausa↔English baseline systems, and Fairseq[2] (Ott et al., 2019) for the rest, include Hausa↔English back-translation and knowledge distillation models.

2 System Overview

2.1 Data Filtering and Preprocessing

In this section, we discuss the preprocessing, normalization and filter techniques carried out in an attempt, in order to reduce spurious uncertainty in the modeling problem.

2.1.1 Text Preprocessing

Generally, we carried out the following text preprocessing steps prior to use in every model:

- Normalization: Unicode canonicalization, replacement of common multiple encoding errors present in training data, standardization of quotation marks into directional variants, conversion of any traditional Chinese characters into simplified forms, conversion of any Chinese full-width characters and segmental Chinese full-width punctuation into half-width forms. Normalize punctuation in all data by using Moses[3] (Koehn et al., 2007)

[1] https://github.com/marian-nmt/marian
[2] https://github.com/pytorch/fairseq
[3] https://github.com/moses-smt/mosesdecoder

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 260–264

(`normalize-punctuation.perl`) script except for every language pair.

- Segmentation: Chinese was segmented using the Jieba[4] segmentation tool, and tokenizer using Moses (`tokenizer.perl`) script for English, German, French and Hausa. For the Hausa tokenizer, we used English tokenizer instead.

- True-case: The word, at the start of a sentence, containing only an initial capital letter was replaced with the capitalized variant. That occurred most frequently in other positions of the English monolingual training data. Thus, in the previous sentence, the initial token would be "words" rather than "Words". We used Moses' script for true-case.

- Subword: The neural machine translation system is capable of open-vocabulary translation by representing rare and unseen words as a sequence of subword units. The model was trained based on subword-nmt[5] on the parallel training corpus.

2.1.2 Data Filtering

For all language pairs, the data filtering process for the training bilingual corpus stayed to the principle with the following rules:

- Filter out the sentence pairs that contain blank lines either from the source side or the target side.

- Filter out the sentence pairs that the source side and the target side at the same.

- Filter out the sentences with the length ratio falling outside from 0.4 to 2.5.

- Filter out the sentences whose punctuation and foreign words taking more than 40 percent.

- Remove the sentences which are longer than 200 words, or exceed a single word with 30 characters.

- Filter out the sentences which contain HTML tags or duplicated translations.

- Filter out the sentences which its word ratio between the source and the target exceeds 1:2.5 or 2.5:1.

- Identify language and delete foreign languages. Filter parallel and monolingual data by language detection using cld2[6].

The rules described above were also employed when cleaning monolingual and back-translation data. In the monolingual data particularly there were some lines that include two or more sentences, we cut them into several sentences by writing a script.

2.2 Data Augmentation

2.2.1 Back-Translation

Back-translation (Sennrich et al., 2016a) is an essential method to integrate the target side monolingual synthetic knowledge when building a state-of-the-art neural machine translation system. Especially for low-resource language tasks, it's indispensable to augment the training data by mixing the pseudo corpus with the parallel part. In that the target side, lexicon coverage was insufficient. The nucleus sampling (Holtzman et al., 2020) in back-translation to generate more suitable pseudo bilingual sentences. We attempted several data augmentation methods as follow, with different single technologies or combinations.

- Beam search: Generated target translation by beam search with beam 5.

- Sampling: Selected a word randomly from the whole distribution in each step, which increases the diversity of pseudo corpus with low precision, compared with beam search.

- Sampling Top-K: Selected a word in a restricted way that only top-K (we set K as 16) words could be chosen.

2.2.2 Forward Translation to Generate Synthetic Parallel Sentence

For Chinese↔English tasks. To generate a more diverse pseudo-parallel corpus, we use forward-translated to do generated synthetic parallel sentences on source monolingual data only by our own ensemble model.

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

[5] https://github.com/rsennrich/subword-nmt

[6] https://github.com/CLD2Owners/cld2

2.2.3 Knowledge Distillation

We used knowledge distillation (Kim and Rush, 2016) to do distillation on the original dataset. Specifically, we translated the source-side of the bilingual data using previously trained proposal models, and generated distilled candidates. We then trained models on filtered data along with the original bilingual data and back-translation data.

2.3 Iterative Back-translation and Fine-tuning

A process which iterative twice between back-translation and fine-tuning was implemented by following steps for the low-resource Hausa↔English tasks.

2.4 Reranking

For German↔English, French↔German tasks, we followed noisy-channel (Yee et al., 2019) reranking using one neural language model and three reverse translation models.

3 Experiment

3.1 Experiment Settings

In order to demonstrate the experiments of the system, there some experiment details should be clarified. To train all of the models used in our system, we made use only of the constrained data sets provided to shared news translation task participants. On the other side, the baseline models were trained on parallel corpus only by cleaned corpus. In terms of model evaluation, the main indicator for the report was calculated according to sacreBLEU[7] (Post, 2018) based on the results which has been removed parts of post-preprocessing such as removed BPE symbols, detruecased, detokenized, etc.

The Transformer-Small was implemented based on Marian (Junczys-Dowmunt et al., 2018) as our baseline for Hausa↔English tasks. For Chinese↔English, German↔English and French↔German tasks, we implemented the Transformer-Big FFN-8192 based on Fairseq (Ott et al., 2019) as our baseline model. We used Adam optimizer (Kingma and Ba, 2014) during training, learning rate was 5e-4, $\beta_1 = 0.9$, $\beta_2 = 0.98$, weight decay was 0.0001, label smoothing was 0.1. Specifically, the learning rate warmed up over the 8,000 steps for pre-normalize architectures Transformer-Big FFN-8192 model. The system shuffled the training data before generating the training batch for each epoch, so the document context information was not considered in this case. FP16 was applied to accelerate training with few performance damage during the training process.

3.2 Chinese↔English

For Chinese↔English system, our parallel corpus included CCMT, wikititles-v3, wikimatrix-v1, para-crawl-v7.1, news-commentary-v16 corpus. While Chinese were segmented by Jieba word segmentation toolkit, English was tokenized by Moses tokenizer script. Based on the result of data Filtering, we used 17 million Chinese↔English parallel data corpus for training the baseline model. As the next step after the preprocessing, we trained BPE (Sennrich et al., 2016b) models which were learned with 32,000 merge operations for joined English and Chinese on the parallel data. We built separately vocabularies for each language, and the final vocabulary size of Chinese was 42K and English was 22K. Baseline train data we followed drop-BPE (Provilkov et al., 2020). We trained the Transformer-Big FFN-8192 model for Chinese↔English.

For back-translation, we selected 20 million News Crawl 2020 English monolingual data for Chinese→English task. All News Crawl Chinese monolingual data and selected 20 million Extended Common Crawl Chinese monolingual data were combined for English→Chinese task. Back-translation data were combined by sampling top-16 and beam search. At the same time, there was a combination between back-translation data and parallel data corpus in order to train Chinese↔English models. We selected 10 million Chinese and English sentences respectively for forward translation and knowledge distillation to generate synthetic parallel sentences.

Our final submissions consisted of three Transformer-Big FFN-8192 models with different configurations, using the beam search with a beam size of 5, and set lenpen 2.0. Table 1 shows that the translation quality was improved by using the proposed techniques.

3.3 Hausa↔English

The parallel corpus for Hausa↔English system included para-crawl-v8, wikititles-v3, Khamenei and Opus corpus, which was tokenized by Moses tokenizer script. It should be clear that Hausa used tokenizer by English mode. After the data filter-

[7] https://github.com/mjpost/sacrebleu

System	zh-en	en-zh
baseline	30.2	42.9
+ Back Translation	33.4	45.1
+ Knowledge Distillation	33.8	46.2
+ Fine-tuning	34.7	47.8
+ Ensemble	35.5	48.6

Table 1: SacreBLEU scores on newstest2020 Chinese↔English tasks.

ing, we used 550 thousand Hausa↔English parallel data corpus for training the baseline model. A joint BPE model was applied with 10,000 merge operations. Moreover, shared vocabularies were selected for Hausa↔English language pairs.

We used Marian trained Transformer-Small[8] model for Hausa↔English baseline, with learning rate ranging from 0.0008 to 0.001, warmup steps fixing at 48,000. Three models(3e3d, 4e4d, 6e4d) were trained under different architectures on single 2080Ti GPU.

For English↔Hausa back-translation, the standard Transformer-Big model implemented in Fairseq. We selected 4.5 million Hausa monolingual data by data filtering and language detection, and 20 million English monolingual data from the News Crawl 2020 were filtered as the back-translation dataset. Every time we handled the back-translation process, the beam search was applied. Then the back-translation and the fine-tune were executed twice. For Hausa→English, due to time constraints, it was limited to one back-translation and fine-tune.

In the fine-tuning stage, 200 sentences from the newsdev2021 were kept randomly as the validation set, and other sentences were attributed to fine-tune the model.

Table 2 shows that the translation quality was improved by using the proposed techniques. Our final submissions consisted of two Transformer-Big models.

3.4 German↔English

For German↔English task, the provided parallel sentences were completely joined together so as to get about 95 million sentence pairs. Then, sentences with lots of punctuation masks and non-alpha-number characters were removed, as well as the sentences whose length ratio was larger than 2. As a result, 52 million sentences were selected to be candidates. After that, BPE was learned jointly with 32k as the merge operations, and the size of the vocabulary was 32,168. The model's parameters for both directions were copied from the Transformer-Big in the paper "Attention is all you need" (Vaswani et al., 2017). Finally, we got three English→German models and two English↔German models for ensembling and reranking. The language model used for reranking was trained with GPT-3 using data cleaned from news 2020. All the models were trained using Fairseq. The overview of our German↔English system is listed in Table 3.

[8]The dimension of word embedding was 256, the dimension of the feed-forward network was 1024, multi-head was 4, encoder and decoder layer was 4.

System	ha-en	en-ha
baseline	13.8	11.6
+ 1st. Back-translation	24.6	22.7
+ 1st. Fine-tuning*	29.7	25.5
+ 2nd. Back-translation*	-	26.2
+ 2nd. Fine-tuning*	-	26.9
+ Ensemble*	31.7	27.4

Table 2: SacreBLEU scores on newsdev2021 Hausa↔English tasks. Steps with extra * marks are evaluated in the tiny 200 lines new validation set.

System	de-en	en-de
baseline	44.1	40.0
+ Ensemble	45.1	41.1
+ Reranking	45.5	41.4

Table 3: SacreBLEU scores on newstest2016 German↔English tasks. Learning rate for training is 0.001 and warmup steps are 4000.

3.5 French↔German

For French↔German task, about 7 million sentences were left after removing the sentences with invalid characters or punctuations from the original parallel sentences. We trained the BPE codes with 32k as the merge operations. The final vocabulary size for German was 32,144 and for French was 32,176. We introduced forward translation in German→French direction using models trained from the original parallel dataset. In both directions, the models were based on the Transformer-Big as the basic architecture. At last, three French→German models and two German→French models, trained from forward-translation, were applied to ensembling and reranking. The language model used for reranking was

trained with GPT-3 using data cleaned from news 2020. The models from this system were completely trained by Fairseq. Check the overview of our German↔French systems in Table 4.

System	de-fr	fr-de
baseline	30.9	27.6
+ Knowledge Distillation	31.3	-
+ Ensemble	32.6	28.8
+ Reranking	34.1	30.9

Table 4: SacreBLEU scores on newstest2019 French↔German tasks.

4 Conclusions

This paper described the Mininglamp submissions to the WMT2021 eight news translation tasks, and our main exploration was using more diversified architectures, back-translation, fine-tuning and ensemble. We used a similar data preprocess and filtering strategy for all the tasks, containing statistical information-based rules. And we experimented with back-translation by different decoding strategies, using the Transformer-Small model and iterative between back-translation and fine-tuning for low-resource.

References

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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. 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.

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.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

The NiuTrans Machine Translation Systems for WMT21

Shuhan Zhou[1], Tao Zhou[1], Binghao Wei[1], Yingfeng Luo[1], Yongyu Mu[1], Zefan Zhou[1], Chenglong Wang[1], Xuanjun Zhou[1], Chuanhao Lv[1], Yi Jing[1], Laohu Wang[1], Jingnan Zhang[1], Canan Huang[1], Zhongxiang Yan[1], Chi Hu[1], Bei Li[1], Tong Xiao[1,2] and Jingbo Zhu[1,2]

[1]NLP Lab, School of Computer Science and Engineering, Northeastern University
[2]NiuTrans Research, Shenyang, China
zhoushuhan199710@163.com, zhoutao_neu@outlook.com
{xiaotong,zhujingbo}@mail.neu.edu.cn

Abstract

This paper describes NiuTrans neural machine translation systems of the WMT 2021 news translation tasks. We made submissions to 9 language directions, including English↔{Chinese, Japanese, Russian, Icelandic} and English→Hausa tasks. Our primary systems are built on several effective variants of Transformer, e.g., Transformer-DLCL, ODE-Transformer. We also utilize back-translation, knowledge distillation, post-ensemble, and iterative fine-tuning techniques to enhance the model performance further.

1 Introduction

Our NiuTrans team participated in the WMT 2021 news translation shared tasks, including English↔Chinese (EN↔ZH), English↔Japanese (EN↔JA), English↔Russian (EN↔RU), English↔Icelandic (EN↔IS) and English→Hausa (EN→HA), nine submissions in total. All of our systems were built with constrained data sets. We adopt some effective models and useful methods, which have been witnessed the success in previous papers (Wang et al., 2018; Li et al., 2019; Zhang et al., 2020; Meng et al., 2020; Wu et al., 2020b; Chen et al., 2020; Yu et al., 2020; Wu et al., 2020a; Wei et al., 2020).

To enhance the performance of the single model, we choose pre-normalized Transformer-DLCL (Wang et al., 2019) and ODE-Transformer (Li et al., 2021a) as the backbone. All systems are built upon the relative position representation (Shaw et al., 2018) due to its strong performance when models are deep (Li et al., 2020). For the system combination, we adopt the post-ensemble (Kobayashi, 2018) to find the most similar hypothesis among several ensemble outputs, which could be regarded as a reranking technique without pre-training. Previous works have emphasized the importance of diversity when building ensemble systems. Besides the architecture diversity, we also adopt iterative ensemble knowledge distillation leveraging the source-side monolingual data to enlarge the diversity. More details please refer to (Li et al., 2019).

Our data preparation pipeline consists of threefold: (i) For the data filtering. We use a stricter cleaning process than last year (Zhang et al., 2020). Details will be discussed in Section 2.1. (ii) For the data augmentation, both iterative back-translation (Sennrich et al., 2016a) method, and iterative knowledge distillation (Freitag et al., 2017) method are employed to take the full advantage of monolingual data provided by the WMT organization. In the back-translation stage, we leverage target-side monolingual sentences to generate source-side pseudo sentences and use a nucleus sampling (Holtzman et al., 2019) decoding strategy to improve the generalization ability. Furthermore, we leverage in-domain source-side monolingual data by applying iterative knowledge distillation. (iii) For data selection, it's hard to find massive in-domain data for low-resource languages to train a neural language model, so we use a statistical n-gram language model (XenC toolkit3[1]) instead.

Domain finetuning is quite essential to improve the translation system given a certain target domain. We use domain adaptation to migrate the models from the general domain to the news domain by iterative finetuning. After in-domain finetuning, we use multiple ensemble combinations by the post-ensemble method.

This paper is structured as follows: In Section 2, we introduce several effective techniques, including data preprocessing, deeper and wider Transformer models, iterative back-translation, itera-

[1]https://github.com/antho-rousseau/XenC

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 265–272
November 10–11, 2021.

Model	Depth	Hidden Size	Filter Size	RPR	Batch size	update freq
Transformer	6	512	2048	✗	4096	1
Transformer (Pre-Norm)	24	512	4096	✓	2048	4
Transformer-DLCL	25	512	4096	✓	2048	4
Transformer-DLCL	30	512	2048	✓	2048	4
Transformer-DLCL	30	512	4096	✓	2048	4
ODE Transformer	6	1024	4096	✓	2048	8
ODE Transformer	12	1024	4096	✓	2048	8

Table 1: The details of several model architectures we used.

tive knowledge distillation, fine-tuning and post-ensemble. In Section 3, we show the experiment settings and report the experimental results of the validation set (newstest2020). Finally, we draw the conclusion in Section 4.

2 System Overview

2.1 Data Preprocessing and Filtering

For word segmentation, we use different tools in six languages. English, Russian, Hausa and Icelandic sentences were segmented by Moses (Koehn et al., 2007), while Chinese and Japanese used NiuTrans (Xiao et al., 2012) and MeCab[2] separately. Then BPE (Sennrich et al., 2016b) with 32K operations is used for five languages sides independently, except for 36K operations in Russian.

The quality of the parallel training data is crucial to the performance of the models, so we use rigorous data filtering scheme as the suggestion in Zhang et al. (2020)'s work. For most language pairs, rules are as follows:

- Filter out sentences that contain long words over 40 characters or over 150 words.
- The word ratio between the source word and the target word must not exceed 1:3 or 3:1.
- Use Unicode to filter sentences with more than 10 other characters.
- Filter out the sentences which contain HTML tags or duplicated translations.
- In monolingual data, some sentences contain two or more sentences. We write a script to cut them into several sentences.

We use these rules to filter bilingual and monolingual data, detecting low-quality sentences with misalignment, translation errors, illegal characters, and missing translation.

[2]https://github.com/taku910/mecab

2.2 Model Architectures

As shown in previous work (Li et al., 2019; Zhang et al., 2020; Meng et al., 2020), deep Transformers bring significant improvements than the baseline on various machine translation benchmarks. In their work, the performance of the model was significantly improved by increasing the encoder depth. We keep the decoder depth unchanged as the brought benefit is marginal when the encoder is strong enough (Li et al., 2021b).

Hence, we train two deep models in our experiment: Transformer DLCL (Wang et al., 2019) and ODE Transformer (Li et al., 2021a) with a larger filter size. ODE Transformer is designed from the ordinary differential equations (ODE) perspective. Higher-order ODE solutions can gain fewer truncation errors, thus reducing the global error and improving the model performance. The details of several models we mainly experimented with are summarized in Table 1.

In addition, we incorporate relative position representation (RPR) into the self-attention mechanism on both the encoder and decoder sides. Preliminary experiments demonstrate that only relative key information is enough, and we set the relative window size to 8.

2.3 Large-scale Back-Translation

Back-translation (BT) is an effective data augmentation technique to boost the performance of NMT models, which use monolingual data to generate pseudo-training parallel data. Back-translation is divided into three stages:

- Using bilingual parallel data to train a target-to-source intermediate ensemble of models.
- Utilizing the ensemble of reverse direction models to translate the target monolingual corpus into the source corpus.

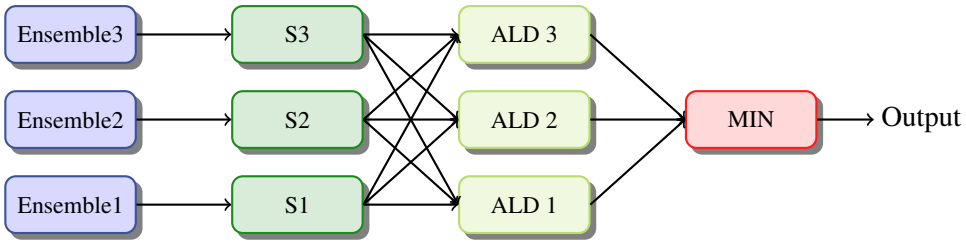


Figure 1: Process of the post-ensemble method. Sn denotes the sentence generated by the n-th ensemble of models. ALD denotes the average Levinstein distance of a sentence with other sentences. For example, $ALD\,1 = \frac{1}{2} * (Levinstein\_distance(S1, S2) + Levinstein\_distance(S1, S3))$. Finally, we select the sentence of the smallest ALD.

- Training models with the bilingual parallel corpus and the synthetic parallel corpus together.

Select in-domain monolingual data during back-translation can significantly alleviate domain adaptation problems (Zhang et al., 2020). Our in-domain data consist of the test sets released in recent years and the News Commentary high-quality monolingual data. Due to insufficient data in the domain, we used a statistical method to select in-domain data, the XenC toolkit. Furthermore, to avoid the high ranking of short sentences, we choose the in-domain source side sentences according to the distribution of sentence tokens number in the previous years' test set.

For all tasks, we employ the beam search and Nucleus Sampling approaches to generate pseudo corpus and the scale of the pseudo corpus was about 1:1 to the real corpus.

2.4 Iterative Knowledge Distillation

Knowledge distillation (KD) has been proven to be a powerful technique to improve the performance of the student model by transferring knowledge from the teacher model (Li et al., 2019; Zhang et al., 2020). Here, we regard the ensemble models as a teacher model and single models as student models. Specifically, we first use the ensemble model to generate synthetic corpus in the forward direction. Then, we merge the synthetic parallel corpus with the bilingual parallel corpus to teach student models. And by searching for better model ensemble combinations, we can provide stronger teacher models for the next round of knowledge distillation. Our experiment found that the gap between the single model and the integrated model gradually narrowed as the iteration progressed. So for the nine tasks we participated in, two iterations of knowledge distillation deliver the best performance.

2.5 Finetuning

Domain adaptation plays an important role in improving the performance of the models. A practical method of domain adaptation is to train models on large-scale out-domain corpus and then fine-tune the models with in-domain corpus (Luong and Manning, 2015). For all tasks, we mainly reuse an iterative fine-tuning process (Zhang et al., 2020) and use the development sets and the test sets of previous years as in-domain corpus.

It is worth noting that, in order to be consistent with the composition of the test set, we select parallel sentences pair from the previous development sets and test sets in which the source side is real and the target side is manually translated. Moreover, we found that iterative fine-tuning can better improve the translation quality of the names of news organizations in the news field.

2.6 Post-ensemble

Ensemble learning is a technique widely used in several WMT shared tasks, which improves performance by using multiple single models. In neural machine translation, a practical method of the model ensemble is to combine the probability distribution on the target vocabulary of different models in each step of sequence prediction. Here, we adopted their method, which uses a greedy-based strategy to find a better combination of models on the development set. However, enumerating all combinations of candidate models is an inefficient and cumbersome way.

In our ensemble experiments, we set the number of the ensemble to four and six. We observed that simply expanding the scale of the ensemble does not necessarily improve translation performance.

Besides, brute force search for all models is costly and unrealistic. As the number of models increases, the ensemble easily exceeds the computer capacity limit. Therefore, for all tasks, we finally search for four single models as an ensemble.

In addition, we use a simple but effective unsupervised ensemble method, post-ensemble, which uses a clustering method to select a majority-like output from multiple ensembles. As shown in the figure 1, we first choose several ensemble combinations composed of different models to obtain more diversity. Then we use these ensembles to generate multiple sentences, respectively. Next, we calculate the Levinstein distance between each sentence, and finally, we select the sentence of the smallest average Levinstein distance with other sentences.

For more detailed content, please refer to the original paper (Kobayashi, 2018). This technology can further improve the performance of the system based on ensemble learning.

3 Experiment

3.1 Experiment Settings

The implementation of our models is based on Fairseq (Ott et al., 2019). All models were trained on 8 RTX 2080Ti GPUs. We selected the pre-norm Transformer-base as the baseline for all tasks and enhanced our deep or wide models by enlarging the model depth and the hidden size, respectively. We used Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.997$ during training. As suggested in Ott et al. (2018) and Wang et al. (2019)'s work, models with larger capacities tend to perform much better within large batch size and learning rate. Due to the high GPU memory consumption, accumulated gradients every two steps where each batch contains 2048 tokens. Training for 15 epochs is sufficient for most tasks, and models have shown convergence in validation perplexity. The max learning rate and warmup step were set to 0.002 and 8000 for deep models, and 0.0016 and 16000 for deep and wide models, e.g., Transformer-DLCL, whose hidden dimension is 768. All the dropout probabilities were set to 0.1, including the residual dropout, attention dropout, and the ReLu dropout. We also used FP16 mix-precision training to accelerate further the training process with almost no loss in BLEU.

3.2 EN↔ZH

For EN↔ZH tasks, the training data consists of ParaCrawl, News Commentary v16, WikiMatrix, UN Parallel Corpus V1.0, and the CCMT Corpus. We regarded the newstest2019 as the valid set and the newstest2020 as the test set to tune the hyperparameters. After filtering the data, we sampled the top 12 and 20 million data according to the XenC score as the bilingual dataset. For the ZH→EN task, we used 12 and 20 million data to train the baseline model, respectively, and found that the model trained by 12 million data is 1 and 1.2 BLEU point higher than the model trained by 20 million data in the valid and test set. We found that the data quality of the bottom 8 million is lower and also selected the 12 million data as our training data.

During the first-step back-translation, we sampled 8 million monolingual data from the combination of News crawl, News Commentary, News discussions, and News crawl. Then we used the baseline model to generate the hypotheses via the beam search strategy as the pseudo dataset. In the second-step back-translation, we utilized the same amount of pseudo data while using nucleus sampling, whose p is 0.9. For ZH→EN and EN→ZH, we got BLEU improvements of 1.8 and 2.9 in the first back-translation and further BLEU improvements of 0.5 and 0.8 in the second back-translation, respectively.

In addition, we implemented knowledge distillation twice to iteratively enhance the single model with the ensemble outputs. The main goal is to make the single student mimic the behavior of the ensemble models, thus obtaining stronger ensemble teachers in the next step. We used the test sets in previous years as in-domain data in EN→ZH and EN→ZH directions respectively, and we used the XenC tool to sample 3 million from the large scale monolingual data based on in-domain data. Then we used the best ensemble of models to construct pseudo data by decoding them and merge them to the original training data to continue training for each model. We got BLEU improvements of 1.1 and 0.6 in the first knowledge distillation and further BLEU improvements of 0.6 and 0.3 in the second knowledge distillation in ZH→EN and EN→ZH.

After knowledge distillations, we used the newstest2017-2019 to fine-tune our models for five epochs with the 0.0001 learning rate and got 2 and 0.4 BLEU improvements in EN→ZH and EN→ZH

System	EN→ZH	ZH→EN	EN→JA	JA→EN	EN→HA
Baseline	41.9	30.1	34.5	21.4	10.9
DLCL30-RPR	42.8	31.0	35.6	21.6	11.9
+Iteratively BT	46.5	33.3	38.4	21.7	16.5
+Iteratively KD	47.4	35.0	41.8	25.9	18.2
+Fine-tune	47.8	37.0	42.0	26.4	-
+Ensemble	48.8	37.2	42.7	27.4	18.5
+Post-ensemble & Post edit	49.0	37.5	43.6	27.4	-

Table 2: BLEU evaluation results on the WMT 2020 EN↔ZH, EN↔JA test sets and WMT 2021 EN→HA development sets.

directions, respectively. In the final stage, we add newstest2020 to the fine-tuning data. Finally, we searched for the best five combinations of 4 out of 12 models for post-ensemble to ensure the diversity of the models. Based on the ensemble method, post-ensemble further brought us +0.2 and +0.3 BLEU in ZH→EN and EN→ZH directions. Our main results showed in table 2, we find that iterative back-translation, iterative knowledge distillation, and iterative fine-tune are effective methods to get significant improvements.

3.3 EN↔JA

For EN↔JA tasks, we chose ParaCrawl v7.1, News Commentary v16, WikiMatrix, Japanese-English Subtitle Corpus, The Kyoto Free Translation Task Corpus, TED Talks total of six parallel data corpora about 17.5 million. For the ParaCrawl v7.1, we only selected 8.5 million data according to the score of sentences provided by the dataset. We chose all of News Crawl and News Commentary and 12 million data sampled from Common Crawl for the Japanese monolingual data. After merging corpora into training data, we found that there were many-to-one situations in both the target side and the source side. Therefore, we sorted sentences and calculated the Levenshtein ratio of two adjacent sentences to remove duplication sentences. We applied this method to all version data before training models and removed 10 percent of the total data. We randomly selected one out of many sentences in which Levenshtein ratios are greater than or equal to 0.9.

We also implemented tagged back-translation, which brought us +2.8 BLEU in the EN→JA task. In addition, beam search and nucleus sampling were used to generate two parts of translations to increase data diversity, and each part contains 12 million data. An interesting phenomenon is that back-translation is useful for EN→JA task while knowledge distillation is helpful for JA→EN task. We suspect this is because the domain of Japanese monolingual more fits the field of the test set.

We also implemented knowledge distillation and fine-tuned iteratively. During the knowledge distillation phase, we used FDA[3] and XenC to select monolingual data more like newstest2020 and generated pseudo data by using both post-ensemble and ensemble methods. During the fine-tuning phase, we used the WMT 2020 valid set and opposite direction test set. After performance stopped increasing at the second fine-tune, we utilized the best ensemble models to regenerate pseudo data by back-translation and knowledge distillation. Then, we retrained multiple deep models. Finally, we put all models together to greedy search for the best combination of 13 models. And this method brought us +0.7 BLEU in JA→EN task. Our main results are shown in table 2.

3.4 EN↔RU

For EN↔RU tasks, we used only two parallel datasets, including ParaCrawl v8 and News Commentary. After the data filter, about 12M sentence pairs were left to build our system. Additionally, we set the merge operations of BPE to 36K.

We also used iterative back-translation, iterative knowledge distillation, and fine-tuned to enhance the model. During the back-translation, English monolingual data is the same as the EN↔JA part, and Russian monolingual data sources consist of News Crawl and News Commentary. During the knowledge distillation, we used FDA to select 4 million sentence pairs from the monolingual dataset according to the newstest2020 and newstest2019. Then we merged them with the official development set to continue training our

[3]https://github.com/bicici/FDA

System	EN→RU	RU→EN	EH→IS	IS→EN
Baseline	22.0	35.6	20.9	28.4
ODE big6-RPR	22.7	36.8	22.4	30.5
+Iteratively BT	23.0	38.2	28.5	34.9
+Iteratively KD	23.3	38.9	30.7	36.0
+Fine-tune	24.4	39.4	-	-
+Ensemble	24.8	39.9	31.2	36.4

Table 3: BLEU evaluation results on the WMT 2020 EN↔RU test sets and WMT 2021 EN↔IS development sets.

Task	Submission	Task	Submission
EN→ZH	35.8	EN→RU	28.4
ZH→EN	31.9	RU→EN	41.8
EH→JA	46.2	EH→IS	30.6
JA→EN	27.2	IS→EN	39.2
EH→HA	19.7	-	-

Table 4: Our final submission results in nine tasks.

models for five epochs. After KD, we used the newstest2019 and newstest2018 to fine-tune our models for five epochs with the 0.0001 learning rate and got 1.1 and 0.5 BLEU improvements in EN→RU and RU→EN.

The detailed and full results can be described in Table 3. Iterative BT, KD, and fine-tune are still very effective and improved 2.8 and 4.3 compared with the base model in EN→RU and RU→EN tasks, respectively.

3.5 EN↔IS

The process of EN↔IS tasks is similar to EN→HA task but more complicated. Concretely, we used four parallel datasets, including ParaCrawl v7.1, Wiki Titles v3, WikiMatrix, and ParIce. After the data filtering, about 5.5 million sentence pairs were left to build the baseline system. The experimental results are listed in Table 3. We obtained significant improvements of 6.1 and 4.4 BLEU in EN→IS and IS→EN directions, respectively.

Then we implemented iterative KD two times and sampled 3 million in-domain source data according to WMT2021 development sets. Table 3 shows that it's a very effective method to get 2.2 and 1.1 improvements. Furthermore, we fine-tuned models iteratively twice to transfer the knowledge into the target domain. Due to implementing two ensemble combinations to decode sentences, the model ensemble still gained 0.7 and 0.8 improvements.

3.6 EN→HA

In the EN→HA direction, we used ParaCrawl v8, Khamenei corpus, and English-Hausa Opus corpus three data sets, obtaining 1.43M parallel data after cleaning. We collected News crawl, Extended Common Crawl, and Common Crawl for the monolingual data, resulting in 5.7M Hausa monolingual data. Considering the insufficient scale of Hausa, we used all monolingual data in each round of back-translation. The implementation details of iterative knowledge distillation and back-translation are almost the same as the EN↔ZH tasks.

Table 2 summarized the results. We can observe that the wide and deep models were still effective in low-resource language pairs. Through the back-translation and knowledge distillation techniques, we gain 4.6 and 1.7 BLEU improvements, respectively.

3.7 Submission Results

The results we finally submitted are shown in table 4. We participated in nine tasks this year. On the whole, all of our systems performed competitively, especially in EH→IS and RU→EN directions.

Through all the experimental results, we found that different methods perform differently on nine tasks. Among them, iterative BT is effective for almost all tasks, except for the JA→EN task. Iterative KD performs better for EN↔ZH, EN↔JA and EH↔IS tasks, while fine-tune is more suitable for ZN→EN and EN→RU tasks.

4 Conclusion

This paper introduced our submissions on WMT21 nine tasks. Our main exploration is using a new effective architectures ODE Transformer and utilizing post-ensemble technology to enhance the system. And we experimented with iterative back-translation by different decoding strategies, iterative knowledge distillation, iterative fine-tuning, model ensembling and post-ensemble.

References

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook AI's WMT20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hayato Kobayashi. 2018. Frustratingly easy model ensemble for abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4165–4176, 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, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Bei Li, Quan Du, Tao Zhou, Shuhan Zhou, Xin Zeng, Tong Xiao, and Jingbo Zhu. 2021a. ODE transformer: An ordinary differential equation-inspired model for neural machine translation. *CoRR*, abs/2104.02308.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Bei Li, Ziyang Wang, Hui Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2021b. Learning light-weight translation models from deep transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13217–13225. AAAI Press.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 995–1005, Online. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 239–247, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. 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.

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)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The NiuTrans machine translation system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages

528–534, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin, and Shiliang Sun. 2020. HW-TSC's participation in the WMT 2020 news translation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 293–299, Online. Association for Computational Linguistics.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020a. The volctrans machine translation system for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 305–312, Online. Association for Computational Linguistics.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020b. Tencent neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 313–319, Online. Association for Computational Linguistics.

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, Jeju Island, Korea. Association for Computational Linguistics.

Lei Yu, Laurent Sartran, Po-Sen Huang, Wojciech Stokowiec, Domenic Donato, Srivatsan Srinivasan, Alek Andreev, Wang Ling, Sona Mokra, Agustin Dal Lago, Yotam Doron, Susannah Young, Phil Blunsom, and Chris Dyer. 2020. The DeepMind Chinese–English document translation system at WMT2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 326–337, Online. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, bei Li, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345, Online. Association for Computational Linguistics.

Improving Similar Language Translation With Transfer Learning

Ife Adebara **Muhammad Abdul-Mageed**
Natural Language Processing Lab
The University of British Columbia
{ife.adebara,muhammad.mageed}@ubc.ca

Abstract

We investigate transfer learning based on pre-trained neural machine translation models to translate between (low-resource) similar languages. This work is part of our contribution to the WMT 2021 Similar Languages Translation Shared Task where we submitted models for different language pairs, including French-Bambara, Spanish-Catalan, and Spanish-Portuguese in both directions. Our models for Catalan-Spanish (82.79 BLEU) and Portuguese-Spanish (87.11 BLEU) rank top 1 in the official shared task evaluation, and we are the only team to submit models for the French-Bambara pairs.

1 Introduction

We present the findings from our participation in the WMT 2021 Similar Language Translation shared task 2021, which focused on translation between similar language pairs in low-resource settings. The similar languages task focuses on building machine translation (MT) systems for translation between pairs of similar languages, without English as a pivot language.

Similarity between languages interacts with MT quality, usually positively (Adebara et al., 2020). Languages described as similar usually share certain levels of mutual intelligibility. Depending on the level of closeness, certain languages may share orthography, lexical, syntactic, and/or semantic structures which may facilitate translation between pairs. However, **(a) scarcity of parallel data** that can be used for training MT models remains a bottleneck. Even high-resource pairs can suffer from **(b) low-data quality**. That is, available data is not always guaranteed to be actual bitext with target standing as a translation of source. In fact, some open resources such as OPUS (Tiedemann, 2012a; Tiedemann et al., 2015) can suffer from noise such as when the source and target sentences belong to the same language. In this work, we tackle both **(a) scarcity** and **(b) low-data quality**. For **a**, we use simple knowledge transfer from already trained MT models to the downstream pair. For **b**, we use a simple procedure of language identification to remove noisy bitext where both the source and target are detected to be the same language or where source or target is identified as a different language from what it is expected to be.

The models we develop are for Spanish to Catalan (ES-CA), Catalan to Spanish (CA-ES), Spanish to Portuguese (PT-ES), Portuguese to Spanish (PT-ES), French to Bambara (FR-BM), and Bambara to French (BM-FR) language pairs[1]. Whenever possible, we choose an available MT model trained with the same source and target languages as our pair of interest. In cases where no such a model exists, we pick a model with either the source or the target language as our intended pair (Section 5). To show the utility of our transfer learning approach to the problem, we also train on one pair from scratch (which we treat as a *baseline*).

We experiment with tokenized **(primary models)** and untokenized **(contrastive models)** settings and compared the settings with models developed by fine-tuning pre-trained models as well as models trained from scratch. Our experiments show that the tokenized settings perform better than the untokenized settings for all language pairs. The model fine-tuned on top of the pre-trained MT model has higher performance than our baseline model from the first epoch compared with the model trained from scratch (for six epochs). Our models for the CA-ES and PT-ES language pairs *achieve top 1 rank in the offical shared task results*, with 82.96 and 47.71 BLEU scores respectively. In addition, we are the only team that submitted for the rest of the language pairs (i.e., ES-PT, FR-BM, and

[1] All models are available on https://github.com/fenimi/Similar-Languages-MT

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 273–278

BR-FR).

The rest of our paper is organized as follows: we discuss related work in Section 2. We describe the data and pre-processing in Section 3. Next, we describe the data cleaning process in Section 4. In Section 5, we describe the models we developed for this task and we discuss the various experiments we perform. We also describe the architectures of the models we developed. Then we discuss the evaluation criteria in Section 6. Evaluation is done on both the validation and test sets. In section 7 we perform error analysis on the output of our models for some language pairs to determine the types of errors the models make. We conclude with discussion of the insights we gained from the shared task in Section 8.

2 Related Work

In recent times, there has been an increase of research interest in low-resource MT scenarios (Jawahar et al., 2021; Baziotis et al., 2020). NMT models, specifically those based on the Transformer architecture, have been shown to perform well when translating between similar languages (Przystupa and Abdul-Mageed, 2019; Adebara et al., 2020; Barrault et al., 2019, 2020), low resource scenarios (Adebara et al., 2021), and in contexts not involving English (Fan et al., 2021).

Furthermore, pre-training techniques have been successful for many NLP tasks (Zoph et al., 2016; Durrani et al., 2021) including NMT (Aji et al., 2020; Weng et al., 2020). Self-supervised pre-training acquires general knowledge from a large amount of unlabeled monolingual or multilingual data to improve the training process of downstream tasks (Aji et al., 2020; Devlin et al., 2018). The pre-trained model acquires some syntactic and semantic knowledge which can be transferred as initialized parameters to improve NMT models and translation quality (Goldberg, 2019; Jawahar et al., 2019; Aji et al., 2020). The intuitive justification for using pre-trained models is that the embedding space becomes more consistent, with semantically similar words closer together.

The knowledge from pre-trained language models (LMs) can be used to initialize the NMT model before training it on parallel data. However, there are certain limitations for MT tasks. First, LMs cannot be easily fine-tuned for MT tasks. Second, there is a discrepancy between pre-training objectives for LMs and the training objective in MT. Existing pre-training approaches such as mBART rely on auto-encoding objectives to pre-train the models, which are different from MT. Furthermore, LMs learn to reconstruct all source tokens with some noises, while NMT learns to translate most source tokens and copy only a few of them. LM pre-training is said to copy about 65% of tokens, while NMT training needs to copy less than 10% (Knowles and Koehn, 2018). The unexpected knowledge/bias can be therefore propagated to the NMT model via pre-training, which may result in NMT models mistakenly copying source tokens to the target side (Liu et al., 2021). For instance, because copying behaviours can be learned, a source word such as "shoe" may be copied to the target by pre-training based NMT models instead of providing a translation. Therefore, fine-tuning MT models on pre-trained LMs still do not achieve adequate improvements.

In order to address the difference in training objectives that using pre-trained language models results in, we use pre-trained MT models to initialize our models. This is still a type of *transfer learning*.

Following the justification for pre-trained models, we hypothesize that two linguistically similar languages will share closer semantic and syntactic relationships. This is based on the assumption that the more similar the source and target languages, the more similar the syntax and semantic properties and the higher the gains from using pre-trained models will be. We now introduce our data.

3 Data

For our experiments, we use parallel data from OPUS (Tiedemann, 2012b). Our data are from the following language pairs Spanish and Catalan, Spanish and Portuguese, and French and Bambara. We use data in the two directions from each of these three pairs. Details about our data is in Table 1.

Pair	Lang	Sent	Words
ES-CA	**ES**	10M	284.6M
	CA	10M	273.3M
ES-PT	**ES**	4.1M	86.6M
	PT	4.1M	82.7M
FR-BM	**FR**	9.9K	179.3K
	BM	9.9K	202.9K

Table 1: Number of sentences and words for the training data used for each language pair. We also report the type token ratio (TTR) before and after tokenization.

3.1 Pre-Processing

We perform pre-processing using the Moses toolkit (Koehn et al., 2007). For each language not supported by Moses, we use the tokenization setting of the language it is translated to. This applies only to Bambara, for which we used tokenization for the French language. We perform data cleaning, as we explain next.

4 Data Cleaning & Analysis

We perform data cleaning on the ES-CA, CA-ES, ES-PT, and PT-ES language pairs. We do not clean the French and Bambara pairs because we had very few training sentences for these. For cleaning, we run the langid tool (Lui and Baldwin, 2012) on the concatenation of the source and target and remove sentences that are not identified as belonging to one or both of the language pair. In Table 2, we provide some examples of data points we remove from the training data during data cleaning. These examples are removed because the claimed language is different from the language predicted by langid. After cleaning, we are left with $\sim$ 10M clean sentences out of $\sim$ 18.3M sentences for the Spanish and Catalan pair, and $\sim$ 3.1M clean sentences out of $\sim$ 4.2M sentences for the Spanish and Portuguese pair, respectively. We note that removed data comprise large portions of each dataset, thus confirming our concerns about data quality.

Sentence	Claimed	Predicted
Animal Crossing: Pico de Santo Tomés	Spanish	English
Quinto Sereno Sammonico	Portug.	Portug.
La sombra del caudillo	Catalan	Italian
Cultura del Nepal	Catalan	Spanish
Morts a Rāwalpindi	Catalan	Spanish

Table 2: Examples removed from our training data. "Claimed" refers to the expected language as coming from source, while "predicted" is what langid.py identified.

5 Models

5.1 Primary and Contrastive Models

We developed our *primary* and *contrastive* models using Transformers from HuggingFace library (Wolf et al., 2019). The primary models were developed using tokenized data while the contrastive models employed untokenized data. For the tokenized setting, we used Moses tokenizer (as explained earlier) while the untokenized setting used the data just as they were made available to us by shared task organizers.

Hyperparameter	Values
encoder layers	6
decoder layers	6
attention heads	8
hidden layers	6
embedding dimension	512
dropout	0.0
vocab size	49,621

Table 3: Hyperparameter settings for the HuggingFace Marian Transformer models.

	Model	#Epochs	#Highest
FR-BM	tok	100	55
BM-FR	tok	100	60
ES-CA	untok	6	3
	tok	8	3
CA-ES	untok	7	7
	tok	8	8
ES-PT	untok	17	15
	tok	35	13
PT-ES	untok	18	16
	tok	34	23

Table 4: Description the number of epochs for training each model and the epoch with the highest BLEU score.

Pair	Untokenized	Tokenized
es-ca	77.3	86
ca-es	87.7	87.8
es-pt	46.9	47.3
pt-es	52.9	53.6
fr-bm	-	6.6
bm-fr	-	6.07

Table 5: BLEU scores on our Dev set.

We used the pre-trained NMT models developed by Helsinki-NLP on HuggingFace. We used pre-trained models closest to the language pairs we trained. For language pairs without existing pre-trained models, we used a close language pair with either the source or target matching one of our downstream task languages in a given pair. Specifically, we used the following Marian models released by Helsinki-NLP: es-ca (for ES-PT), ca-es (for CA-ES and PT-ES), fr-en (for FR-BM), and en-fr (for BM-FR).

As an example, we report the hyperparameters for the CA-ES *primary* model in Table 3. This model had the best BLEU and RIBES score for this

Pre-Trained Model	Pair	Untokenized System			Tokenized System		
		BLEU	RIBES	TER	BLEU	RIBES	TER
ca-es	ca-es	76.8	95.19	15.421	**82.79**	**96.98**	**10.918**
es-ca	es-pt	35.61	82.48	52.61	38.10	85.35	46.556
ca-es	pt-es	43.86	85.10	43.801	**47.71**	**87.11**	**39.213**
fr-en	fr-bm	-	-	-	1.32	24.79	97.89
en-fr	bm-fr	-	-	-	3.62	36.17	101.52

Table 6: BLEU, RIBES and TER Scores on the Test set for the Tokenized (primary) and untokenized (contrastive) configurations. The models for CA-ES and PT-ES language pairs were the best performing models highlighted in bold type.

Pair	Category	Text
ES-CA	**Source**	Por consiguiente, el Fondo debe movilizarse para aportar una contribución financiera en favor de Bulgaria, Grecia, Lituania y Polonia.
	Untok Output	Por lo tanto , el Fondo debe movilizarse para que se conceda una contribución financiera a Bulgaria, Grecia, Lituania y Polonia.
	Tok Output	Per tant, el Fons s'ha de mobilitzar per aportar una contribució financera a favor de Bulgaria, Grècia, Lituània i Polònia.
	Reference	Per tant, el Fons s'ha de mobilitzar per aportar una contribució financera en favor de Bulgària, Grècia, Lituània i Polònia.
CA-ES	**Source Text**	A fi de reduir al mínim el temps necessari per mobilitzar el Fons, aquesta Decisió s'ha d'aplicar a partir de la data de la seva adopció,
	Untok Output	A fin de reducir al mínimo el tiempo necesario para movilizar el Fondo, esta Decisión debe aplicarse a partir de la fecha de su adopción,
	Tok Output	Con el fin de reducir al mínimo el tiempo necesario para movilizar el Fondo, esta Decisión se ha de aplicar a partir de la fecha de su adopción.
	Reference	Con el fin de reducir al mínimo el tiempo necesario para movilizar el Fondo, la presente Decisión debe aplicarse a partir de la fecha de su adopción,
ES-PT	**Source Text**	Posición del Parlamento Europeo de 6 de abril de 2017 (pendiente de publicación en el Diario Oficial) y Decisión del Consejo de 11 de mayo de 2017.
	Untok Output	Posição do Parlamento Europeu de 6 de A bril de 2017 (pendente de publicação no Jornal Oficial) e D ecisão do Conselho de 11 de Maio de 2017.
	Tok Output	Posição do Parlamento Europeu de 6 de A bril de 2017 (indiferente à publicação no Jornal Oficial da União Europeia e decisão do Conselho de 11 de Maio de 201).
	Reference	Posição do Parlamento Europeu de 6 de abril de 2017 (ainda não publicada no Jornal Oficial) e decisão do Conselho de 11 de maio de 2017.
PT-ES	**Source Text**	Os Estados-Membros transmitem os dados referentes ao transporte por vias navegáveis interiores no seu território nacional à Comissão (Eurostat).
	Untok Output	Los Estados miembros transmitirán a la Comisión (Eurostat) los datos relativos al transporte por vías navegables interiores en su territorio nacional.
	Tok Output	Los Estados miembros transmitirán a la Comisión los datos relativos al transporte por vías navegables interiores en su territorio nacional (convenientREAT)
	Reference	Los Estados miembros transmitirán los datos relativos al transporte por vías navegables interiores en su territorio nacional a la Comisión (Eurostat).
FR-BM	**Source Text**	vous pourriez peut-être organiser de petits groupes pour lire et discuter de ce livre, chapitre par chapitre.
	Tok Output	- An y'a mɛn kura in mɛn: Mama denmuso, an ka jamana de wa k'a furu min ma kɔnɔ.
	Reference	aw bɛse ka to ka mɔgɔw dalajɛka gafe in kalan; ani ka hakilina falenfalen kɛsigidaw kan kelen kelen.
BM-FR	**Source Text**	Hakilijigin ka jɛsin kɛnɛyabaarakɛlaw ma
	Tok Output	cher agent de santé villageois,
	Reference	cher agent de santé villageois,

Table 7: Examples sentences from the various pairs and corresponding translations based on the untokenized and tokenized models. Examples are from the Dev set. We highlight the differences between the outputs from the untokenized model and the reference text with blue highlights and the differences between the tokenized model and the reference text in red highlights. It can be observed that the number of errors in the untokenized model (based simply on the number of blue highlights here) is larger than that in the tokenized model (less errors/red highlights

language pair. We trained each model for different number of epochs due to time and GPU constraints. We show the number of epochs each model is trained for and the epoch with the highest BLEU score in Table 4. We did not train any *contrastive* models for FR-BM and BM-FR pairs, so we report the training for the primary (tokenized) models only.

5.2 Baseline

We developed a single baseline model based on Transformers as implemented in Fairseq. This model does not use any pre-trained MT models nor tokenization. This model was developed for the ES-PT pair for six epochs and it achieved a BLEU score of 37.6. For comparison, we developed a model for the same pair (i.e., ES-PT) based on an already available pt-es pre-trained MT model. After six epochs, this ES-PT model employing transfer learning achieved 52.18 BLEU (thus significantly outperforming our baseline). Based on this result, we resumed with experiments for all other language pairs *without* including a baseline. Ideally, we would train such baseline models for all

the pairs. However, due to limited time and GPU resources, we only trained a baseline for a single pair.

6 Evaluation

We evaluated our models on both the Dev and Test sets. We used the checkpoint with the best BLEU score as evaluated on DEV as our best model. We used a beam size of four during evaluation on both Dev and Test and evaluated on de-tokenized data.

Evaluation on Dev set. We report the results on the Dev sets for each language pair in Table 5. As explained, the models were trained with both tokenized and untokenized data. As Table 5 shows, the tokenized setting yielded the highest performance for *all* language pairs. We show sample outputs from our tokenized models (from Dev data) in Table 7.

Evaluation on Test set. Our Test set performance was evaluated by the shared task organizers using BLEU (Papineni et al., 2002), RIBES, and TER (Snover et al., 2009). We report the scores in Table 6. Each of our CA-ES and PT-ES models ranked top 1 based on the official shared task results. In addition, we were the only team to submit models in the official competition for the French-Bambara pairs. As with the Dev set, the tokenized setting gave the highest performance for all language pairs.

7 Effect of Language Similarity

In order to gain some insight into the interference of similarity between languages of a given pair, we performed an analysis based on Levenstein distance that allows us to identify the percentage of cognates shared between the languages. We then compared system output to the reference sentences, trying to quantify how much the system is able to translate cognates correctly (in this case the correct translation will have the same cognate word in the target as it is in the source). We performed this analysis for one language pair: CA-ES and found that the model learned the cognates correctly up to 80% of the time.

8 Conclusion

We describe our contribution to the WMT2021 Similar Languages Translation Shared Task. We develop models for ES-CA, CA-ES, ES-PT, PT-ES, FR-BM, and BM-FR and show the improvement our models make with tokenized data when compared to untokenized data. We also show the utility of transfer learning based on fine-tuning NMT pre-trained models. Future work can investigate how the choice of pre-trained models affects the downstream tasks.

Acknowledgements

We gratefully acknowledge support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Social Sciences Research Council of Canada (SSHRC), Compute Canada (www.computecanada.ca), and UBC ARC–Sockeye (https://doi.org/10.14288/SOCKEYE).

References

Ife Adebara, Muhammad Abdul-Mageed, and Miikka Silfverberg. 2021. Translating the unseen? yoruba-english mt in low-resource, morphologically-unmarked settings. *arXiv preprint arXiv:2103.04225*.

Ife Adebara, El Moatez Billah Nagoudi, and Muhammad Abdul Mageed. 2020. Translating similar languages: Role of mutual intelligibility in multilingual transformers. In *Proceedings of the Fifth Conference on Machine Translation*, pages 381–386, Online. Association for Computational Linguistics.

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710.

Loïc Barrault, Magdalena Biesialska, Ondrej Bojar, M. Costa-jussà, C. Federmann, Yvette Graham, Roman Grundkiewicz, B. Haddow, M. Huck, E. Joanis, Tom Kocmi, Philipp Koehn, Chi kiu Lo, Nikola Ljubesic, Christof Monz, Makoto Morishita, M. Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *WMT@EMNLP*.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussa, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Christos Baziotis, Barry Haddow, and Alexandra Birch. 2020. Language model prior for low-resource neural machine translation. *arXiv preprint arXiv:2004.14928*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. How transfer learning impacts linguistic knowledge in deep nlp models? *arXiv preprint arXiv:2105.15179*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond English-Centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Yoav Goldberg. 2019. Assessing Bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2021. Exploring text-to-text transformers for English to Hinglish machine translation with synthetic code-mixing. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 36–46, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does Bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Rebecca Knowles and Philipp Koehn. 2018. Context and copying in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3034–3041.

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 companion volume proceedings of the demo and poster sessions*, pages 177–180.

Xuebo Liu, Longyue Wang, Derek F Wong, Liang Ding, Lidia S Chao, Shuming Shi, and Zhaopeng Tu. 2021. On the copying behaviors of pre-training for neural machine translation. *arXiv preprint arXiv:2107.08212*.

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.

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 ACL*, pages 311–318. ACL.

Michael Przystupa and Muhammad Abdul-Mageed. 2019. Neural machine translation of low-resource and similar languages with backtranslation. In *Proceedings of the 4th Conference on MT (Volume 3: Shared Task Papers, Day 2)*, pages 224–235.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or hter? exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268.

Jörg Tiedemann. 2012a. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

Jörg Tiedemann. 2012b. 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 Languages Resources Association (ELRA).

Jörg Tiedemann, Filip Ginter, and Jenna Kanerva. 2015. Morphological segmentation and opus for finnish-english machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 177–183.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. 2020. Acquiring knowledge from pre-trained model to neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9266–9273.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

T4T Solution: WMT21 Similar Language Task for the Spanish-Catalan and Spanish-Portuguese Language Pair

Miguel Canals
miguelknals@gmail.com

Marc Raventós Tato
marcraven@gmail.com

Abstract

This system description describes the participation in the EMNLP 2021 Sixth Conference on MT (WMT21) - Shared Task: Similar translation for the language pairs SPA<>CAT and PTG<>SPA for our T4T solution. The main objective has been to prove that good data with a good standard NMT toolkit, as OpenNMT, is able to provide good results. We have focus in the corpus cleaning (both from the physical and from the statistical side), try to find some alternatives to subword segmentation (syllabic and byte-pair-enconding), and finally use OpenNMT as out-box system with a transformer model. The results have been pretty close to the best ones, if not the best.

1 Introduction

Current available NMT systems have become so complex and resource computing demanding, that the idea behind this project is try to find out if simple logical solutions and standard tools are able to provide good results at least in close languages (according Ethnologue Lexical similarity coef for the language pairs are 0.85 CA<>ES and 0.89 for PT<>ES) (Collin, 2010).

The first thing that come with a little surprise is how we can explain that so similar languages have persistently get so different BLEU (Papineni et al., 2002) scores in previous WMT years, as stated in Table 1 for WMT2020.(Barrault et al., 2020)

ES-CA	CA-ES	ES-PT	PT-ES
86.44	77.08	32.69	33.82

Table 1: Results WMT 2020 for similar languages CA-ES/PT-ES best BLEU score.

We suspect this 50 BLEU score difference is direct result of corpus quality or diversity. CA<>ES corpus provided by the organization uses a very reliable source, the DOG (The official Catalan Government Diary) (approx 40% of words), and even though PT<>ES uses also a similar domain (mainly news and legal), its legal composition (Europarl/JARC) is based in probably a mix of indirect translations. We think one of the best ways to improve a NMT system, is to use the best data you can.

We have have focused in the physical cleaning of the corpus (duplicate strings, unusual sentences, tokenized text in some sources, deal with the UTF-8 universe coding for punctuation, numerical data and the upper/lower casing issue). We have developed a set of python programs for these cleaning tasks and an adhoc tokenizer.

We also have tried also to run some cleaning procedure based in some basic statistical information of the bitext corpus. As there is a quite large source-target, we have scored word probabilities in bitext corpus sentences, and then somehow score sentence probabilities and decide to use or not these sentences. This simple cleaning has indeed increased the score of the model for corpus in-data, but is not so clear if it helps with data out of the corpus.

The last step in data preparation, to deal with the vocabulary size issue, has been the subword segmentation. We have used python standard tools for syllabic segmentation with good results (corpus data has achieved best score than BPE (byte-pair-enconding), but again, with data outside of the corpus, BPE (Sennrich et al., 2015) has proven better. At the end, we have used Google SentencePiece (Kudo and Richardson, 2018) BPE implementation.

After that, we have used what we think a proven toolkit for NMT, OpenNMT (Klein et al., 2017), out of the box, without any modification, using its web publish options for the Transformer model. In the last step we have used the inverse python programs in order to generate the final version of the test source translated file.

We have focused the system from a practical engi-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 279–283

neering point of view. The whole project has been based in currently available local only mid size system, with consumer grade multi-gpu environment. Following this fast approach, and due the nature of the main focus on the corpus, we have used simple models that OpenNMT provides (2-layer LSTM with 500 hidden units on both the encoder and decoder) in order to choose several options and parameters used later for the final transformer model, as this last model is close to the limit a midsize system can provide.

2 Cleaning the corpus

In our approach we have joined all data sources (all monolingual sources and the matching language for the bilingual text corpus) in order to create a mono corpus, and the bilingual text corpus. For the model training we have used the dev data provided by the organization. The typical size for this file in OpenNMT is around 2000 lines, so we have used data from the bilingual text corpus in order to reach this typical size.

2.1 Physical cleaning steps for the bilingual corpus

These are the direct "physical" tasks in order to prepare a corpus with what they look "standard" sentences.

- Removal of duplicated sentences.
- As many strings are already tokenized we have detokenize all the corpus with the Moses (Koehn et al., 2007) detokenizer, as our custom tokenizer works with untokenized text. (Some punctuation is changed, but indeed fixes many punctuation format errors as coma not correctly joined to the words).
- Perform physical cleaning. These are some steps based on the manual inspection of the corpus in order to remove noise sentences or fix others. For instance, remove all left chars until an alphabetic char is found, remove some keywords leftovers, sentences should have at least a spell correct word (Németh, 2010), remove any text between parenthesis or remove duplicates.
- Using python nltk (Bird et al., 2009) package we have removed all sentences that probably are made up of more than one sentence.

2.2 Statistical cleaning for the bilingual corpus

Using the bitext corpus we have created source/target dictionary and all instances of where source word and target word appears in the same sentence pair. Using simple rules we can try to score the probability of the source word given a target word, and somehow score words and sentences. Then create a list and remove the ones with worst scores. Most of this cleaning is based in heuristic parameters.

The clean is indeed effective as for instance, score for corpus data for PT->ES using this cleaning can raise from 49,38 BLEU score to 67.27, but these gains are not matched when we have used test data outside the corpus. We suspect, this cleaning creates an ideal statistical data set that cannot explain "real" data outside the corpus. So we have used this feature in a moderate way, removing 15% of the low matching sentences. Many of these sentences are indeed removed for a good reason, but many times too are not, because translators many times do not follow a statistical behavior.

We suspect this is an open field. This "cleaning" is close related to word alignment, so probably it would have been wiser use some GIZA++(Och and Ney, 2003) or fastalign word based solution.

3 Tokenization

One of the big issues to deal with real data, is the tokenization. After reviewing several available tools, we ended creating a python custom tokenizer that has the following features.

- It uses a list of split chars (comma, dot, hyphen, ...). The number of these chars that are not alphabetic can be quite large, and is a source of many problems. This list of split chars is generated by the tokenizer itself in a first scanning phase.
- Numbers are replaced by variables (as ((n0)), ((n1)). These numbers are kept in an independent file in order to be used if detokenization is required. This will avoid the use of numbers, another big source of undesired vocabulary.
- Casing is indicated with special tags before the upper word in to ways, ((up)) for first uppercase only first letter words, or ((aup)) for

all uppercase letter words. This avoids most of casing issues and allow us to work with a full lowercase input in the neuronal toolkit.

- Tokenizer also keeps track of the spaces for words and split chars.

These features provide a robust tokenization <> detokenization reversibility.

Sentence example:

> Vista la Directiva 91/494/CEE del Consejo, de 26 de junio de 1991, sobre ...

is tokenized as:

> ((up)) vista la ((up)) directiva ((n0)) @@/@@ ((n1)) @@/@@ ((aup)) cee del ((up)) consejo @@, de ((n2)) de junio de ((n3)) @@, sobre

4 Word segmentation: BPE and syllabic

Word segmentation is further step in order to to reduce neuronal network vocabulary.
We have followed two approaches, the well know BPE subword segmentation, but also an uncommon one, a syllabic segmentation. We have used again a known python tool (https://pyphen.org/) to split words in syllables.
Results are quite interesting as they have been quite consistent. Using a syllabic segmentation:

- BLEU scores for corpus test data in all NTM models (LSTM or Transformer) have been better.
- BLEU scores for external data have been worst.

So the promising syllabic segmentation, has not responded so well with data outside de corpus. Due this, BPE has been chosen for final models.

5 Evaluation

After testing in more simple neuronal network models (LSTM) the final setup has consisted of a corpus cleaned (in the physical sense, and also in an statistical sense removing approx 15% of sentence corpus sentences with the highest perplexity). This clean corpus has been detokenized with our adhoc tokenizer (that lowercase the corpus, replaces numbers by variables, and handles punctuation and upper/lower casing).

After this cleaning, the number of words for ES<>PT has been around 2M lines (55.3M words) and ES<>CA around 9.5M lines (176M words).

Then we have used 16000 terms SentencePiece BPE vocabulary on this detokenized corpus in order to reduce vocabulary. We have removed sentences with more than approx 170 tokens for the sentences the neuronal network has ingested (This length has kept the model below the memory limit of each one of the GPU cards).

We have set the model configuration using the published Transformer(Uszkoreit, 2017) model in the OpenNMT site (https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model). According OpenNMT documentation, this setup mimic the Google (Vaswani et al., 2017) setup that replicate its WMT results.

We have tested our models against test data form our corpus (23_SP_TRANSF_Statclean_3,5 for PT<>ES and 25_SP_CAES_2_TRANSF_Statclean_3.5 for CA<>ES) and also from the test data from WMT2020 (test20 for PT<>ES and test20.v2 CA<>ES).

In Figure 1 we can see the results for the PT<>ES for both test sets and both directions. The transformer model converges really fast after 30-40K steps (as the size of the corpus is not very large). We have used the best score (PT->ES BLEU score 55.96 at 55K steps and ES->PT BLEU score 54.68 at 60K steps) for the final evaluation.

In Figure 2 we can see the results for the CA<>ES for both test sets and both directions. We have used the best score (CA->ES BLEU score 84.34 at 70K steps and ES->PT BLEU score 83.77 at 85 steps) for the final.

6 Results

In Table 2 we can compare the best score of each one of the 3 teams that have submitted results for this WMT 2021 task (http://www.statmt.org/wmt21/similar.html).

7 Conclusions

We think we have accomplish the objective to achieve good results with good data and out of box toolkit as OpenNMT.
It has proven more difficult than expected find recipes to improve the corpus quality beyond the physical cleaning. What we have found suggest (without prove) that:

- Cleaning the corpus trying to remove sentences with low translation probability to be correct looks to us that can improve the corpus for sure, but is not so clear will happen the same for data outside the corpus. The idea of find correct paired translated sentences in the bitext, is a translation/alignment problem by itself, and probably the simple statistical system we have used has much room to improve.
- Syllabic word sub segmentation can improve greately the corpus quality, but has not improved the score with data outside the corpus. The reason is unknown.

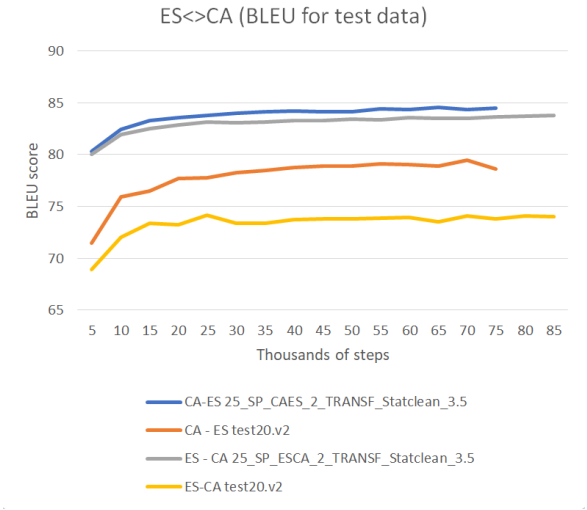


Figure 1: Results PT<>ES BLEU score for test data from the corpus and external to the corpus

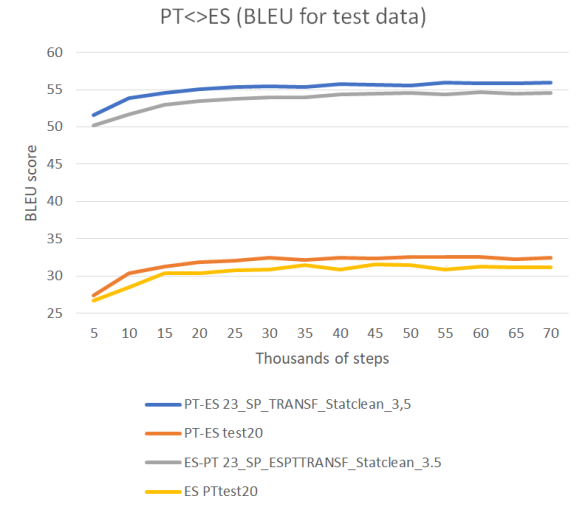


Figure 2: Results CA<>ES BLEU score for test data from the corpus and external to the corpus

	BLEU		RIBES		TER	
	Best score	T4T	Best score	T4T	Best score	T4T
PT-ES	47.71	46.29	87.11	87.04	39.21	40.12
ES-PT	**40.74**	**40.74**	**85.69**	**85.69**	**43.34**	**43.34**
CA-ES	82.79	77.93	96.98	96.04	10.92	16.5
ES-CA	79.69	78.60	**96.24**	**96.24**	14.63	16.13

Table 2: Results for the bests system and T4T

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Richard Oliver Collin. 2010. Ethnologue. *Ethnopolitics*, 9(3-4):425–432.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn, Marcello Federico, Wade Shen, Nicola Bertoldi, Ondrej Bojar, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, Richard Zens, et al. 2007. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. In *CLSP Summer Workshop Final Report WS-2006, Johns Hopkins University*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

László Németh. 2010. Hunspell. *Dostupno na: http://hunspell. sourceforge. net/[01.10. 2013]*.

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 of the Association for Computational Linguistics*, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Jakob Uszkoreit. 2017. Transformer: A novel neural network architecture for language understanding. *Google AI Blog*, 31.

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.

Neural Machine Translation for Tamil–Telugu Pair

Sahinur Rahman Laskar, Bishwaraj Paul, Prottay Kumar Adhikary
Partha Pakray, Sivaji Bandyopadhyay
Department of Computer Science and Engineering
National Institute of Technology Silchar
Assam, India
{sahinur_rs, bishwaraj_ug, prottay_ug, partha}@cse.nits.ac.in
sivaji.cse.ju@gmail.com

Abstract

The neural machine translation approach has gained popularity in machine translation because of its context analysing ability and its handling of long-term dependency issues. We have participated in the WMT21 shared task of similar language translation on a Tamil-Telugu pair with the team name: CNLP-NITS. In this task, we utilized monolingual data via pre-train word embeddings in transformer model based neural machine translation to tackle the limitation of parallel corpus. Our model has achieved a bilingual evaluation understudy (BLEU) score of 4.05, rank-based intuitive bilingual evaluation score (RIBES) score of 24.80 and translation edit rate (TER) score of 97.24 for both Tamil-to-Telugu and Telugu-to-Tamil translations respectively.

1 Introduction

Machine translation (MT) works as an interface that handles language ambiguity concerns via automatic translation between two different languages. Neural machine translation (NMT) attains state-of-the-art results for both high and low-resource language pairs translation (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Pathak et al., 2018; Pathak and Pakray, 2018; Laskar et al., 2019; Laskar et al., 2020a). The NMT utilizes an artificial neural network to predicts the likelihood of a sequence of words. But NMT requires a sizeable parallel corpus to get effective MT output, challenging for low-resource pair translation. In this WMT21 shared task, we have participated on a similar language pair translation task of Tamil–Telugu pair using NMT. We aim to utilize similarity features among such a similar language pair and monolingual data to overcome the less availability of parallel corpus. The transformer model (Vaswani et al., 2017) based NMT is considered in this work, since it outperforms RNN based NMT. Moreover, NMT performance can be enhanced utilizing monolingual data (Weng et al., 2019; Wu et al., 2019; Ramachandran et al., 2017; Variš and Bojar, 2019; Qi et al., 2018). To evaluate the performance of our system's output, WMT21 organizer used standard evaluation metrics, namely, BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006) which are reported in Section 4.

2 Related Work

There are limited works on the Tamil–Telugu pair (Chakravarthi et al., 2021). The literature survey found similar works on Indian similar language pairs, such as Hindi–Nepali (Laskar et al., 2019) and Hindi–Marathi (Laskar et al., 2020b) at WMT19 and WMT20. Both (Laskar et al., 2020b, 2019) used transformer model based NMT. Moreover, Ramachandran et al. (2017); Variš and Bojar (2019); Qi et al. (2018) pre-trained methods are incorporated in NMT to utilize advantage of monolingual data for low-resource pairs translation. In this work, GloVe (Pennington et al., 2014) pre-trained word embeddings are used in transformer model (Vaswani et al., 2017) based NMT for both Tamil-to-Telugu and Telugu-to-Tamil translation.

3 System Description

Our system mainly consists of the following parts: data preprossessing, model training and testing. These have been described in the following subsections. The dataset description is presented in Section 3.1. For our system, we have used the OpenNMT-py toolkit (Klein et al., 2017) for the data preprocessing, training and testing.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 284–287
November 10–11, 2021.

Corpus	Type	Sentences	Tokens		Source
			Tamil	Telugu	
Parallel	Train	40147	588919	625308	WMT21 Organizer
	Validation	1261	25443	25844	
	Test	1735	33911	35895	
Monolingual	Tamil	31542481	488507451		IndicNLP
	Telugu	47877462	574131374		

Table 1: Dataset Statistics

3.1 Dataset

The parallel corpus for Tamil-Telugu pair is provided by the WMT21 organizer[1]. It consists of 40147, 1261, 1735 sentence pairs for train, validation and test set. Apart from this, we also collected Monolingual data from the IndicNLP[2] corpus. It consists of 31542481 Tamil sentences and 47877462 Telugu sentences. This monolingual corpus is specifically used for deriving pretrained embeddings to use in the model. The dataset statistics are described in the table 1.

3.2 Data Preprocessing

The OpenNMT-py toolkit is used to preprocess the parallel data and then generates a vocabulary of size: 50002 for the source-target sentences by tokenizing and indexing in a dictionary. It was done in both ways independently, considering Tamil as source and Telugu as target and then with Telugu as source and Tamil as the target to train models for both ways for translation in either direction. We have used GloVe (Pennington et al., 2014) to pretrain on the monolingual corpora to obtain word vectors. These word vectors are specifically used in the form of word embeddings in the transformer model during the training process.

3.3 System Training

After the data preprocessing, the pre-trained embeddings and parallel dataset are used for training our model for both Tamil-to-Telugu and Telugu-to-Tamil. We have adopted a transformer model to implement both of the trained models separately. The transformer model consists of a self-attention mechanism, encoder, and decoder layers. The self-attention comes into play, where the relevancy of one word to other words of the sentence is represented as an attention vector that contains the context between words in that sentence. Multiple such attention vectors are calculated, and the weighted average is taken so that the interactions with other words are captured properly rather than their value. More specifically, the embeddings are converted into three spaces: query, key, and value. The dot product of its query vector and all the key vectors are calculated for every embedding. Since the hidden state of the previous embedding is not needed in calculating the current embedding's hidden state, the self-attention can be done in parallel for all embeddings. Thus, it can be run in parallel for all embeddings simultaneously. This speeds up the training and translation process a lot. Now, the target sentences are passed to the decoder layers similarly to the encoders and then passed to the self-attention block. The difference is that in attention layers, the next word of the target sentence is masked so that the word will be predicted using previous results for learning. It is called a masked multi-head attention block. The attention vectors thus produced and the outputs from the encoder layers are then passed to another attention block called encoder-decoder attention block. The attention vectors for every word in the sentences are the output. Then we pass it through a feed-forward network for making output acceptable for further layers.

Our transformer model consists of six layers for both encoders and decoders and eight attention heads. We used adam optimizer with a learning rate 0.001 and a drop-out of 0.1 for normalization. The rest of the parameters were selected as the default configuration of the toolkit. This configuration is used for both models, the Tamil-to-Telugu and vice versa.

3.4 System Testing

The obtained trained models are used in system testing, where the test data is used to obtain the predicted translation for both Tamil-to-Telugu and vice-versa independently.

[1] http://statmt.org/wmt21/similar.html

[2] https://indicnlp.ai4bharat.org/corpora/

Translation	System Type	BLEU	RIBES	TER
Tamil to Telugu	Primary	4.05	24.80	97.24
Telugu to Tamil	Primary	4.05	24.80	97.24

Table 2: Our System results for Tamil-Telugu pair at WMT21

4 Result

Our system's outputs were submitted to the organizer for evaluation. Consequently, the results of the shared task on "Similar Language Translation" were announced separately for Tamil-to-Telugu[3] and Telugu-to-Tamil[4]. The ranking of the systems is mainly based on BLEU score, while the RIBES and TER scores are also given. Our team name is CNLP-NITS. For the Tamil-to-Telugu translation system, we achieved 4th rank with a BLEU score of 4.05 and 6th rank with a BLEU score of 4.05 for the Telugu-to-Tamil translation. The results of our system are reported in the Table 2. The system performance is identical for both translation directions. We need to perform a human evaluation in future work to identify the test set and predicted output are identical or not.

5 Conclusion and Future Work

This work reports our system description along with results, which we have participated in the WMT19 shared task of similar language pair: Tamil-Telugu. Both direction of translations, transformer model based NMT is used and utilized monolingual data through pre-trained word embeddings. We will investigate multilingual NMT approach in future to improve such low-resource translation quality.

Acknowledgement

Authors would like to thank WMT21 Shared task organizers for organizing this competition and also, thank Center for Natural Language Processing (CNLP) and Department of Computer Science and Engineering at National Institute of Technology, Silchar for providing the requisite support and infrastructure to execute this work.

[3] https://mzampieri.com/workshops/wmt/2021/TA_TE.pdf

[4] https://mzampieri.com/workshops/wmt/2021/TE_TA.pdf

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Shubhanker Banerjee, Richard Saldanha, John P. McCrae, Anand Kumar M, Parameswari Krishnamurthy, and Melvin Johnson. 2021. Findings of the shared task on machine translation in Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 119–125, Kyiv. Association for Computational Linguistics.

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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. 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.

S. R. Laskar, A. Dutta, P. Pakray, and S. Bandyopadhyay. 2019. Neural machine translation: English to hindi. In *2019 IEEE Conference on Information and Communication Technology*, pages 1–6.

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, and Sivaji Bandyopadhyay. 2020a. EnAsCorp1.0: English-Assamese corpus. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 62–68, Suzhou, China. Association for Computational Linguistics.

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, and Sivaji Bandyopadhyay. 2020b. Hindi-Marathi cross lingual model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 396–401, Online. Association for Computational Linguistics.

Sahinur Rahman Laskar, Partha Pakray, and Sivaji Bandyopadhyay. 2019. Neural machine translation: Hindi-Nepali. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 202–207, Florence, Italy. 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.

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.

Amarnath Pathak and Partha Pakray. 2018. Neural machine translation for indian languages. *Journal of Intelligent Systems*, pages 1–13.

Amarnath Pathak, Partha Pakray, and Jereemi Bentham. 2018. English–mizo machine translation using neural and statistical approaches. *Neural Computing and Applications*, 30:1–17.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, Doha, Qatar. ACL.

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, New Orleans, Louisiana. Association for Computational Linguistics.

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, Copenhagen, Denmark. 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.

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, page 3104–3112, Cambridge, MA, USA. MIT Press.

Dušan Variš and Ondřej Bojar. 2019. Unsupervised pretraining for neural machine translation using elastic weight consolidation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 130–135, Florence, Italy. 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.

Rongxiang Weng, Heng Yu, Shujian Huang, Weihua Luo, and Jiajun Chen. 2019. Improving neural machine translation with pre-trained representation. *CoRR*, abs/1908.07688.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Low Resource Similar Language Neural Machine Translation for Tamil-Telugu

Vandan Mujadia and Dipti Misra Sharma
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
vandan.mu@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

This paper describes the participation of team oneNLP (LTRC, IIIT-Hyderabad) for the WMT 2021 task, similar language translation[1]. We experimented with transformer based Neural Machine Translation and explored the use of language similarity for Tamil-Telugu and Telugu-Tamil. We incorporated use of different subword configurations, script conversion and single model training for both directions as exploratory experiments.

1 Introduction

Machine Translation (MT) is a field of Natural Language Processing which aims to translate a text from one natural language to another. The meaning of the source text must be fully preserved in the resulting translated text in the target language. Recent years have seen significant quality advancements in machine translation with the advent of Neural Machine Translation. For the translation task, different types of machine translation systems have been developed and they are mainly categorized into Rule based Machine Translation (RBMT)(Forcada et al., 2011), Statistical Machine Translation (SMT) (Koehn, 2009) and Neural Machine Translation (NMT) (Bahdanau et al., 2014).

Neural machine translation (NMT) shows high quality in terms of output fluency and translation quality, when large amounts of parallel data are available (Barrault et al., 2020). Unfortunately, for most language pairs, parallel data is either scare or non-existent. To overcome this, unsupervised MT (UMT) (Artetxe et al., 2020) focuses on utilising monolingual data to generate synthetic parallel training data. Other techniques like back-translation(Sennrich et al., 2015),(Hoang et al., 2018), (Feldman and Coto-Solano, 2020) or denoising(Kim et al., 2019) also rely on parallel corpora of other language pairs and/or large quantities of monolingual data.

This paper describes our experiments for very low resourced similar language translation. For our work, we focused only on Tamil-Telugu language pair (both directions) and participated in a constrained setting.

We experimented only with Transformer (Vaswani et al., 2017) based Neural Machine Translation throughout. Along with it, to tackle high agglutination of both languages, we explored the morph (Virpioja et al., 2013) induced sub-word segmentation with byte pair encoding (BPE)(Sennrich et al., 2016).

Similar to Multilingual Neural Machine Translation (MNMT), we explored the use of a tag trick, where a token like "< 2xx >" (xx is language code) is prefixed to each source sentence to indicate the desired target language(Dabre et al., 2020). Here, we trained a single model for both directions (Tamil-Telugu and Telugu-Tamil) on given parallel data and monolingual data under MNMT setting.

The sections of the paper are organised as following: Section 2 describes Data, Section 3 and 4 describe pre-processing and Training Configuration and in Section 5 we talk about results and we conclude in section 6.

2 Data

We utilised provided parallel corpora for Tamil<->Telugu MT task. Apart form parallel corpus, we randomly selected 0.1M monolingual corpora from IndicCorp monolingual corpus[2] for Tamil and Telugu. Table-1 describes the training and development data (parallel and monolingual) used in all our experiments under constrained setting.

[1] https://www.statmt.org/wmt21/similar.html

[2] https://indicnlp.ai4bharat.org/corpora/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 288–291

Data	Sents	Token	Type
Train			
Tamil (Parallel)	40,147	0.68M	74K
Telugu (Parallel)	40,147	0.72M	90K
Development			
Tamil (Parallel)	1261	29K	9K
Telugu (Parallel)	1261	30K	10K
Tamil (Mono)	0.1M	-	-
Telugu (Mono)	0.1M	-	-

Table 1: Tamil-Telugu WMT2021 Training data

3 Data Pre-Processing

To tokenize and clean both Tamil and Telugu corpora (train, test, valid and monolingual), we used IndicNLP Tool[3] with in-house tokenizer as a first step. Following subsections explain other pre-processing steps of experiments.

3.1 Morph + BPE Segmentation

Based on token/type ratio, both Tamil and Telugu are morphologically rich languages from Table-1. Translating from (and to) morphologically-rich agglutinative language is more difficult due to their complex morphology and large vocabulary. We address this issue with morphology and BPE(Sennrich et al., 2016) based segmentation method as prescribed in (Mujadia and Sharma, 2020). We utilized unsupervised Morfessor (Virpioja et al., 2013) by training it on monolingual data of Tamil and Telugu. We then applied this trained Morfessor model on our corpora (train, test, development) to get meaningful stem, morpheme, suffix segmented sub-tokens for each word in a sentence. Subsequently, we applied the subword algorithm on top of the morph segmentation and used the derived sequence in training.

3.2 Training as Multilingual Neural Machine Translation (MNMT)

As an exploratory experiment, we configure a similar low resource machine translation problem as a multilingual machine translation problem. For both translation directions (Tamil-Telugu and Telugu-Tamil) we trained a single model to take advantage of language similarity among these languages. First, we converted both languages into Roman script using litcm[4]. Second, we prefixed "<2TE>" for Tamil to Telugu and "<2TA>" for Telugu to Tamil to the respective source sentences. Apart from this, we also utilised monolingual data as a monolingual translation. For this we prefixed "<2TE>" for Telugu to Telugu and "<2TA>" for Tamil to Tamil translation.

4 Training Configuration

Throughout all experiments, we used Transformer sequence to sequence architecture with the following configuration.

- Morph + BPE based subword segmentation, Embedding size : 512 Transformer for encoder and decoder, rnn_size 512, heads 4 encoder - decoder layers : 2, label smoothing : 1.0, dropout : 0.30, Optimizer : Adam, Beam size : 4 (train) and 10 (test), training steps : 20K

For these experiments, we used shared vocab across trainings. We used Opennmt-py (Klein et al., 2020) toolkit with above configuration for our experiments.

Using the above described pre-processing and configuration, we performed experiments on word level, BPE level and morph + BPE level for input and output. The results are discussed in following Result section.

5 Result

Feature	**BPE**	**Dev**
Script Conversion (ta to te)	-	0.57
Word	-	5.12
BPE	20K	6.07
Morph + BPE	20K	6.25
Morph + BPE (MNMT)	20K	6.65

Table 2: BLEU scores for Tamil-Telugu on Development set. BPE stands for byte pair encoding (subword), Morph for Morphological segment and MNMT for Multilingual Neural Machine Translation based method as discussed in Section-3.2

Table-2 and Table-3 show performance of our systems with different configurations in terms of BLEU score (Papineni et al., 2002) for Tamil-Telugu and Telugu-Tamil respectively on the development data. To get trivial, non-translation baseline, we used aksharamukha[5] script conversion

[3]http://anoopkunchukuttan.github.io/indic nlp library/
[4]https://github.com/irshadbhat/litcm
[5]https://aksharamukha.appspot.com/converter

Feature	BPE	Dev
Script Conversion (te to ta)	-	0.41
Word	-	5.72
BPE	20K	6.37
Morph + BPE	20K	6.45
Morph + BPE (MNMT)	20K	6.76

Table 3: BLEU scores for Telugu-Tamil on Development set. BPE stands for byte pair encoding (subword), Morph for Morphological segment and MNMT for Multilingual Neural Machine Translation based method as discussed in Section-3.2

tool to convert script from Tamil-Telugu (both direction). We achieved highest 6.65 and 6.76 development and 3.67 and 5.03 test BLEU scores for Tamil-Telugu and Telugu-Tamil systems respectively (all are of MNMT based systems).

Table-2 and Table-3 show that non-translation baselines are also low in terms of BLEU scores which indicates that the task much harder even though languages are similar. The results show that for low resource settings, transformer network based MT models can be improved with morph based segmentation along with byte pair encoding for morph rich languages. Also, forming it as a Multilingual machine translation problem, along with monolingual data, it improves the quality of MT models. This may be due to language similarity and use of monolingual data, as it is helping models to do better generalization by learning better source language encoding and target language fluency.

6 Conclusion

From our experiments, we conclude that linguistic feature such as morph based segmentation with subword segments along with MNMT is a promising approach for similar language translation.

References

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. *arXiv preprint arXiv:2004.14958*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, et al. 2020. Proceedings of the fifth conference on machine translation. In *Proceedings of the Fifth Conference on Machine Translation*.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976.

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.

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.

Yunsu Kim, Jiahui Geng, and Hermann Ney. 2019. Improving unsupervised word-by-word translation with language model and denoising autoencoder. *arXiv preprint arXiv:1901.01590*.

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The opennmt neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pages 102–109.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Vandan Mujadia and Dipti Misra Sharma. 2020. Nmt based similar language translation for hindi-marathi. In *Proceedings of the Fifth Conference on Machine Translation*, pages 414–417.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

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.

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.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Similar Language Translation for Catalan, Portuguese and Spanish Using Marian NMT

Reinhard Rapp
Athena Research Center
Magdeburg-Stendal University of Applied Sciences
University of Mainz
reinhardrapp@gmx.de

Abstract

This paper describes the SEBAMAT contribution to the 2021 WMT Similar Language Translation shared task. Using the Marian neural machine translation toolkit, translation systems based on Google's transformer architecture were built in both directions of Catalan–Spanish and Portuguese–Spanish. The systems were trained in two contrastive parameter settings (different vocabulary sizes for byte pair encoding) using only the parallel but not the comparable corpora provided by the shared task organizers. According to their official evaluation results, the SEBAMAT system turned out to be competitive with rankings among the top teams and BLEU scores between 38 and 47 for the language pairs involving Portuguese and between 76 and 80 for the language pairs involving Catalan.

1 Introduction

In recent years, neural machine translation (NMT) has become the state of the art in machine translation (MT). Using toolkits such as Marian NMT (Junczys-Dowmunt et al., 2018), it is relatively straightforward to construct end-to-end NMT systems which need only little pre-processing of the training corpora and post-processing of the system output. As NMT is a supervised approach to MT based on machine learning technology, training is usually conducted using sentence aligned human translations. Given a large number of source/target-language sentence pairs, the neural system fully automatically learns how to translate.

The SEBAMAT submission to the Similar Language Translation (SLT) task[1] of the 6th Conference on MT is based on work conducted as part of the SEBAMAT project[2] (semantics-based machine translation; Rapp & Tambouratzis, 2020). This project has a focus on experiments introducing semantics into MT but, for comparative purposes, also deals with standard NMT systems. The latter were used as the basis for the current shared task. During the SEBAMAT project a number of MT systems had been developed for language pairs involving English, French, German, Greek and Spanish, but there had been no prior work on Catalan and Portuguese. The aims of the participation in the SLT shared task were the following:

- See in how far the SEBAMAT-based MT systems are competitive.
- Extend the number of SEBAMAT languages by Catalan and Portuguese.
- Find out whether systems for new language pairs can be developed in a very short time.
- See whether reasonably well working systems can be developed without much proficiency of the respective languages on the developer side.

2 Resources

2.1 Corpora

For the training of the NMT systems sentence-aligned parallel corpora are required. We used all parallel corpora suggested by the SLT shared task organizers who in their task description explicitly stated that no additional parallel corpora were allowed for training.

For Catalan–Spanish the following parallel corpora were used:

- Wiki Titles v3 (476,475 sentence pairs) (Barrault et al., 2020)[3]
- ParaCrawl (6,870,183 sentence pairs) (Bañón et al., 2020)
- DOGC v2 (10,933,622 sentence pairs) (Tiedemann, 2012).

[1] http://www.statmt. org/wmt21/similar.html
[2] https://cordis.europa.eu/project/id/844951
[3] http://data.statmt.org/wikititles/README

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 292–298

For Portuguese–Spanish, these parallel corpora were used:

- Europarl v10 (1,801,845 sentence pairs) (Koehn, 2005)
- News Commentary v16 (48,259 sentence pairs) (Tiedemann, 2012)
- Wiki Titles v3 (649,833 sentence pairs) (Barrault et al., 2020)
- Tilde MODEL (13,464 sentence pairs) (Rozis & Skadiņš, 2017).
- JRC-Acquis (1,650,126 sentence pairs) (Steinberger et al., 2006; Tiedemann, 2012)[4]

The above length specifications were taken from the SLT 2021 website's corpus download page. We did not use any of the comparable corpora provided by the SLT task organizers which, among others, included about 65 million sentences of Spanish news crawl.[5] The reason is that in the SEBAMAT project we achieved fairly good translation results when training with the Europarl corpus only. For example, we obtained BLEU scores (Papineni et al., 2002) well above 40 for Spanish–English and Greek–English when evaluated with randomly held out data. The Europarl corpus comprises in the order of 2 million sentences per language pair for many languages. As the parallel corpora for the shared task were much larger than this (with the Portuguese–Spanish language pair even including the respective language parts of the Europarl corpus), we saw no need to extract additional parallel sentences from comparable corpora. Such sentences are usually much noisier than parallel sentences based on human translations and could therefore possibly even reduce the quality of the NMT training in this high resource scenario.

Given the good quality of the training data provided by the shared task organizers, we only had to convert some of the files from a two-column translation memory format to the standard Moses format, and then concatenate all files of the source language as well as all files of the target language to form a large parallel training set. The concatenation was done in the order as listed above. However, as Marian NMT by default randomly shuffles the sentence pairs for training, the order of concatenation should not be of importance in our scenario.

2.2 Hardware

Marian NMT supports training using CPUs or GPUs. According to our experiments in the SEBAMAT project, training times in NMT can typically be reduced by about two orders of magnitude by conducting the training on a current GPU rather than on a (single) CPU. We therefore used a PC with an nVidia RTX 3090 GPU, supported by an i9 CPU. With 24 GB of memory, 28.3 billion transistors and 35.58 TFLOPS FP32 (float) performance, this GPU is state of the art in 2021, so – depending on parameter settings – with a single GPU we typically had training times of only a few hours per language pair. As our operating system we used Ubuntu 20.04 LTS.

As a side note, let us mention that performance in CPU-based training can be increased by using several CPU cores in parallel, which is supported by Marian NMT. With the 16 cores of the i9 processor, this looks promising if an appropriate GPU is not at hand. However, according to our experiments, each of the processors requires the full amount of memory. Therefore, if we assume 8 GB of memory per CPU core (which is typically the minimum for serious NMT work), we would require a total of 128 GB of RAM if we wished to use all 16 cores.

2.3 Software

As the translation engine we used the Marian NMT toolkit as it is well established and, for the reason that it is implemented in the C++ programming language, runs very fast (Kim et al., 2019), thereby substantially reducing training times. This is a particularly important consideration in a shared task where time tends to be very limited.

Marian NMT was installed on the above PC together with the nVidia driver and CUDA software. Our pre-processing pipeline involves the following steps: tokenization, cleaning, i.e. removal of very long sentences and sentence pairs with very different lengths, true-casing, and byte-pair encoding. For the first three steps we used the Moses tools *tokenizer*, *clean-corpus-n*, and *truecase* (Koehn et al., 2007).[6] For byte-pair encoding we used Rico Sennrich's Python program *bpe* (Sennrich et al., 2016). For post-processing of the translations, the tokenization and true-casing was reversed using the Moses tools *detruecase* and *detokenizer*.

[4] https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis

[5] http://data.statmt.org/news-crawl/README

[6] http://statmt.org/moses/

```
# train model
if [ ! -e "model/model.npz.best-translation.npz" ]
then
  $MARIAN_TRAIN \
  --devices $GPUS --sync-sgd --seed 1111 \
  --model model/model.npz --type transformer \
  --train-sets data/corpus.bpe.pt data/corpus.bpe.es \
  --max-length 100 \
  --vocabs model/vocab.ptes.yml model/vocab.ptes.yml \
  --mini-batch-fit -w 10000 --maxi-batch 1000 \
  --early-stopping 10 --cost-type=ce-mean-words \
  --valid-freq 5000 --save-freq 5000 --disp-freq 500 \
  --valid-metrics translation ce-mean-words perplexity cross-entropy \
  --valid-sets data/corpus-dev.bpe.pt data/corpus-dev.bpe.es \
  --valid-script-path "bash ./scripts/validate.sh" \
  --valid-translation-output data/valid.bpe.es.output --quiet-translation \
  --valid-mini-batch 64 \
  --beam-size 6 --normalize 0.6 \
  --log model/train.log --valid-log model/valid.log \
  --enc-depth 6 --dec-depth 6 \
  --transformer-heads 8 \
  --transformer-postprocess-emb d \
  --transformer-postprocess dan \
  --transformer-dropout 0.1 --label-smoothing 0.1 \
  --learn-rate 0.0003 --lr-warmup 16000 \
  --lr-decay-inv-sqrt 16000 --lr-report \
  --optimizer-params 0.9 0.98 1e-09 --clip-norm 5 \
  --tied-embeddings-all \
  --exponential-smoothing \
  --overwrite --keep-best
fi
```

Table 1: Parameters for Marian NMT training (Portuguese → Spanish).

The Moses tokenizer works well for Spanish and Portuguese, but has some problems with Catalan as there are some peculiarities in this language, most notably the interpunct (as used e.g. in the word *cel·la*). An insightful discussion on this can be found on GitHub.[7] As we did not have time to adapt the tokenizer to Catalan, to account for a few obvious errors, we did some minimalistic automatic post-processing (replacing a few short character sequences) as described in Section 4.

3 Experiments

To obtain any information on the training data and on the development and test sets, it was required to register for the shared task. We did so on July 13, 2021, so had seven days until July 19 when the submission of the results was due. This did not leave us much time for parameter optimization which is why we mostly took the standard parameters as suggested in the Marian NMT documentation for the Transformer architecture (Vaswani et. al, 2017).

We only did a few test runs with various settings concerning the number of merge operations in byte pair encoding (later to be referred to as *vocabulary size*), but did not have time to systematically optimize it. In our primary submissions, this parameter is set to 40,000, whereas in the comparative submissions, as in some previous SEBAMAT work, it is set to 85,000. For all language pairs and data sets (development and test), the smaller size performed better in terms of BLEU scores, although the exact size appears to be not very critical within a wide range of values.

To provide details on the core part of our experiments, in Table 1 we show the script for the Marian NMT training. As the parameters are well described in the Marian NMT documentation, we do not discuss them here. Let us only mention that during training BLEU scores are computed periodically on the development set, and that the training stops if the best score cannot be improved within ten iterations.

[7] https://github.com/alvations/sacremoses/issues/43

4 Results

We pre-processed the corpora as described in section 2 and trained the system for the language pairs Catalan→Spanish, Spanish→Catalan, Portuguese →Spanish and Spanish→Portuguese. We then inspected the translation results. For all language pairs the translations looked ok except for Spanish→Catalan. For this language pair we noticed, apparently because the tokenizer was not well suited for Catalan, the following three types of systematic errors in the output:

- There were extra spaces within Catalan words such as *paral·lel* because the *interpunct* (*punt volat*) was incorrectly interpreted as a separator between words rather than between syllables, which is why during tokenization blanks were inserted around it. We corrected this by replacing all «␣·␣» sequences («␣» stands for blank) in the translation output by «·».
- We found extra spaces before the apostrophe in phrases such as «l'efectiu» or after the apostrophe in phrases such as «d'informes». We therefore removed all spaces before and after apostrophes in the translation output.
- We noticed that whereas in the translated output «'» was used as the apostrophe, the sample translations in the development set used «´» instead. The reason is probably a discrepancy between training corpora and development sets. Assuming that the test set would have the same characteristics as the development set, we replaced in the translation output all occurrences of the former by the latter.

Vocabulary size	Language pair	BLEU score
40,000 (primary submission)	ca–es	80.72
	es–ca	83.32
	pt–es	50.37
	es–pt	44.96
85,000 (contrastive submission)	ca–es	79.40
	es–ca	81.21
	pt–es	47.29
	es–pt	42.77

Table 2: Results for the development sets. ca = Catalan, es = Spanish, pt = Portuguese. Without the interpunct and apostrophe substitutions, the BLEU score of es–ca (primary) is 69.40 and the BLEU score of es–ca (contrastive) is 68.01.

Vocabulary size	Language pair	BLEU	RIBES	TER	Rank
40,000 (primary)	ca–es	78.65	94.76	15.805	2
	es–ca	79.69	95.76	14.632	1
	pt–es	46.51	86.31	41.235	2
	es–pt	40.35	84.99	45.258	2
85,000 (contrastive)	ca–es	76.78	94.46	17.067	5
	es–ca	77.32	95.35	16.744	3
	pt–es	43.12	84.99	45.068	4
	es–pt	38.90	83.89	47.044	3

Table 3: Shared task results for the test sets as computed by the shared task organizers.

Especially the substitution of the apostrophes resulted in an improvement of several BLEU points, whereas the effects of blank removal before and after apostrophes differed depending on the software used for automatic evaluation. When using *Tilde's interactive BLEU score evaluator*[8] this change had no effect, whereas with the Moses *multi-bleu-detok.perl* tool, which we used in our scripts, a small improvement was obtained. The discrepancy can be explained by assuming that tools for computing BLEU scores often introduce some forms of tokenization or de-tokenization by themselves, and that these operations can slightly differ between tools.

Table 2 shows the BLEU scores obtained with the *multi-bleu-detok.perl* tool on the development sets for the four language pairs and the two parameter settings (byte pair encoding vocabulary sizes of 40,000 vs. 85,000). Table 3 shows the official BLEU scores for the test sets as computed by the shared task organizers who also provided scores for the RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006) measures on the evaluation section of the SLT webpage. These scores we cite in Table 3. The last column shows our submissions' ranks among the other teams participating in the competition. As can be seen, our primary submissions (byte pair encoding vocabulary size of 40,000) won the competition for Spanish → Catalan, and ranked second for the other three language pairs.

As can be expected from the evaluation scores, the translation quality is to the most part very good. This is particularly true for the language pairs involving Catalan. Table 4 shows a translation example for Portuguese → Spanish which is harder than Catalan → Spanish. For all language pairs, we

[8] https://www.letsmt.eu/Bleu.aspx

found a few occurrences of hallucinations (Raunak et al., 2021), mainly of the type where the beginning of a long sentence is translated well but towards the end a phrase translation is repeated over and over again. To give an idea how often this happened: manual inspection of the results for Catalan → Spanish (vocabulary size 85,000) showed that such hallucinations occurred in 7 of the 970 sentence translations of the development set.

FIRST THREE SEGMENTS OF TEST SET (PORTUGUESE)	SEBAMAT TRANSLATION (SPANISH)
O plano pretende contribuir para a realização dos objetivos da política comum das pescas e, em especial, para garantir que a exploração dos recursos biológicos marinhos vivos restabeleça e mantenha as populações das espécies exploradas acima de níveis que possam produzir o rendimento máximo sustentável (MSY).	El plan pretende contribuir a la consecución de los objetivos de la política pesquera común y, en particular, garantizar que la explotación de los recursos biológicos marinos vivos restablezca y mantenga a las poblaciones de especies explotadas por encima de niveles que puedan producir el rendimiento máximo sostenible (RMS).
Coordenar, em consulta com a Comissão, os métodos de diagnóstico da doença de Newcastle nos Estados-Membros, nomeadamente, mediante:	Coordinar, en consulta con la Comisión, los métodos de diagnóstico de la enfermedad de Newcastle en los Estados miembros, en particular mediante:
Por conseguinte, atualmente é preciso cofinanciamento da União para assegurar que a Finance Watch e a Better Finance recebam os recursos necessários para atingirem os objetivos pretendidos nos próximos anos, e para proporcionar estabilidade financeira a ambas as organizações, aos seus peritos e ao seu pessoal administrativo, que até agora conseguiram lançar as suas atividades relevantes em pouco tempo.	Por lo tanto, hoy es necesario cofinanciar la Unión para garantizar que Finance Watch y Better Finance reciban los recursos necesarios para alcanzar los objetivos deseados en los próximos años y para proporcionar estabilidad financiera a ambas organizaciones, a sus expertos y a su personal administrativo, que hasta ahora han logrado lanzar sus actividades relevantes en poco tiempo.

Table 4: First three Portuguese segments of the SLT 2021 test set and their translations to Spanish as produced by the primary SEBAMAT NMT system.

The hallucinations could be detected by looking at the ratio of sentence lengths between a source language sentence and its translation, and/or by detecting repetitive phrases towards the end of a sentence translation. However, according to Raunak et al. (2021), hallucinations are a problem of training data quality, which to improve would have been too time-consuming. We thought of greedy solutions such as cutting off repetitive sentence ends, but did not implement them for lack of time and as they would be hard to justify.

5 Discussion and conclusions

Given the observation that the language pairs involving Catalan achieved considerably higher evaluation scores than those involving Portuguese, the question arises how this can be explained. Our somewhat speculative answer is as follows: As Catalan's grammar and sentence structure is very similar to Spanish, with differences mainly on the vocabulary side, extremely high scores can be achieved because in many cases, often in the form of a word-by-word translation, there is just one obvious way how to translate a sentence for both man and machine. This is only to a lesser extend true for Spanish–Portuguese, so the lower scores are likely caused by more variability in acceptable translation options, rather than by lower translation quality.

When comparing the BLEU scores in Tables 2 and 3, it can be seen that our system performed significantly worse on the test sets than it did on the development sets. From this we conclude that probably the test data is less representative of the training data than the development data. Problems with overfitting seem unlikely as in the previous SEBAMAT work we had usually used randomly held out sentences of the training data for both development and testing. In such a scenario, the results were very similar in both cases, with only minor unsystematic discrepancies in BLEU scores.

Finally, let us try to answer the questions raised in the introduction. As it was ranked first for Spanish → Catalan and second for the other three language pairs, it appears that especially our primary systems (with vocabulary size 40,000) are competitive. Let us mention, however, that all participating systems showed similarly convincing evaluation scores, so that minor differences in parameter choice (such as vocabulary size) or in the organizers' selection of the test set may have had a noticeable impact on the rankings.

Like many other studies, this work provides once again evidence how powerful NMT is, and how well the Marian toolkit works: Within a week it was possible for a single developer to add two new language pairs (in two directions each) to the SEBAMAT portfolio, despite mediocre proficiency of Spanish and hardly any proficiency of Catalan and Portuguese. Of course, achieving good translation quality was considerably facilitated by the similarity of the languages and by the good quality and size of the training data provided by the shared task organizers.

Acknowledgments

This work was funded by the Marie Curie Individual Fellowship SEBAMAT (grant agreement number 844951) within the European Commission's Horizon 2020 Framework Programme. I would like to thank my colleagues at Athena R.C., the SLT and the WMT organizers, and all researchers who generously made available the tools and resources used in this work.

References

Bañón, Marta; Chen, Pinzhen; Haddow, Barry; Heafield, Kenneth; Hoang, Hieu; Esplà-Gomis, Miquel; Forcada, Mikel L.; Kamran, Amir; Kirefu, Faheem; Koehn, Philipp; Ortiz Rojas, Sergio; Pla Sempere, Leopoldo; Ramírez-Sánchez, Gema; Sarrías, Elsa; Strelec, Marek; Thompson, Brian; Waites, William; Wiggins, Dion; Zaragoza, Jaume (2020). ParaCrawl: Web-scale acquisition of parallel corpora. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 4555–4567.

Barrault, Loïc; Biesialska, Magdalena; Bojar, Ondřej; Costa-jussà, Marta R.; Federmann, Christian; Graham, Yvette; Grundkiewicz, Roman; Haddow, Barry; Huck, Matthias; Joanis, Eric; Kocmi, Tom; Koehn, Philipp; Lo, Chi-kiu; Ljubešić, Nikola; Monz, Christof; Morishita, Makoto; Nagata, Masaaki; Nakazawa, Toshiaki; Pal, Santanu; Post, Matt; Zampieri, Marcos (2020). Findings of the 2020 Conference on Machine Translation (WMT20). *Proceedings of the Fifth Conference on Machine Translation*, 1–55.

Isozaki, Hideki; Hirao, Tsutomu; Duh, Kevin, Sudoh, Katsuhito; Tsukada, Hajime (2010). Automatic evaluation of translation quality for distant language pairs. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 944–952.

Junczys-Dowmunt, Marcin; Grundkiewicz, Roman; Dwojak, Tomasz; Hoang, Hieu; Heafield, Kenneth; Neckermann, Tom; Seide, Frank; Germann, Ulrich; Aji, Alham; Bogoychev, Nikolay; Martins, André; Birch, Alexandra (2018). Marian: Fast Neural Machine Translation in C++. *Proceedings of ACL 2018, System Demonstrations*, 116–121.

Kim, Young Jin; Junczys-Dowmunt, Marcin; Hassan, Hany; Fikri Aji, Alham; Heafield, Kenneth; Grundkiewicz, Roman; Bogoychev, Nikolay (2019). From research to production and back: ludicrously fast neural machine translation. *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 280–288.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Mo-ran, Richard Zens, Chris Dyer, Ondrej Bojar, Alex-andra Constantin, Evan Herbst. (2007). Moses: Open Source Toolkit for Statistical Machine Trans-lation. Annual Meeting of the Association for Compu¬tational Linguistics (ACL), demonstration session, Prague, 177–180.

Koehn, Philipp (2005). Europarl: a parallel corpus for statistical machine translation. *Proceedings of the Tenth Machine Translation Summit,* Phuket, Thailand, 79–86.

Papineni, Kishore; Roukos, Salim; Ward, Todd; Zhu, Wei-Jing (2002). BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 311–318.

Rapp, Reinhard; Tambouratzis, George (2020). An overview of the SEBAMAT project. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, 491–492.

Raunak, Vikas; Arul Menezes, Marcin Junczys-Dowmunt (2021). The curious case of hallucinations in neural machine translation. *NAACL-HLT*, 1172–1183.

Rozis, Roberts; Skadiņš, Raivis (2017). Tilde MODEL – multilingual open data for EU languages. *Proceedings of the 21st Nordic Conference of Computational Linguistics (NODALIDA)*, Gothenburg, Sweden, 263–265.

Sennrich, Rico; Barry Haddow, Alexandra Birch (2016). Neural machine translation of rare words with subword units. *Proc. of the 54th Annual Meeting of the ACL (Vol. 1: Long Papers)*, 1715–1725.

Snover, Matthew; Dorr, Bonnie; Schwartz, Rich; Micciulla, Linnea; Makhoul, John (2006). A study of translation edit rate with targeted human annotation. Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, 223–231.

Steinberger, Ralf; Pouliquen, Bruno; Widiger, Anna; Ignat, Camelia; Erjavec, Tomaž, Tufis, Dan; Varga, Dániel (2006). The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006).* Genoa, Italy, 24–26.

Tiedemann, Jörg (2012). Parallel data, tools and interfaces in OPUS. *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Istanbul, 2214–2218.

Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jacob; Jones, Llion; Gomze, Aidan N. G.; Kaiser, Lukasz; Polosukhin, Illia (2017). Attention is all you need. *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 6000–6010.

NITK-UoH: Tamil-Telugu Machine Translation Systems for the WMT21 Similar Language Translation Task

Richard Saldanha,
Ananthanarayana V. S and **Anand Kumar M**
Department of Information Technology,
National Institute of Technology Karnataka
NH 66, Srinivasnagar, Surathkal, Mangalore
Karnataka 575025, India
richardsaldanha.207it005@nitk.edu.in
anvs@nitk.edu.in
m_anandkumar@nitk.edu.in

Parameswari Krishnamurthy
Centre for Applied Linguistics
and Translation Studies,
University of Hyderabad
Prof. CR Rao Road
Gachibowli, Hyderabad
Telangana 500046, India
pksh@uohyd.ac.in

Abstract

In this work, two Neural Machine Translation (NMT) systems have been developed and evaluated as part of the bidirectional Tamil-Telugu similar languages translation subtask in WMT21. The OpenNMT-py toolkit has been used to create quick prototypes of the systems, following which models have been trained on the training datasets containing the parallel corpus and finally the models have been evaluated on the dev datasets provided as part of the task. Both the systems have been trained on a DGX station with 4 - V100 GPUs.

The first NMT system in this work is a Transformer based 6 layer encoder-decoder model, trained for 100000 training steps, whose configuration is similar to the one provided by OpenNMT-py and this is used to create a model for bidirectional translation. The second NMT system contains two unidirectional translation models with the same configuration as the first system, with the addition of utilizing Byte Pair Encoding (BPE) for subword tokenization through the pre-trained MultiBPEmb model. Based on the dev dataset evaluation metrics for both the systems, the first system i.e. the vanilla Transformer model has been submitted as the Primary system. Since there were no improvements in the metrics during training of the second system with BPE, it has been submitted as a contrastive system.

1 Introduction

Tamil is a language, predominantly spoken in Tamil Nadu, a state in Southern India, along with countries with a large Tamil speaking diaspora such as Sri Lanka, Malaysia and Singapore, to name a few. Telugu on the other hand is the official language of two Southern states in India, namely Andhra Pradesh and Telangana. It is also spoken among the Telugu speaking immigrant population in the USA, Canada and the UK. Both languages belong to the Dravidian family of languages which comprise of Tamil, Telugu, Kannada and Malayalam as the major languages spoken in South India. Despite belonging to the same family of languages, there are many differences between Tamil and Telugu, such as the script used for writing and linguistic differences in terms of phonology, morphology, syntax among others. Tamil belongs to the Southern branch of Dravidian languages, which has a rich literary tradition spanning more than 2000 years. Telugu, on the other hand, belongs to the South Central branch of Dravidian languages and has a considerable amount of different linguistic characteristics when compared to Tamil as described by Krishnamurthy (2019).

As part of the similar language translation's subtask for Dravidian Languages, namely Tamil (TA) and Telugu (TE), we have attempted to build Neural Machine Translation (NMT) models using the OpenNMT-py toolkit [1], which helps to generate quick prototypes for the NMT models with the desired configurations. The first NMT system (submitted as the primary system) in this work is a Transformer based 6 layer encoder-decoder model which provides a single model for bidirectional translation between Tamil and Telugu using the datasets provided for this shared task. The second NMT system (submitted as the contrastive system) consists of two unidirectional translation models with the same configuration as the first system, but with the addition of utilizing Byte Pair Encoding (BPE) for subword tokenization using the pre-trained MultiBPEmb model (Heinzerling and Strube, 2018).

The rest of the work is described in sections that pertain to the related work, data, system descrip-

[1] https://opennmt.net/OpenNMT-py/main.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 299–303
November 10–11, 2021.

Dataset Type	Dataset Name	Number of samples
Parallel Aligned TA-TE pairs (Training)	PM India	26009
Parallel Aligned TA-TE pairs (Training)	News	11038
Parallel Aligned TA-TE pairs (Training)	MKB	3100
Parallel Aligned TA-TE pairs (Dev)	Dev	1261
Non Aligned TA-TE sets (Test)	Test	1735 (per language set)

Table 1: Dataset statistics for parallel aligned Tamil-Telugu pairs used as train and dev (validation) datasets along with non aligned samples used as the test set.

Dataset Type	Dataset Name	Language	Longest Line Length
Training	PM India	TA	659
Training	News	TA	1524
Training	MKB	TA	412
Dev	Dev	TA	923
Test	Test	TA	1544
Training	PM India	TE	718
Training	News	TE	1356
Training	MKB	TE	376
Dev	Dev	TE	1004
Test	Test	TE	757

Table 2: Dataset statistics for Longest Line.

tion, results and conclusion.

2 Rationale for Selecting the Models and Related Work

There has been a significant amount of work done on developing machine translation systems for Indian languages, with some notable examples for Dravidian languages such as Tamil and Malayalam described in Kumar et al. (2019). This shared task provides a unique challenge in terms of the constraint on the parallel aligned language pair data made available for training. The other challenges include the linguistically rich and domain specific content present in the Prime Minister of India (PMI) and the Mann ki baat (MKB) datasets, where topics related to India's domestic and foreign policy issues can be found.

In order to address the challenge of lengthy input (samples containing more than 300 space delimited tokens), the Transformer model described by Vaswani et al. (2017) was adopted. This model provides the multi head attention mechanism which helps retain context for longer length sentence samples. To reduce the vocabulary, reduce the training time and possibly improve the translation quality (through sub word tokenization), a MultiBPEmb model trained with a vocabulary of 100000 tokens from 275 languages has been utilised (Heinzerling and Strube, 2018).

Other methods to improve translation quality, that have not been explored as part of this work are the use of back translation using monolingual corpus or corpora, on the lines of the one described by Sennrich et al. (2016). Factored NMT (which uses data tagged on the basis of morphology and Parts of Speech (POS)) such as the one described by García-Martínez et al. (2016) is another possible candidate suitable for the kind of challenge provided by the similar language translation task, as the use of POS and morphological information can reduce the number of tokens and make the models more generalizable in terms of predictions.

3 Data

The datasets used in the NMT systems for this work are the parallel aligned Tamil and Telugu (TA-TE) language pairs provided as part of the Dravidian Language sub task of the Similar Language Translation shared task[2]. Some statistics about the dataset are outlined in Table 1.

3.1 Dataset preprocessing

Due to the moderate size of the training dataset, which contains 40147 samples, along with the topic

[2] https://wmt21similar.cs.upc.edu/

Model Configuration Name	Model Configuration Value
Corpus Weights for PMI dataset	23
Corpus Weights for News dataset	19
Corpus Weights for MKB dataset	3
Source and Target Sequence Length	1600
Save checkpoint after steps	500
Number of training steps	100000
Number of validation steps	5000
Training batch size	4096
Dev(validation) batch size	16
Optimizer	Adam
Number of Encoder Decoder Layers	6 (each)
Number of Attention heads	8

Table 3: Training Configuration for Transformer based Encoder-Decoder Model (Primary System).

overlap of sentence samples between the training and dev datasets as well as test set (to a certain extent) on topics such as the Indian Prime Minister's statements on domestic issues and foreign policies in the PM India dataset, the entire training dataset has been utilized in its original form.

The length wise statistics of the dataset (in terms of space delimited tokens) is given in Table 2, this was taken as the deciding factor in fixing the maximum input length as 1600 for the NMT systems developed. The tokenization for the primary system was done as space delimited tokens which yielded a shared Tamil-Telugu vocabulary of 194860 tokens. On the other hand on using the MultiBPEmb model for subword tokenization gave a vocabulary of 14056 tokens for Tamil (TA) and 13170 tokens for Telugu (TE), which included some words in English as well.

4 System Description

As mentioned in section 1, the PyTorch based toolkit OpenNMT-py has been used to create rapid prototypes for NMT models (the motivations for the same can be seen in section 2), which have then been trained on the datasets provided, validated against the provided dev sets and finally translations for the test sets described in section 3 have been obtained and submitted to the committee for evaluating the Similar Language Translation task.

A DGX station with 4 - V100 GPUs have been used to train the models utilized in this task. A Transformer based 6 layer encoder-decoder model on the lines of the NMT system described by Vaswani et al. (2017), was trained for 100000 training steps as the first NMT system to be evaluated. The configuration for this model is the same as that provided by OpenNMT-py. In order to save time, a single bidirectional translation model for TA-TE language pair has been created, which can translate from Tamil to Telugu and vice versa. The datasets used in this system were doubled in terms of the number of samples when compared to the second NMT system (constrastive submission), by reversing the position of the TA-TE language pair and appending them to the original datasets. No special tagging identifiers were used as the Tamil and Telugu scripts are distinct.

Basic space delimited tokenization was applied on the datasets, which resulted in a combined TA-TE vocabulary of 194860 tokens being generated, the relevant key configuration for this model are listed in Table 3.

The corpus weights help assign varied importance to the particular datasets used in this task, the values for these weights were determined after visual analysis of the dev(validation) dataset which indicated the dev dataset's contents had a greater overlap with PMI, News and (Mann ki Baat - which roughly translates to "From the heart") MKB in that particular order. The training time for the entire model was 18 hours.

The second NMT system consists of two unidirectional translation models with the same configuration as the first system, with the addition of utilizing Byte Pair Encoding (BPE) for subwords using the pretrained MultiBPEmb model (Heinzerling and Strube, 2018). The intuition behind using BPE was to reduce the vocabulary size using subword tokenization. The choice of the pre trained BPE model was based on the relevance of content

System Name	Source Language	Target Language	BLEU	RIBES	TER
Primary System (Transformer Based)	TA	TE	4.321	7.4	99.1
Contrastive System (Transformer Based + BPE subword)	TA	TE	0.003	0.0	130.6
Primary System (Transformer Based)	TE	TA	3.908	9.0	98.7
Contrastive System (Transformer Based + BPE subword)	TE	TA	0.029	3.0	105.0

Table 4: Dev dataset BLEU, RIBES and TER Corpus level scores using the VizSeq library.

System Name	Source Language	Target Language	BLEU	RIBES	TER	System Rank
Primary System	TA	TE	6.09	17.03	-	1
Contrastive System	TA	TE	0.00	0.03	-	9
Primary System	TE	TA	6.55	19.61	98.356	4
Contrastive System	TE	TA	0.04	1.00	-	9

Table 5: Test dataset BLEU, RIBES, TER scores and BLEU based System Rank in the Shared Task

used for BPE model training, languages supported and size of the vocabulary. Heinzerling and Strube (2018) describes a MultiBPE model with a 100000 vocabulary which was deemed suitable for this task as it supported Tamil and Telugu, was trained on WikiNews and could use a single vocabulary like the first NMT system used in this work. During training it was found that the translations for the Dev set couldn't distinguish between Tamil and Telugu subwords correctly, due to the failure in vocabulary matching for the candidates used in the evaluation and possibly due to the vocabulary shared between the languages. Hence, this system was trained twice generating two unidirectional models for TA-TE and TE-TA translations. The training time for each model was 5 hours, which is less when compared to the primary system due to the number of samples used (the primary system uses double the number of samples) and the vocabulary size (the contrastive system has a smaller and fixed vocabulary as a pre trained BPE model has been used).

5 Results

The evaluation metrics used to evaluate the systems in this task are BiLingual Evaluation Understudy (BLEU) score as described by Papineni et al. (2002), Rank-based Intuitive Bilingual Evaluation (RIBES) score as described by Isozaki et al. (2010) and Translation Error Rate (TER) as described by Snover et al. (2006).

Corpus level metrics for the dev dataset were computed using the VizSeq python library which is an implementation of several metrics described by Wang et al. (2019).The metrics for the dev dataset are listed in Table 4.

Based on the evaluation metrics of the Dev (validation) dataset translations for both the systems evaluated in this work, the first system i.e. the vanilla Transformer model has been submitted as the Primary system. Since there were no improvements in the metrics (the reason for it can be seen in section 6), during training of the second system which consists of the Transformer model along with the use of MultiBPEmb model for sub word tokenization, hence the second system has been submitted as a contrastive system.

Table 5 lists the evaluation metrics[3] applied on the test dataset and the BLEU based system rank in the shared task provided by the evaluation committee [4,5].

6 Conclusion and Future Work

The analysis of the evaluation metrics, from section 5, on the dev dataset indicates that the primary system, which is a Transformer based Encoder-

[3]The results of the TER metrics for the test set translations have been marked as - (refer Table 5), when the values exceed 100.0

[4]https://mzampieri.com/workshops/wmt/2021/TA_TE.pdf

[5]https://mzampieri.com/workshops/wmt/2021/TE_TA.pdf

Decoder model, performs better than the contrastive system which contains Transformer based NMT models with BPE for subword tokenization. The reason for this is possibly due to the lack of vocabulary matching the candidates being evaluated and also due to the shared vocabulary of the MultiBPEmb model. The choice of a pre trained MultiBPE model was to reduce effort on the embeddings, but in hindsight training the MultiBPE model using the given datasets or fine tuning the pre trained MultiBPE model on the given datasets would have been a better choice.

As seen from the evaluation of translations obtained using the Dev and Test datasets using BLEU, RIBES and TER metrics in section 5, there is a considerable scope of improvement in the scenario where a constraint is placed on the number of datasets containing parallel corpus language pair samples, that can be used for training. The possible reason for the low BLEU scores in the primary system is the relatively small number of samples used along with the presence of a large variety in the linguistic forms present in the datasets. In the case of the contrastive system, the low BLEU scores can be attributed to the use of the pre trained MultiBPE model (a pre trained BPE model fine tuned on the given datasets would have helped improved the scores). Some approaches that have the potential to improve the results are, the use of back translation using monolingual corpus (through training corpus augmentation and providing more training examples for the model to learn), utilizing domain specific corpora from the shared machine translation task for Indian Languages described in section 2. Factored NMT, an NMT which uses input tagged on the basis of morphology and Parts of Speech (POS) to reduce the number of tokens, the use of alternative BPE models trained on content which are a close match to the dataset used in the shared task, are other promising alternatives.

References

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation architectures. In *International Workshop on Spoken Language Translation (IWSLT'16)*.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2989–2993, Miyazaki, Japan. European Language Resources Association (ELRA).

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952.

Parameswari Krishnamurthy. 2019. Development of telugu-tamil transfer-based machine translation system: An improvization using divergence index. *Journal of Intelligent Systems*, 28(3):493–504.

M. Anand Kumar, B. Premjith, Shivkaran Singh, S. Rajendran, and K. P. Soman. 2019. An overview of the shared task on machine translation in indian languages (mtil) – 2017. *Journal of Intelligent Systems*, 28(3):455–464.

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.

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.

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: Technical Papers*, pages 223–231.

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.

Changhan Wang, Anirudh Jain, Danlu Chen, and Jiatao Gu. 2019. Vizseq: A visual analysis toolkit for text generation tasks. `https://arxiv.org/pdf/1909.05424.pdf`. (Accessed on 08/04/2021).

A3-108 Machine Translation System for Similar Language Translation Shared Task 2021

Saumitra Yadav, Manish Shrivastava
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
saumitra.yadav@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In this paper, we describe our submissions for the Similar Language Translation Shared Task 2021. We built 3 systems in each direction for the Tamil $\Longleftrightarrow$ Telugu language pair. This paper outlines experiments with various tokenization schemes to train statistical models. We also report the configuration of the submitted systems and results produced by them.

1 Introduction

Machine translation is a process of translating text from a source to a target language. There are multiple ways of building such a system - Rule-based, Data-driven, Hybrid etc. In this shared task, we use data-driven method to create machine translation system for Tamil $\Longleftrightarrow$ Telugu. Due to low-resource setting of this language pair in the shared task, we use Statistical Machine translation method (Koehn et al., 2003),(Koehn and Knowles, 2017) to build systems.

Tamil Telugu language pair comes under the bracket of similar languages. Similar languages show similarity in their lexical and syntactical properties (Kunchukuttan et al., 2014a). This may be due to them being in close proximity of each other for long time. This can also be due to common ancestry. In the current digital context, translation between similar languages is of importance. But there can be scarcity of good quality parallel text. In the current shared task, we have a language pair which is morphologically rich and with $\simeq$39K parallel sentences. So, following Kunchukuttan and Bhattacharyya (2017) and Kunchukuttan et al. (2014b) we use sentencepiece[1] (Kudo and Richardson, 2018) and morfessor[2] (Virpioja et al., 2013) to segment tokens in the dataset into subwords. And due to the size of parallel text ($\simeq$39K parallel text) coming under purview of low resource, we make use of Moses[3](Koehn et al., 2007) to create statistical machine translation models(Koehn and Knowles, 2017).

For this shared task we developed 3 translation systems (1 Primary and 2 Contrastive) in each direction Tamil $\Longleftrightarrow$ Telugu. For each output we post-processed and detokenized translation output depending on the tokenization scheme for target language. To choose a primary and 2 contrastive systems, we compared BLEU (Papineni et al., 2002) scores on output of development dataset for each system using sacrebleu[4] (Post, 2018). The Following sections give more details about the systems developed.

2 SMT systems using different schemes

We used various tokenization schemes to build translation systems. Evaluated these systems on the development dataset. After post-processing, detokenizing and scoring each translation output, we submit output systems as primary and contrastive submissions accordingly.

2.1 Data and preprocessing

We used parallel data provided by the organizers to train all the models. IndicNLP[5] (Kunchukuttan, 2020) was used to normalize and tokenize datasets. 2 Subword models were trained on tokenized text for each language. Sentencepiece(Kudo and Richardson, 2018) was used to prepare a subword tokenizer model with vocabulary size set to 32000 and character coverage set to 0.9995. Another alternative tokenization model was trained on morfessor(Virpioja et al., 2013). To create 3 systems for each translation direction, we used the

[1] https://github.com/google/sentencepiece
[2] https://github.com/aalto-speech/morfessor
[3] https://github.com/moses-smt/mosesdecoder
[4] https://github.com/mjpost/sacrebleu
[5] https://github.com/anoopkunchukuttan/indic_nlp_library

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 304–306

Dataset with tokenization	Tamil			Telugu			Total number of Lines
	Total Token Count	Total Unique Token	Avg Token Per line	Total Token Count	Total Unique Token	Avg Token Per line	
Train.basicTok	691433	74341	17.22	725365	72949	18.06	39836
Dev.basicTok	30017	9683	23.80	30359	9467	24.07	1261
Train.spm	770632	31674	19.63	956023	31782	24.35	39246
Dev.spm	36672	8647	29.08	41779	9112	33.13	1261
Train.morf	956485	13956	24.47	947463	17823	24.24	39081
Dev.morf	45279	5496	35.90	43602	6380	34.57	1261

Table 1: Statistics of Tamil and Telugu datasets

following tokenization schemes,

- basicTok: bitext is tokenized with IndicNLP.
- morf: each training file in the parallel text is tokenized into subwords with the respective morfessor model.
- spm: each training file in the parallel text is tokenized into subwords with the respective sentencepiece model

Table 1 shows the statistics of the Tamil and Telugu dataset for each tokenization scheme after using `clean-corpus-n.perl` script with 1,70 as min,max line length for training text. No additional monolingual dataset was used in building any of the models.

2.2 MT Systems

We build a trigram language model with kneser ney smoothing for each language in each tokenization scheme using KenLM (Heafield, 2011). And used Moses (Koehn et al., 2007) to train an SMT system. MERT (Och, 2003) is used for tuning the trained model on development datasets. The performance of all systems, for each language direction on respective tokenized development datasets, is given in Table 2. For this shared task, we submit 3 systems (1 PRIMARY and 2 CONTRASTIVE) for each language direction for evaluation. Depending on scores on development dataset, systems build were submitted as,

	Tamil ->Telugu	Telugu ->Tamil
basicTok	7.7	9.9
spm	5.2	9.0
morf	7.7	9.8

Table 2: BLEU score on development dataset for each system

- For Telugu to Tamil,
  - A3-108_TE_TA_PRIMARY.txt: basicTok Telugu -> basicTok Tamil system - trained using SMT model - tokenized using indic nlp library.
  - A3-108_TE_TA_CONTRASTIVE1.txt: morf Telugu -> morf Tamil system - trained using SMT model - tokenized using morfessor into subwords for training
  - A3-108_TE_TA_CONTRASTIVE2.txt: spm Telugu -> spm Tamil system - trained using SMT model - tokenized using sentencepiece into subwords for training
- For Tamil to Telugu,
  - A3-108_TA_TE_PRIMARY.txt: morf Tamil -> morf Telugu system - trained using SMT model - tokenized using morfessor into subwords for training
  - A3-108_TA_TE_CONTRASTIVE1.txt: basicTok Tamil -> basicTok Telugu system - trained using SMT model - tokenized using indic nlp library.
  - A3-108_TA_TE_CONTRASTIVE2.txt: spm Tamil -> spm Telugu system - trained using SMT model - tokenized using sentencepiece into subwords for training

2.3 Results

This subsection compares the results of our systems, which we received from organizers, in terms of BLEU scores. Table 3 shows the BLEU scores for Telugu to Tamil systems. In comparison with other systems, all of our system outputs score highest. We were hoping that, in test cases, models using subwords for training and translating would prove to be better than basicTok, but that was not the case. Instead models trained on basicTok fared better.

System Type	BLEU	RIBES	TER
PRIMARY (basicTok)	8.37	43.55	95.884
CONTRASTIVE1 (morf)	7.89	46.24	95.627
CONTRASTIVE2 (spm)	7.43	42.54	94.964

Table 3: Scores on test dataset for each Telugu to Tamil system

Table 4 shows the BLEU score we received for Tamil to Telugu systems. Our system outputs from

System Type	BLEU	RIBES	TER
CONTRASTIVE1 (basicTok)	5.54	40.58	98.082
PRIMARY (morf)	5.23	42.37	98.662
CONTRASTIVE2 (spm)	3.32	34.42	-

Table 4: Scores on test dataset for each Tamil to Telugu system

CONTRASITVE1 and PRIMARY submission are in the top 3 in comparison with other systems. Here again, we see basicTok model fared a bit better than model trained on morf segmented dataset. And sentencepiece model was $\simeq$2 BLEU points behind both the systems. These BLEU scores (CONTRASTIVE1, PRIMARY) are in the top 3. Again, we were hoping, that in test cases, models using subwords for training and translating would prove to be better. But as was case in Telugu to Tamil, here also models trained on basicTok dataset fared better, followed by models trained on morfessor segmented dataset.

References

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. 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.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2017. Learning variable length units for SMT between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 14–24, Copenhagen, Denmark. Association for Computational Linguistics.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014a. Shata-anuvadak: Tackling multiway translation of Indian languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1781–1787, Reykjavik, Iceland. European Language Resources Association (ELRA).

Anoop Kunchukuttan, Ratish Pudupully, Rajen Chatterjee, Abhijit Mishra, and Pushpak Bhattacharyya. 2014b. The iit bombay smt system for icon 2014 tools contest. *NLP Tools Contest at ICON 2014*.

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*, pages 160–167, Sapporo, Japan. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Netmarble AI Center's WMT21 Automatic Post-Editing Shared Task Submission

Shinhyeok Oh*, **Sion Jang***, **Hu Xu***, **Shounan An**, **Insoo Oh**
Netmarble AI Center

Abstract

This paper describes Netmarble's submission to WMT21 Automatic Post-Editing (APE) Shared Task for the English-German language pair. First, we propose a Curriculum Training Strategy in training stages. Facebook Fair's WMT19 news translation model was chosen to engage the large and powerful pre-trained neural networks. Then, we post-train the translation model with different levels of data at each training stages. As the training stages go on, we make the system learn to solve multiple tasks by adding extra information at different training stages gradually. We also show a way to utilize the additional data in large volume for APE tasks. For further improvement, we apply Multi-Task Learning Strategy with the Dynamic Weight Average during the fine-tuning stage. To fine-tune the APE corpus with limited data, we add some related subtasks to learn a unified representation. Finally, for better performance, we leverage external translations as augmented machine translation (MT) during the post-training and fine-tuning. As experimental results show, our APE system significantly improves the translations of provided MT results by -2.848 and +3.74 on the development dataset in terms of TER and BLEU, respectively. It also demonstrates its effectiveness on the test dataset with higher quality than the development dataset.

1 Introduction

Automatic Post-Editing (APE) aims to improve the quality of an existing Machine Translation (MT) system by learning from human-edited samples (Chatterjee et al., 2019, 2020). With the continuous performance improvements of Neural Machine Translation (NMT) systems along with deep learning advancements, developing APE systems has faced a big challenge. Simple translation errors are hard to find in machine translation outputs, and the remaining errors are still hard to solve. In recent years, transfer learning and data augmentation techniques have shown their efficiency when training models on datasets with limited size (Devlin et al., 2019). Therefore, such approaches are also adopted in APE tasks (Lopes et al., 2019).

Participants in WMT21 APE shared tasks are required to develop systems to automatically post-edit the translation outputs from an unknown MT system. In this year, the same data has been re-post-edited to improve the quality. As a result of performing statistics on the development set, the evaluation scores are 19.057 and 68.79 in terms of TER and BLEU, which are much higher than the scores of last year, 31.374 and 50.37, respectively. The central distribution of TER has shifted to the left compared to last year. We find that the section in the range of 5 to 10 has the most examples, which indicates that over-correction problems should be considered during the APE tasks. In addition, the dataset has been changed in terms of the domain (from IT to Wikipedia), which results in the change in data distribution. Therefore, directly using previous datasets or officially provided synthetic corpus (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018) to enlarge the training set of APE tasks might not be appropriate under such circumstances. In work by Yang et al. (2020), considering the change of data distribution, they select to use additional MT candidates as the data augmentation method to improve feature diversity in their APE systems, which significantly improves the APE performance.

Inspired by this idea, we decided to solve the APE task as NMT alike task and utilize the external MT at the fine-tuning stage. However, because of the limited size of the APE corpus and the improvement of MT quality, fine-tune the model only on the APE data, easily reach the performance ceiling in spite of using external translation. To solve the aforementioned issues, existing works for other

*These authors equally contributed to this work.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 307–314
November 10–11, 2021.

Natural Language Processing (NLP) tasks have adopted several Multi-task Learning (MTL) methods with the auxiliary task (Whang et al., 2021; Oh et al., 2021). We wondered whether it is possible to apply MTL mechanism with APE task to the fine-tuning stage since MTL trains the model to encourage representation sharing and improve generalization performance. Furthermore it aims to alleviate the data sparsity problem with a limited number of data in each task (Zhang and Yang, 2021). Therefore, we add some related NLP tasks along with the APE task. Our experiment results demonstrate that such approaches can further improve performance.

As mentioned above, large-volume data, such as news translation data and artificial synthetic data, can not be used to enlarge the APE corpus directly during the fine-tuning because of the large gap in data distribution. We wondered if there is a way to apply any learning method to the post-training so that we can utilize more data to train a more robust and powerful model. In work by (Xu et al., 2020), they applied Curriculum Learning according to the difficulty of each example on a single training stage. Inspired by the research, we try to apply Curriculum Learning across multiple training stages. As the training stage increases, we make the system learn to solve the different tasks by gradually providing extra information, described in Section 3 in detail. Extensive experiments show the effectiveness of applying the Curriculum Learning Strategy during the training phase. Finally, We combined these two approaches to make our final APE system, which significantly improves the performance of the APE task.

Our APE system is built based on Transformer (Vaswani et al., 2017) and is post-trained on WMT21 News-Translation Data (Koehn, 2005; Tiedemann, 2012; Rozis and Skadiņš, 2017; Bhatia et al., 2016; Tiedemann, 2012) and artificial synthetic data (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018) provided by APE Task with Curriculum Learning Strategy. For fine-tuning, MTL is applied with related NLP subtasks such as Part-Of-Speech (POS), Named Entity Recognition (NER), Masked Language Model (MLM), and Keep/Translate are added to the model to reduce the over-fitting as well as achieve better performance, described in Section 4 in detail. For better training efficiency, the Dynamic Weight Average (DWA) mechanism (Liu et al., 2019) is applied during the MTL to keep the correct balance between these subtasks. Here we summarize our contributions as follows:

- We design Multi-task Learning Strategy (MLS) with DWA to the fine-tuning stage, which improves the training efficiency and the performance significantly.
- We adapt Curriculum Training Strategy (CTS) to our APE system during the post-training across the multiple training stages, which shows the effectiveness in performance. In addition, we showed a way to utilize the additional data in large volumes in APE tasks.

2 Base System

Our system is based on Facebook FAIR's WMT19 News Translation Model (Ng et al., 2019), which used the big Transformer (Vaswani et al., 2017) and provided the pre-trained weights. We use both of them as our base system. In addition, we utilize data augmentation with external MT, which has been proposed by Yang et al. (2020) to generate the external translated sentence (mt_ext) and help generate the post-editing sentence (pe). An input sentence X that contains a source sentence (src), a translated sentence by the machine translation system (mt), and an external translated sentence (mt_ext) is defined as,

$$X = [src \texttt{<SEP>} mt \texttt{<SEP>} mt\_ext], \quad (1)$$

and output a sequence, $H = [h_{src_0}, h_{src_1}, ..., h_{src_n}, h_{\texttt{<SEP>}}, h_{mt_0}, ..., h_{mt_m}, h_{\texttt{<SEP>}}, h_{mt\_ext_0}, ..., h_{mt\_ext_l}] \in \mathbb{R}^{d_h \times (n+m+l+2)}$, where d_h represents a dimension of the encoder, and n, m, l represents the number of tokens for src, mt, mt_ext, respectively. We represent the parameters of the encoder as Θ_s. Then, H is fed into the decoder, and the decoder target is defined as $Y = [pe]$.

3 Curriculum Training Strategy (CTS)

CTS has been inspired by Curriculum Learning (Xu et al., 2020) that is applied according to the difficulty of each example on a single training stage, which has already been applied to our baseline architecture by Ng et al. (2019). In addition, we propose CTS, which applied Curriculum Learning across multiple training stages. CTS aims at step-by-step learning. In an early stage, the system learns to solve easy problems or something that needs to know beforehand and complex problems or target tasks in the later stages.

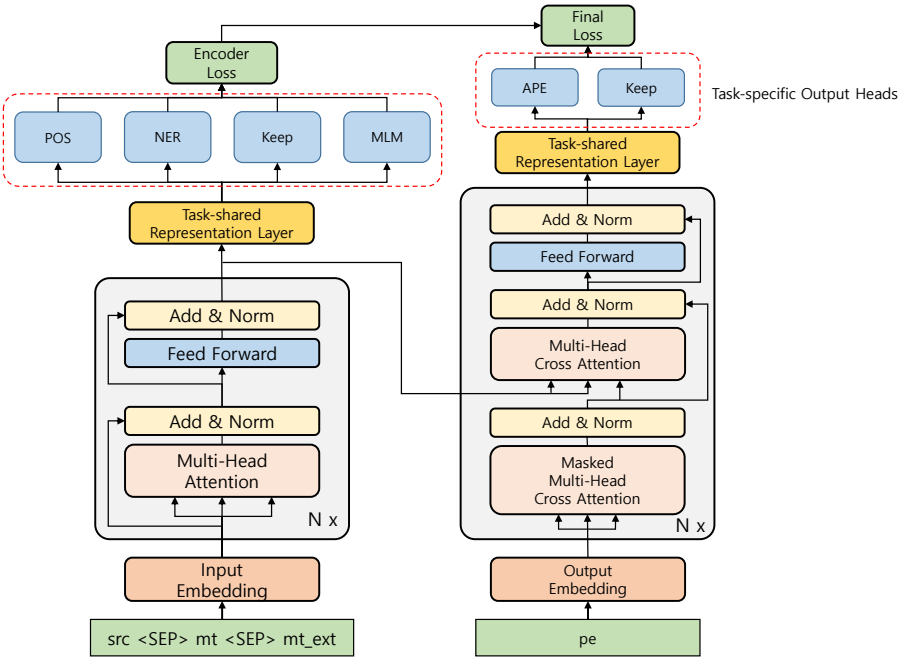


Figure 1: Overall architecture

3.1 Step 1: Understanding for Machine Translation

$$X = [src], \tag{2}$$

The APE task has to understand the machine translation system because the APE task modifies the mt results. Therefore, we designed the first step of the curriculum with the input as Equation 2 and the target as pe.

3.2 Step 2: Learning about Post-Editing

$$X = [src \texttt{<SEP>} mt], \tag{3}$$

After the first step, our system understands as the machine translation system. In this step, we make our system learn how to edit mt to pe with the input as Equation 3 and the target as pe.

3.3 Step 3: Post-Editing with External MT

For the second step, our system learns about the post-editing mechanism. In this step, we make the system learn to take the External MT into account with the input as Equation 1 and the target as pe.

3.4 Fine-Tuning

Finally, we fine-tune the APE system using the data given in the challenge with the input as Equation 1 and the target as pe.

4 Multi-task Learning Strategy (MLS)

Existing works for MTL propose jointly learning methods among related tasks. MTL aims to improve the generalization performance of the whole tasks by sharing knowledge representations of other tasks and can also alleviate the data sparsity problem where each task has limited labeled data (Zhang and Yang, 2021). Therefore, we utilize MLS for our system because WMT21 APE shared task provides only 7,000 train sentences. In NMT, existing works for MTL applied POS, NER, or MLM as subtasks and provided improved results (Chatterjee et al., 2017; Wang et al., 2020). Despite the impressive results, they applied only a few subtasks, such as one or two. Since we defined the APE task as NMT alike problem in our work, it would be helpful to leverage these subtasks into our work to achieve better performance. We find out that all these subtasks are cooperative with each other and benefit our system. Inspired by the word-level quality estimation task, we also add the Keep/Translate classification tasks for encoder and decoder to handle the high-quality APE task, which is described in Section 4.2 in detail. Since utilizing multiple subtasks, we have to consider the loss ratio between these subtasks. In our work, we apply the Dynamic Weight Average method described in Liu et al. (2019), and more details are described in Section 4.4. Our final system based on the model

post-trained using CTS with fine-tuning the APE data with MLS.

4.1 Architecture

Our architecture is described in Figure 1. The overall flow of the APE task is the same in Section 2. In this section, we explain five auxiliary subtasks consisting of POS, NER, MLM, Keep/Translate for the encoder, and Keep/Translate for the decoder. For the encoder, the encoding vector H is fed into Task-shared Representation Layer in Figure 1 like a Fully-connected Neural Network (FNN), and the output is represented as,

$$H_s = (W_1 H + b_1), \tag{4}$$

where $W_1 \in \mathbb{R}^{d_h^s \times d_h}$, and d_h^s represents a dimension of the Task-shared Representation Layer.

4.2 Subtasks

POS & NER POS and NER task aims to predict parts of speech and named entities about an input sequence, respectively. Task-shared Representation Layer H_s is fed into Task-specific Output Heads on Figure 1 like a FNN, and the output is represented as,

$$\hat{Y}^{pos} = \text{softmax}(W_2 H_s + b_2), \tag{5}$$

where $W_2 \in \mathbb{R}^{C_{pos} \times d_h}$ is trainable parameters and C_{pos} is the number of class of POS task. The parameters of Task-specific Output Heads for POS task are represented as Θ_{pos}. Likewise, $\hat{Y}^{ner}$ is obtained as in Equation 5 for NER task, where the parameters are represented as Θ_{ner}.

MLM In MLM task, we copy the input tokens from X to X^{mlm}, which is represented by $X^{mlm} = \{x_1, ..., x_{n+m+l+2}\}$, where n, m, l represents the number of tokens for src, mt, mt_ext, respectively. Then, we randomly mask 15% of the tokens X^{mlm} using the special token $mask$, and define the target as original input tokens. X^{mlm} is fed into the encoder. Then, the output representation is used to the input for Task-specific Output Heads for MLM task as,

$$\begin{aligned} \hat{Y}^3 &= \text{softmax}(W_3 H_s + b_3), \\ \hat{Y}^{mlm} &= \{\hat{Y}_r^3 | x_r = mask, \\ &\quad \forall\, r \in \{0, ..., n+m+l+2\}\} \end{aligned} \tag{6}$$

where $W_3 \in \mathbb{R}^{C_{mlm} \times d_h}$ represents trainable parameters and C_{mlm} is the number of vocab for the encoder. The parameters of a linear projection layer are represented as Θ_{mlm} for MLM task.

Keep/Translate Considering the characteristics of the APE data with relatively low TER scores, we decide to add Keep/Translate classification subtask to both Encoder and Decoder in our APE system. Keep/Translate subtask aims to predict the labels of the input sequence, where is $\hat{Y}^{kt} \in \{Keep, Translate\}$. In this subtask, each token in the input will be labeled with $Keep$ or $Translate$. For label generation, we apply to the pair of src-mt and src-mt. First, we use SimAlign (Jalili Sabet et al., 2020) to perform word alignment on the pe-mt pair. To each aligned word pair, we labeled them with Keep if they are equal. Otherwise, they will be marked as Translate. As for the pair of src-mt, we also do word alignment to find the correspondence between the source and target side. On the src side, the tokens are labeled with the same name as the corresponding words on the mt side. In our case, the same procedure on pe-mt is conducted for the pair of mt_ext and pe because we use the mt_ext as our data augmentation method. Figure 2 shows an example of label generation in the Keep/Translate task for better understanding. The output is represented as,

$$\hat{Y}^{kt} = \text{softmax}(W_4 H_s + b_4), \tag{7}$$

where $W_4 \in \mathbb{R}^{C_{kt} \times d_h}$ is trainable parameters and C_{kt} is the number of class of Keep/Translate task. The parameters of Task-specific Output Heads for Keep/Translate task are represented as Θ_{kt} and $\hat{Y}^{kt}$ is obtained as in Equation 7 for Keep/Translate task.

4.3 Loss

As described above, five subtasks are used in our system, and most of them have data with imbalanced labels. The imbalanced ratio reaches 1:2160, 1:15, and 1:6 between minority and majority classes in POS Tagger, NER, and Keep/Translate subtasks, respectively. With such imbalanced data, the Cross-Entropy loss used in classification problems may result in performance degradation in some tasks. To improve the performance, the Focal loss (Lin et al., 2017) is considered as an alternative candidate because a Focal Loss function addresses class imbalance during training in tasks. It applies a modulating term to the cross-entropy loss in order to focus on learning the hard negative examples. It reduces the relative loss for well-classified examples ($p_t > 0.5$), putting more focus on hard, misclassified examples. Equation 8 describes the

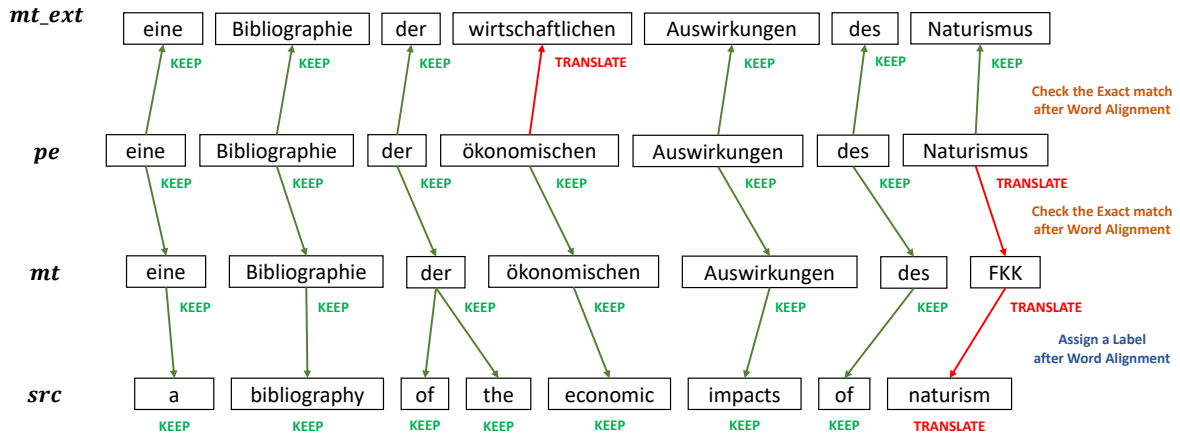


Figure 2: A label generation example in the Keep/Translate task

Focal Loss, where p_t is the probability of each class predicted by the model and γ represents the focusing parameter. Considering the imbalanced property of each task, we apply the Focal Loss to three of our subtasks, such as POS Tagger, NER, and Keep/Translate in the decoder.

$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t) \tag{8}$$

Class Balanced Loss is designed to use a re-weighting scheme that uses the effective number of samples for each class to re-balance the loss, thereby yielding a class-balanced loss (Cui et al., 2019). As the number of samples increases, there is information overlap among data. Therefore, the marginal benefit that a model can extract from the data diminishes. The effective number of samples, which played as the expected volume of samples, is used to capture the diminishing marginal benefits by using more data points of a class. For Keep/Translate task in the decoder, it just considered the PE as input, so we applied the Focal Loss to the subtask. However, for Keep/Translate task in the encoder, as one of the data augmentation methods, the external MT is also considered as input along with the src and mt. As the information of input increases, we think it may cause information overlap among data because mt and the mt_ext have the most in common. Therefore, we apply the Class-Balanced Loss as our loss function in Keep/Translate subtask in the encoder. Equation 10 describes the Class-Balanced Loss (L_{cb}), where C is the total number of classes, z_y is the output from the model for class y, n_y is the number of samples in the ground-truth class and $\beta \in [0, 1)$ is a hyperparameter which can be calculated in Equation 9. In Equation 9, i denotes the class index, $i \in \{1, 2, ..., C\}$, and N is the number of samples.

As for the MLM task, since it does not suffer from the data imbalance problem, we use the Cross-Entropy loss in our work as other works do.

$$\begin{aligned} N_i &= N, \\ \beta_i &= \beta = (N-1)/N \end{aligned} \tag{9}$$

$$L_{cb} = -\frac{1-\beta}{1-\beta^{n_y}} \log\left(\frac{\exp(z_y)}{\sum_{j=1}^{C}\exp(z_j)}\right) \tag{10}$$

4.4 Dynamic Weight Average

For most Multi-Task learning networks, it's difficult to find the best ratio between each task in subtasks manually. Therefore, we apply the Dynamic Weight Average (DWA) (Liu et al., 2019) to our work, which adapts the task weighting over time by considering the rate of change of the loss for each task.

Equations 11 and 12 describe DWA. Here, $\lambda_k(\cdot)$ represents the weighting for task k, $w_k(\cdot)$ calculates the relative descending loss rate for each task in each epoch, t is an iteration index, and T represents a temperature that controls the softness of task weighting. $\mathcal{L}$ in Equation 12 is the loss value, calculated as the average loss in each epoch over several iterations.

$$\lambda_k(t) := \frac{Kexp(\omega_k(t-1)/T)}{\sum_i Kexp(\omega_i(t-1)/T)} \tag{11}$$

$$\omega_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)} \tag{12}$$

	TER	BLEU
CTS-best (ensemble)	**16.44**	71.88
CTS-best (single)	16.46	**71.94**
w/o *step* 2	16.70	71.84
w/o *step* 3	17.33	70.24
w/o *step* 2 & 3	17.28	70.88
baseline	19.06	68.79

Table 1: CTS results on WMT21 APE development dataset. CTS-best (ensemble) is built by two similar single models. we submitted CTS-best (ensemble) as CONTRASTIVE result.

4.5 Joint Learning Procedure

All tasks are jointly trained, and the objective is defined as,

$$\mathcal{L} = \frac{1}{K}\sum_{i}^{K} \lambda_{\text{i}} \mathbb{L}(Y_i, f(X_i)), \qquad (13)$$

where λ is a dynamic weight determining the degree of subtasks and f is the training classifier. Note that the parameter K is the number of subtasks. $\mathbb{L}(Y, f(X))$ is the loss of f w.r.t. the target Y.

5 Experiments

5.1 Datasets

Following existing works, we utilize additional resources (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018), which have source sentences (src), machine translation sentences (mt), and post-editing sentences (pe). Moreover, we also utilize some of News-Translation data for the WMT21 (Koehn, 2005; Tiedemann, 2012; Rozis and Skadiņš, 2017; Bhatia et al., 2016; Tiedemann, 2012), which has source sentences (src) and translated sentences that can be used as pe. For evaluation and fine-tuning, we use the data for WMT21 automatic post-editing shared task. Moreover, we utilize translated sentences using Google Translate and Quality Estimation NMT Model (Fomicheva et al., 2020). The former is used to make mt_ext from the additional resources and the data for WMT21 automatic post-editing. The latter is used to make mt from News-Translation data. We filtered all the training data based on and number checking logic, which filters the pairs with different numbers in source and target side.

	TER	BLEU
MLS w DWA	**16.21**	**72.53**
MLS w/o DWA	16.37	72.34

Table 2: Ablation analysis of DWA on the WMT21 APE development dataset.

5.2 Experimental Settings

For the first step of CTS, we utilize WMT19 en-de weights by Fairseq (Ng et al., 2019). In the second step, we utilize News-Translation data with translated sentences with Quality Estimation NMT Model as mt. In the third step, we make our system learn with Junczys-Dowmunt and Grundkiewicz (2016); Negri et al. (2018) and Google Translate as mt_ext. Finally, when learning the fine-tuning step, which contains MLS, we utilize the data for WMT21 Automatic Post-Editing shared task.

5.3 Results: CTS

To study the effectiveness of CTS, we conduct ablation experiments on WMT21 Automatic Post-Editing development dataset. We set the $baseline$, which is a system that leaves all the test instances unmodified. As shown in Table 1, we can observe that the $step$ 3 is more effective than the $step$ 2, and that using only $step$ 2 doesn't help APE. As our system is learning step by step with CTS, it allows that our system has strengths in the APE task.

5.4 Results: MLS

Table 2 presents the ablation analysis about DWA when fine-tuning with MLS on WMT21 APE development set. From the result, we can observe that MLS with DWA has better performance than the one without applying it. For that reason, we adopt DWA at a fine-tuning stage with MLS in our APE Task.

To find the best combination of subtasks in MLS, we conducted an ablation analysis on the same development dataset. Vanilla in the table is a system without adding any subtasks. We add the subtasks one by one during the fine-tuning to see the effect of each subtask on the performance. As shown in Table 3, the one using all the subtasks performs best among all the combinations, which means that these subtasks are cooperative in the APE task.

	TER	BLEU
Vanilla	16.71	71.75
w/ POS	16.49	72.12
w/ NER	16.52	72.19
w/ MLM	16.55	72.00
w/ Keep/Translate	16.45	72.32
Fine-tuned with MLS	**16.21**	**72.53**

Table 3: The Multi-task Learning results on WMT21 APE validation dataset. Fine-tuned with MLS using all subtasks model is submitted as PRIMARY result.

	TER	BLEU
Netmarble_CONTRASTIVE	**17.28**	**71.55**
Netmarble_PRIMARY	17.97	70.53
baseline	18.05	71.07

Table 4: Official results on WMT21 APE test dataset.

5.5 Official Results

Table 4 shows the official results of our proposed methods on WMT21 test dataset. The test dataset has baseline scores of 18.05 and 71.07, which is higher than the development dataset with 19.06 and 68.79 in terms of TER and BLEU, respectively. Despite its high quality, our proposed methods showed effectiveness on this test dataset.

5.6 Implementation Details

We set the batch size to 256 for the *step* 2 and *step* 3 in CTS at each GPU, 16 for the fine-tuning and MLS. We set the initial learning rate to 1e-4 using scheduler in Fairseq (Ng et al., 2019) for all experiments. The average runtime of one epoch for each approach was about 360 minutes for the *step* 2, 90 minutes for the *step* 3, and 40 seconds for MLS. We train our models using AdamW (Loshchilov and Hutter, 2019) optimizer and conduct experiments with 16 Tesla A100 GPUs for CTS, Tesla V100 GPU for MLS.

6 Conclusion

In this paper, we propose an APE system based on CTS and MLS. CTS allows understanding between machine translation and automatic post-editing, and shows a way using additional data in large volume in APE task. MLS learns a shared unified representation from related subtasks to improve the performance. We submitted the system, which Fine-tunes with MLS, as our primary version and the ensembled CTS as our contrastive version. The experimental results show that our system is able to effectively detect and correct the errors made by a high-quality NMT system, improving the score by -2.848 and +3.74 on the development dataset in terms of TER and BLEU, respectively. Our proposed methods also achieved performance improvement on the test dataset with higher quality.

References

K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

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, Volume 1: Research Papers*, pages 157–168, Copenhagen, Denmark. Association for Computational Linguistics.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. 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, Berlin, Germany. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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).

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Shinhyeok Oh, Dongyub Lee, Taesun Whang, IlNam Park, Seo Gaeun, EungGyun Kim, and Harksoo Kim. 2021. Deep context- and relation-aware learning for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 495–503, Online. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. 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'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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*, volume 30. Curran Associates, Inc.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034, Online. Association for Computational Linguistics.

Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2021. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14041–14049.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. Hw-tsc's participation at wmt 2020 automatic post editing shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 797–802, Online. Association for Computational Linguistics.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Adapting Neural Machine Translation for Automatic Post-Editing

Abhishek Sharma* **Prabhakar Gupta** **Anil Nelakanti**
Amazon Prime Video
{naabhiss, prabhgup, annelaka}@amazon.com

Abstract

Automatic post-editing (APE) models are used to correct machine translation (MT) system outputs by learning from human post-editing patterns. We present the system used in our submission to the WMT'21 Automatic Post-Editing (APE) English-German (En-De) shared task. We leverage the state-of-the-art MT system (Ng et al., 2019) for this task. For further improvements, we adapt the MT model to the task domain by using WikiMatrix (Schwenk et al., 2021) followed by fine-tuning with additional APE samples from previous editions of the shared task (WMT-16,17,18) and ensembling the models. Our systems beat the baseline on TER scores on the WMT'21 test set.

1 Introduction

Automatic Post-Editing (APE) is the task of automatically correcting machine translation (MT) outputs. Along with fixing systematic errors in MT outputs, APE models can adapt general purpose MT systems to new domains and provide better translations to reduce the human post-editing effort (Chatterjee et al., 2015). APE has seen significant progress with Transformer based models (Yang et al., 2020; Lopes et al., 2019; Chatterjee et al., 2019, 2020) dominating the landscape as opposed to the earlier Statistical Machine Translation (SMT) based models (Simard et al., 2007; Béchara et al., 2012) and RNN based sequence-to-sequence models (Junczys-Dowmunt and Grundkiewicz, 2017). To track this progress, WMT has been conducting APE shared tasks since 2015 on different data domains and language pairs (Bojar et al., 2015, 2016, 2017; Chatterjee et al., 2018, 2019, 2020).

WMT 2021's shared task focused on English-German and English-Chinese language pairs. We participated in the English-German sub-task and describe our submission in this paper. Participants were provided a training set with 7000 instances and a development set with 1000 instances. Each dataset consisted of *source*, *machine-translation*, *post-edit* triplets. The source sentences came from the English Wikipedia, the MT outputs were generated with a black-box state-of-the-art MT system and the post-edits were created by professional translators correcting MT outputs. The test set consisted of 1000 pairs of source and MT outputs for which the participants had to submit the post-edits generated by their systems. The task organisers provided two additional synthetic post-editing datasets – 'artificial training data' (Junczys-Dowmunt and Grundkiewicz, 2016) and 'eSCAPE corpus' (Negri et al., 2018) and permitted using additional data to train the model. TER scores (Snover et al., 2006) and BLEU (Papineni et al., 2002) scores were used as primary and secondary evaluation metrics respectively.

Last year's entries primarily focused on transfer learning (Yang et al., 2020; Lee, 2020; Wang et al., 2020) and novel data augmentation techniques (Lee et al., 2020b,a; Wang et al., 2020). The winning submission (Yang et al., 2020) was based on fine-tuning a pre-trained machine translation model for the APE task.

We take a similar line of approach by leveraging an existing state-of-the-art machine translation model. We first fine-tune an MT model on WikiMatrix (Schwenk et al., 2021) — a mined bitext from Wikipedia — to bridge the domain gap, followed by further tuning to the APE task with post-editing samples. To deal with the limited training data, we exploit APE data from the previous editions of the WMT shared tasks. We describe the details of our experiments in Section 3 with gains and observations from individual tuning steps mentioned above.

*Work done as intern at Amazon Prime Video

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 315–319
November 10–11, 2021.

2 Related Work

The last year's WMT'20 APE shared task saw methods using transfer learning with data augmentation techniques perform well. Yang et al. (2020) fine-tune state-of-the-art transformer-based MT system on APE data using bottleneck adapter layers (Houlsby et al., 2019) to avoid overfitting. They additionally use outputs from an external MT system as input to the model and converged to ensembling to achieve 66.89 BLEU score on the WMT'20 development set to make it to the top of the final leaderboard.

Data augmentation techniques where post-edits are synthesized to augment human-edited data was shown to be effective in the last year's submissions for addressing the training data limitation. However, data augmentation must be done carefully to prevent a mismatch between the error distributions in gold and synthetic data (Yang et al., 2020). Wang et al. (2020) use data augmentation along with dual conditional cross-entropy model (Junczys-Dowmunt, 2018) based filtering to ensure data quality, model adaptation to target domain, and ensembling to achieve 56.06 BLEU on the development set and the second rank on the leaderboard. Similarly, Lee et al. (2020b) performed data augmentation by creating a novel noising scheme to synthesize four kinds of errors for APE training, namely, insertion, deletion, substitution and shifting/reordering noise to attain 53.77 BLEU score.

The other submissions to the WMT'20 task used variations of the language models to generate edits. Lee et al. (2020a) trained a model by jointly optimizing losses for masked language and translation language models while Lee (2020) tailored a language model to make corrections by replacing poor quality words to improve the overall sentence-level quality. These two submissions were able to get 55.67 and 53.82 BLEU scores respectively on the WMT'20 development set.

In comparison, our model is a pre-trained MT model adapted to the target domain and further fine-tuned on the APE data. These improvements give us about five absolute points gain over the no post-editing baseline (that returns MT output without changes) on the BLEU score to arrive at 55.85 which is competitive with all but one of last year's submissions on the WMT'20 development set.

Dataset	Train	Dev	Test	Domain
WMT'16	12000	1000	2000	IT
WMT'17	11000	–	2000	IT
WMT'18	13442	1000	3023	IT
WMT'21	7000	1000	1000	Wikipedia

Table 1: WMT APE shared task data for En-De

3 Method

We describe our baseline model followed by the details of domain and task adaptation in this section.

3.1 Baseline translation model

Limited by availability of training data, we used transfer learning approach (as is common in related tasks with few samples, see Ruder et al. (2019)) beginning with a pre-trained MT model. We used the MT models from FAIR's WMT'19 submission[1] (Ng et al., 2019) that is an ensemble trained for the News Translation task using fairseq (Ott et al., 2019) library. It takes a single source sentence as input and returns translation in the target language. To use this model for the APE task, we concatenated the *source* and the *machine-translation* with a special token to make the input. Thus, we fine-tune the NMT model on the APE dataset with `source <sep> machine-translation` as input and `post-edited` reference as the output.

3.2 Pre-training on domain-specific data

FAIR's WMT'19 NMT model was trained on Newscrawl and Commoncrawl datasets while the source of this year's APE data is Wikipedia. To fix the domain mismatch in NMT model's training data and our task, we fine-tune the NMT model on WikiMatrix (Schwenk et al., 2021) before fine-tuning the model with APE data. WikiMatrix is mined from Wikipedia using the multi-lingual sentence embeddings from the LASER toolkit (Artetxe and Schwenk, 2019). We ensure that the model is fine-tuned on only high-quality parallel data by using a higher threshold of 1.1 for extracting parallel sentences (rather than the default 1.04) to get $64k$ parallel sentences.

3.3 Fine-tuning on APE data

To further address the data limitation, we use samples from earlier editions of the APE shared task; WMT'16, WMT'17 and WMT'18. Although the

[1] `transformer.wmt19.en-de`

Model	BLEU↑	TER↓
Do Nothing	68.79	19.06
MT fine-tuned on WMT'21	68.74	18.45
MT fine-tuned on (WMT'16-18 + WMT'21) (A)	69.34	18.27
MT fine-tuned on WikiMatrix and further on (WMT'16-18 + WMT'21) (B)	69.12	18.34
Ensemble (A + B)	**69.38**	**18.18**

Table 2: Results on the WMT 2021 APE development set. Higher BLEU and lower TER is better. The "Ensemble" model is the ensemble of the two best performing single models (the ones with 69.12 and 69.34 BLEU scores).

Model	BLEU↑	TER↓
Do Nothing	**71.07**	18.05
Model (A)	70.54	**17.74**
Ensemble (A + B)	70.50	17.85

Table 3: Results on the WMT 2021 APE test set. Higher BLEU and lower TER is better. The Model (A) is the one described in the same from Table 2 and the "Ensemble" model is the ensemble of the two best performing single models.

domain of the data in the previous editions of this shared task challenge is different from the current one, we preferred using this data over synthetic APE data similar to (Yang et al., 2020). We prefer this because unlike in WMT datasets where the post-edits are human revisions of the MT output, synthetic APE datasets have post-edited sentences independent of the MT output, causing the error patterns and data distributions to vary significantly. Hence, we combine the WMT'16, WMT'17 and WMT'18 datasets to get $45k$ *source*, *machine-translation* and *post-edit* triplets. We present the details of the data in Table 1.

4 Results and conclusion

We report the results of our model on the WMT'21 development and test set. We use BLEU scores (Papineni et al., 2002) [2] for quality estimates relative to a human reference and TER scores (Snover et al., 2006) for quantifying human post-editing effort.

We report improvements over the `Do Nothing` baseline. This baseline refers to the system that returns the base machine translation output as the post-edit without any changes. We submitted the best performing single model and the ensemble model in Table 2 for evaluation. In Table 3 we present the results reported by the organizers for baseline, our model fine tuned on WMT'16-18 + WMT'21 (model A) and our ensemble model (A + B). The Do Nothing baseline from last year (Chatterjee et al., 2020) was reported at 50.21 BLEU score and this year it is reported at 71.07 BLEU score. These numbers suggest that the baseline machine translation engine used in this year's task proved to be of very high quality for the dataset used; leaving very little room for APE models to improve the translation similar to the observation made in (Chatterjee et al., 2018). This is the only logical conclusion we could draw since the data used last year and this year are the same with human post-editing re-done. Using data from previous years' tasks clearly improves both BLEU and TER scores on the development set. While fine-tuning on WikiMatrix data itself has not led to improvements on the development set, it helps improve performance when used in ensemble with the other model. The model A beats the baseline on TER metric by 0.31 points on the test set while both our model A and ensemble system manage to outperform previous year's best entry.

Further extending this work, we wish to study more carefully the impact of adaptation by switching the order of domain and task adaptation, effect of noise in training sample by tuning threshold (Wieting and Gimpel, 2018), and evaluate if synthetic data can be selectively augmented for greater metric gains.

[2] calculated using `multi-bleu.perl` script from the Moses toolkit (Koehn et al., 2007)

References

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Hanna Béchara, Raphaël Rubino, Yifan He, Yanjun Ma, and Josef van Genabith. 2012. An evaluation of statistical post-editing systems applied to RBMT and SMT systems. In *Proceedings of COLING 2012*,

pages 215–230, Mumbai, India. The COLING 2012 Organizing Committee.

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, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, 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, 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, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. 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: Shared Task Papers*, pages 710–725, Belgium, Brussels. 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. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. 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, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. 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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Dongjun Lee. 2020. Cross-lingual transformers for neural automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 772–776, Online. Association for Computational Linguistics.

Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee. 2020a. POSTECH-ETRI's submission to the WMT2020 APE shared task: Automatic post-editing with cross-lingual language model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 777–782, Online. Association for Computational Linguistics.

WonKee Lee, Jaehun Shin, Baikjin Jung, Jihyung Lee, and Jong-Hyeok Lee. 2020b. Noising scheme for

data augmentation in automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 783–788, Online. Association for Computational Linguistics.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. 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).

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. 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. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich 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: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Jiayi Wang, Ke Wang, Kai Fan, Yuqi Zhang, Jun Lu, Xin Ge, Yangbin Shi, and Yu Zhao. 2020. Alibaba's submission for the WMT 2020 APE shared task: Improving automatic post-editing with pre-trained conditional cross-lingual BERT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 789–796, Online. Association for Computational Linguistics.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. HW-TSC's participation at WMT 2020 automatic post editing shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 797–802, Online. Association for Computational Linguistics.

ISTIC's Triangular Machine Translation System for WMT' 2021

Hangcheng Guo, Wenbin Liu, Yanqing He*, Tian Lan,
Hongjiao Xu, Zhenfeng Wu, You Pan
Institute of Scientific and Technical Information of China, Beijing, China
{guohc2020, liuwb2019, heyq, lantian,
xuhj, wuzf, pany}@istic.ac.cn

Abstract

This paper describes the ISTIC's submission to the Triangular Machine Translation Task of Russian-to-Chinese machine translation for WMT' 2021. In order to fully utilize the provided corpora and promote the translation performance from Russian to Chinese, the pivot method is used in our system which pipelines the Russian-to-English translator and the English-to-Chinese translator to form a Russian-to-Chinese translator. Our system is based on the Transformer architecture and several effective strategies are adopted to improve the quality of translation, including corpus filtering, data pre-processing, system combination, model averaging, model ensemble and reranking.

1 Introduction

The Institute of Scientific and Technical Information of China (ISTIC) participated in the Triangular Machine Translation Task of Russian-to-Chinese in the Sixth Conference on Machine Translation[1] (WMT' 2021). This paper demonstrates the overall framework of the ISTIC's submission and its technical details.

In this evaluation, we adopted the neural machine translation architecture of Google Transformer(Vaswani et al., 2017) as a part of our system. We use the three parallel corpora released by the evaluation organizer and adopted a two-stage method for data pre-processing. Several filtering methods of the corpus are explored to reduce the data noise and improve the data quality. As for model construction, we use the pivot method to get a Russian-to-Chinese translator by bridging the trained Russian-to-English translator and English-to-Chinese translator. Model averaging(Claeskens and Hjort, 2008), model ensemble(Lutellier et al., 2020) and reranking(Ng et al., 2019) strategies are adopted to generate the final output translation. We removed spaces between words and restored the target language translation results to the prescribed file format in data post-processing. In our experiment, the performance of the system under different settings was compared and further analyzed the experimental results.

The structure of this paper is as follows: the second part introduces the technical architecture of our machine translation system; the third part describes the data pre-processing, parameter settings, experimental results, and related analysis; the fourth part gives the conclusion and future work.

2 System Overview

The overall framework of the ISTIC's triangular machine translation system is shown in Figure 1.

2.1 Single Transformer System

Our baseline single system used in participated evaluation tasks is the Transformer based encoder-decoder architecture. Transformer is completely based on a self-attention mechanism. It can achieve algorithm parallelism, speed up model training, further alleviate long-distance dependence and improve translation quality(Zhang and Zong, 2020). The encoder and the decoder are formed by stacking N identical layer blocks, where N is set to 6.

2.2 Context-based Combination System

As shown in Figure 2, based on the Transformer model, our team adopts a context-based(Voita et al., 2018) system combination method, which is an encoder-decoder structure composed of n identical network layers, where n is set to 6. Two different methods of system combination are designed according to the fusion in different positions, which are Encoder Combination method and Decoder Combination method. Both of them adopt multi-encoder(Li et al., 2020) to encode the source sen-

*Corresponding author: Yanqing He, heyq@istic.ac.cn.

[1] http://www.statmt.org/wmt21/triangular-mt-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 320–324
November 10–11, 2021.

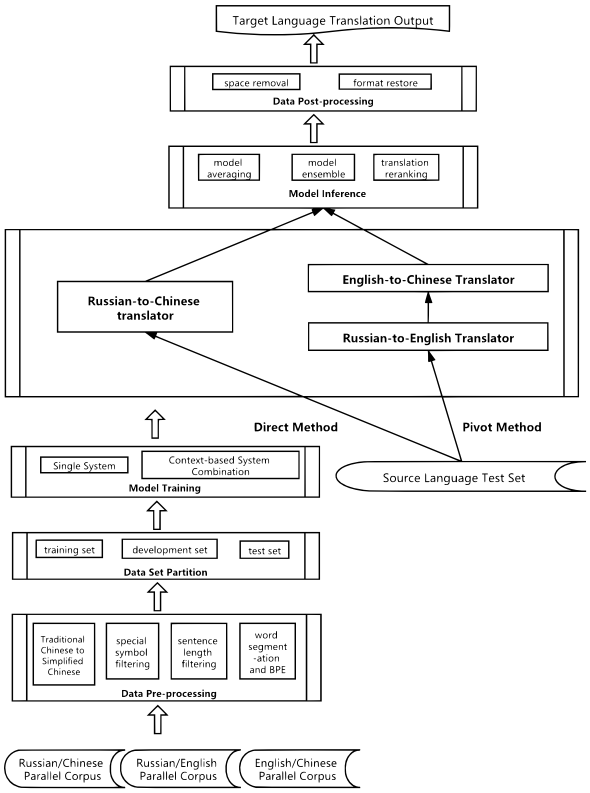


Figure 1: Overall framework

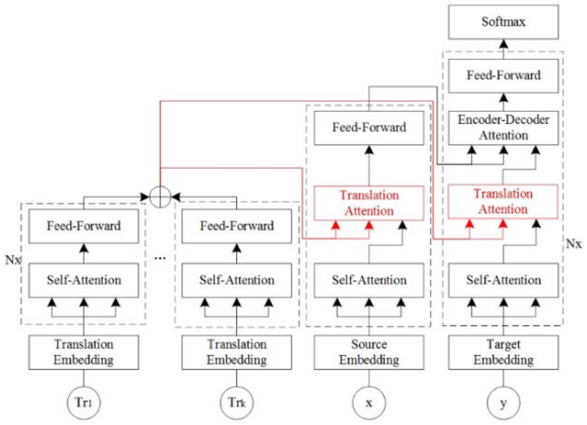


Figure 2: Context-based combination system

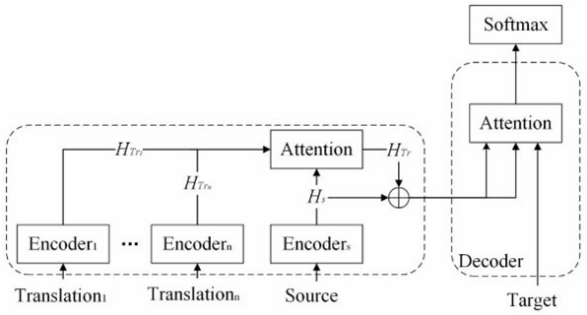


Figure 3: Encoder combination model

tences and the context information from machine translation results of the source sentence. In the Encoder Combination method, the hidden layer information of context (multi-system translation) is transformed into new representation through attention network, and merges the hidden layer information of source sentence through gating mechanism at encoder end; In Decoder Combination method, the hidden layer information of multi-system translation and the hidden layer information of source sentence is calculated at the decoder to obtain the fusion vector. The attention calculation method is the same as the original transformer model, to obtain a higher quality fusion translation.

The Encoder Combination model (see Figure 3) uses multiple system translations, and then converts the system translations into new representations through the attention network, integrating the hidden layer information of homologous language sentences for attention fusion through the gating mechanism in the Encoder. In the Encoder Combination mode and the Self-Attention of the multi-system translation Encoder, Q, K, and V are all from the upper layer output of the multi-system translation Encoder; in the Self-Attention of the source language Encoder, Q, K, and V are all from the upper layer output of the source language Encoder; in the Translation Attention of the source language Encoder, both K and V come from the upper hidden layer state H_{T_r} of the multi-system translation Encoder, and Q comes from the upper layer hidden state H_s of the source language Encoder. H_s represents the hidden state of the source language sentence, H_{T_r} represents the hidden state of the multi-system translation, and H represents the hidden state of the Translation Attention part of the Encoder.

$$H_{T_r} = Concat(H_{T_r1}, ..., H_{T_rn}) \quad (1)$$

$$H = MutiHead(H_{T_r}, H_s) \quad (2)$$

The Decoder Combination model (see Figure 4) combines the hidden layer information of multiple encoders with attention in the decoder. The Decoder can process multiple encoders separately, and then fuse them using the gating mechanism inside the Decoder to obtain the combined vector. In the Decoder Combination mode and the Self-Attention of the target language Decoder, Q, K, and V are all from the output of the previous layer of the target language Decoder; in the Translation Attention of the target language Decoder, Q comes from the

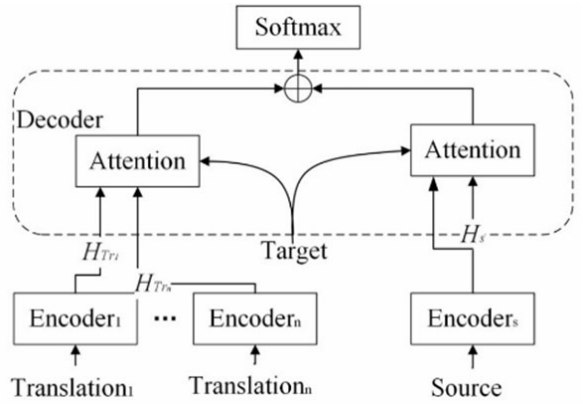


Figure 4: Decoder combination model

output of the upper layer of the target language Decoder, K comes from the upper hidden layer state H_s of the source language Encoder, and V comes from the upper hidden layer state H_{T_r} of the multi-system translation Encoder; in the Encoder-Decoder Attention of the target language Decoder, Q comes from the upper layer output of the target language Decoder, K, V come from the previous output of the source language Encoder. H_s represents the hidden layer state of the source language sentence, H_{T_r} represents the hidden layer state of the multi-system translation, $H_{Decoder}$ represents the hidden layer state of the upper layer output of the Decoder, and H represents the hidden state of the Translation Attention part of the Decoder.

$$H = MutiHead(H_{T_r}, H_s, H_{Decoder}) \quad (3)$$

2.3 Direct Method

In the direct method (see Figure 5), we use the pre-processed Russian/Chinese parallel corpus to train a direct Russian-to-Chinese translator by means of the single Transformer System or the context-based Combination System, depending on which kind of system performs best.

2.4 Pivot Method

In the pivot method (see Figure 5)(Park and Zhao, 2019), firstly, we use the pre-processed Russian/English parallel corpus to train a Russian-to-English translator; secondly, we use the pre-processed English/Chinese parallel corpus to train an English-to-Chinese translator; finally, we pipeline them to form a pivot Russian-to-Chinese translator. All translators can be trained by means of the single Transformer System or the context-based Combination System. By comparing the

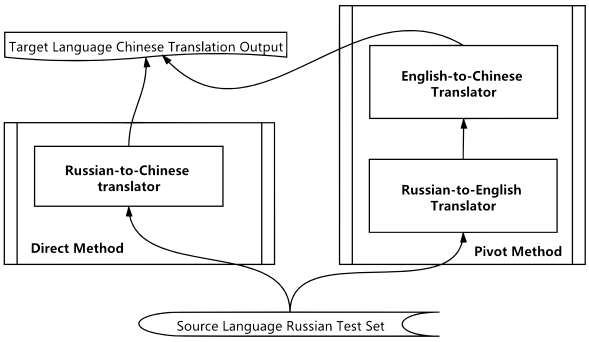


Figure 5: Direct and pivot method

experimental results, the system with optimal performance is accepted for Russian-to-Chinese translation.

3 Experiments

3.1 Data Pre-processing

The evaluation organizers provide three parallel corpora: the Chinese/Russian corpus is crawled from the web and aligned at the segment level, and combined with different public resources; the Chinese/English corpus combines several public resources; the Russian/English corpus gathers multiple public resources. A two-stage method(Wei et al., 2020) is used for data pre-processing, consist of a general pre-processing stage and a specific pre-processing stage. The general pre-processing stage includes conversion from traditional Chinese to simplified Chinese by the hanziconv[2] package , conversion between full angle and half-angle, special character filtering, same content filtering, sentence length filtering, and sentence length ratio filtering. Among them, sentence length of the Chinese language is calculated in the unit of "character" and sentence length of non-Chinese language is calculated in the unit of "token". Sentence length filtering removes sentence pairs which source sentence length or target sentence length exceeds the range of [5, 200]. Sentence length ratio filtering excludes the sentence pairs whose ratio of source sentence length and target sentence length exceeds the range of [0.2, 20]. In the specific pre-processing stage, the word segmentation of English and Chinese sentences is implemented using the lexical tool Urheen[3] and the word segmentation of Russian sentences is implemented using the lexical

[2] https://github.com/berniey/hanziconv

[3] https://www.nlpr.ia.ac.cn/cip/software.html

Direction	Before Pre-processing	After Pre-processing
Russian-English	69217438	42939395
English-Chinese	28579587	22233706
Russian-Chinese	33422682	21892537

Table 1: Data pre-processing results

Direction	Train Set	Dev Set	Test Set
Russian-English	42935974	2000	1421
English-Chinese	22231506	1100	1100
Russian-Chinese	21891537	965	1000

Table 2: Data partition results

tool Natasha[4]. The scales of sentence pairs of all corpora before and after data pre-processing are shown in Table 1.

After data preprocessing, we split the corpora into training set, development set and test set. The scales of the data partition are shown in Table 2.

3.2 System Settings

The open-source project fairseq[5](Ott et al., 2019) is chosen for this evaluation system. The main parameters are set as follows. Each model uses 1-3 GPUs for training, and the batch size is 2048. The embedding size and hidden size are set to 1024, the dimension of the feed-forward layer is 4096. We use six self-attention layers for both encoder and decoder, and the multi-head self-attention mechanism has 16 heads. The dropout mechanism(Provilkov et al., 2020) was adopted, and dropout probabilities are set to 0.3. BPE(Sennrich et al., 2016) is used in all experiments, where the merge operations is set to 32000. The maximum number of tokens is set to 4096. The loss function is set to "$label\_smoothed\_cross\_entropy$". The parameter "$adam\_betas$" is set to (0.9, 0.997). For the baseline system, the initial learning rate is 0.0007, the warm-up steps are set to 4000, and the maximum epoch number is set to 15. For the Encoder Combination system and Decoder Combination system[6], the initial learning rate is 0.0001, the warm-up steps are set to 4000, and the maximum epoch number is set to 10.

3.3 Experimental results

In the training of Russian-to-English translator, English-to-Chinese translator and Russian-to-Chinese translator, the single Transformer systems

[4] https://github.com/natasha/natasha

[5] https://github.com/pytorch/fairseq/tree/v0.6.2

[6] https://github.com/libeineu/Context-Aware

System	Russian-English	English-Chinese	Russian-Chinese
Transformer	20.89	17.83	16.07
Transformer+ Encoder Combination	21.53	18.87	16.66
Transformer+ Decoder Combination	21.76	18.91	16.79

Table 3: BLEU results on self-built test set

Method	BLEU
Primary: Pivot Method	19.2
Contrast: Direct Method	18.1

Table 4: BLEU results on released test set

are trained for 15 epochs. The context-based combination systems with Encoder Combination model or Decoder Combination model are trained for 10 epochs. The best epoch model and the last epoch model are ensembled to generate better results. The BLEU(Papineni et al., 2002) scoring results on the self-built test set are shown in Table 3.

The context-based combination systems with Decoder Combination model are used as our final submission since they outperform other systems.

Our primary submission uses the pivot method, which use English translation as the bridge. The Russian sentences are translated into English intermediate results by the well-trained Russian-to-English translator and then the English intermediate results are translated into Chinese output by the well-trained English-to-Chinese translator. Our contrast submission uses the direct method, which uses the well-trained Russian-to-Chinese translator to generate the target output.

As a result, our primary submission achieves a BLEU score of 19.2 and ranked the fourth among all participating teams. Our contrast submission achieves a BLEU score of 18.1 (shown in Table 4).

4 Conclusions

This paper introduces the main technologies and methods of ISTIC's submission in WMT 2021. To sum up, our model is constructed on the Transformer architecture of self-attention mechanism and context-based system combination method. In the aspect of data pre-processing, we explore several corpus filtering methods. In the process of translation output, the strategies of model ensemble and reranking are adopted. Experimental results show that these methods can effectively improve the quality of translation. It is worth mentioning that the pivot language translation bridge method

outperforms the direct translation method.

Acknowledgements

This research has been partially supported by ISTIC Fund ZD2021-17 and QN2021-12.

5 References

References

Gerda Claeskens and Nils Lid Hjort. 2008. *Model Selection and Model Averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518, Online. Association for Computational Linguistics.

Thibaud Lutellier, Hung Viet Pham, Lawrence Pang, Yitong Li, Moshi Wei, and Lin Tan. 2020. Coconut: Combining context-aware neural translation models using ensemble for program repair. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2020, pages 101—-114, New York, NY, USA. Association for Computing Machinery.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeonghyeok Park and Hai Zhao. 2019. Korean-to-chinese machine translation using chinese character as pivot clue. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33), Pacific Asia Conference on Language, Information and Computation*, pages 558–566.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. 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.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000—-6010, Red Hook, NY, USA. 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.

Jiaze Wei, Wenbin Liu, Zhenfeng Wu, You Pan, and Yanqing He. 2020. ISTIC's neural machine translation system for IWSLT'2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 158–165, Online. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2020. Neural machine translation: Challenges, progress and future. *CoRR*, abs/2004.05809.

HW-TSC's Participation in the WMT 2021 Triangular MT Shared Task

Zongyao Li, Daimeng Wei, Hengchao Shang, Xiaoyu Chen,
Zhanglin Wu, Zhengzhe Yu, Jiaxin Guo, Minghan Wang,
Lizhi Lei, Min Zhang, Hao Yang, Ying Qin,
Huawei Translation Service Center, Beijing, China
{lizongyao,weidaimeng,shanghengchao,chenxiaoyu35,
wuzhanglin2,yuzhengzhe,guojiaxin1,wangminghan,
leilizhi,zhangmin186,yanghao30,qinying}@huawei.com

Abstract

This paper presents the submission of Huawei Translation Service Center (HW-TSC) to WMT 2021 Triangular MT Shared Task. We participate in the Russian-to-Chinese task under the constrained condition. We use Transformer architecture and obtain the best performance via a variant with larger parameter sizes. We perform detailed data pre-processing and filtering on the provided large-scale bilingual data. Several strategies are used to train our models, such as Multilingual Translation, Back Translation, Forward Translation, Data Denoising, Average Checkpoint, Ensemble, Fine-tuning, etc. Our system obtains 32.5 BLEU on the dev set and 27.7 BLEU on the test set, the highest score among all submissions.

1 Introduction

This paper introduces our submission to the WMT21 Triangular task. We adopt Transformer (Vaswani et al., 2017) architecture and strictly obey the constrained condition in terms of data usage. On one hand, we perform multiple data filtering strategies to enhance data quality; on the other hand, we leverage multilingual model (Johnson et al., 2017), pivot language, forward (Wu et al., 2019) and back translation (Edunov et al., 2018), and data denoising (Wang et al., 2018) strategies to further enhance training effects. In addition, we also adopt fine-tuning (Sun et al., 2019) and ensemble (Garmash and Monz, 2016), two widely used strategies, to further enhance system performance. We compare and contrast different strategies based on our experiment results and give our analysis accordingly.

The overall training process is illustrated in Figure 1. Section 2 mainly focuses on our training techniques, including model architecture, data processing and training strategies. Section 3 describes our experiment settings and training process. Section 4 presents the experiment results while section 5 analyze how our multilingual, data denoise and data augmentation strategies influence system performances.

2 Method

2.1 Model Architecture

Our system uses Transformer (Vaswani et al., 2017) model architecture, which adopts full self-attention mechanism to realize algorithm parallelism, accelerate model training speed, and improve translation quality. In this shared task, Transformer-Deep (Wang et al., 2019) is used, which features 35-layer encoder, 6-layer decoder, 768 dimensions of word vector, 3072-hidden-state, 16-head self-attention, and pre-norm.

2.2 Data Processing an Augmentation

We strictly comply with the constrained condition and use only the officially provided data.

2.2.1 Data Filtering

We perform the following steps to cleanse all data:

- Filter out repeated sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Convert XML escape characters.
- Normalize punctuations using Moses (Koehn et al., 2007).
- Delete html tags, non-UTF-8 characters, unicode characters and invisible characters.
- Filter out sentences with mismatched parentheses and quotation marks; sentences of which punctuation percentage exceeds 0.3; sentences with the character-to-word ratio greater than 12 or less than 1.5; sentences of which the source-to-target token ratio higher

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 325–330

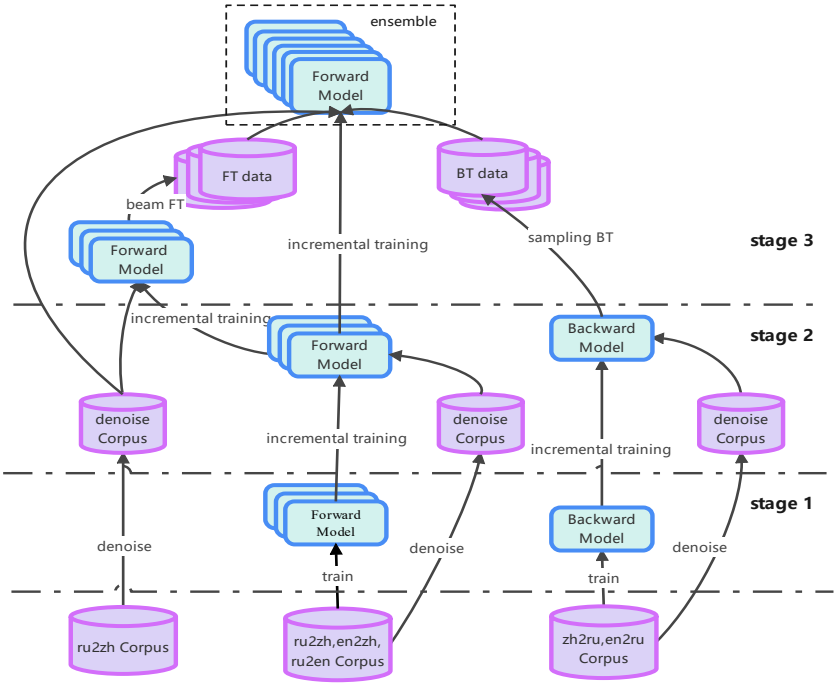


Figure 1: This figure shows the training process for the WMT 2021 Triangular MT Shared Task, which consists of three stages. In stage 1, three forward models and one backward model are trained. In stage 2, denoise corpus is used to train models incrementally. In stage 3, the synthetic data by FTST and denoise corpus are used to train models incrementally. Finally, model ensemble is used to boost the performance.

than 3 or lowers than 0.3; sentences with more than 120 tokens.

- Apply langid (Joulin et al., 2016b,a) to filter sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter sentence pairs with poor alignment, about 10% of the data is filtered.

We perform the additional steps to process Chinese data:

- Convert traditional Chinese characters to simplified ones.
- Convert fullwidth forms to halfwidth forms.

Data sizes before and after cleansing are listed in Table 1.

2.2.2 Data Augmentation

Back-translation (Edunov et al., 2018) is an effective way to boost translation quality by using monolingual data to generate synthetic training parallel data. As described in (Wu et al., 2019), similar to back translation, the monolingual corpus in source language can also be used to generate forward translation text with a trained MT model, and the generated forward and backward translation data can both be merged with the authentic bilingual data. This strategy can increase the data size to a large extent.

Since there is no officially provided monolingual data, we use the target side of en2zh data and the source side of zh2ru data filtered out in section 2.2.1 for back translation. We adopt the top-k sampling method. Then, we use the source side of ru2en data for forward translation, which is done based on beam search. Through sampling, we ensure that the sizes of data generated by forward and back translation are relatively equal. In this paper, we refer to the combination of forward and sampling back translation as FTST.

2.2.3 Filter Using LaBSE

Apart from the commonly used data cleansing methods, we also explore other techniques based on neural networks. LaBSE (Feng et al., 2020) is a multilingual BERT embedding model that can measure semantic similarities across languages. In our experiment, we notice that traditional data cleansing methods described in section 2.2.1 are unable to produce high-quality data, so we further filter the data using pre-training model LaBSE. For all parallel data, we calculated the similarity scores and

language pair	Raw data	Data Filtering	Filter Using LaBSE
en-zh	28.6M	14.7M	13.3M
en-ru	69.2M	45.1M	36.0M
ru-zh	33.4M	19.1M	14.7M

Table 1: Data sizes before and after filtering by different methods.

filtered out sentence pairs below a threshold. For Russian-Chinese data, the threshold is set to 0.7. For Russian-English and English-Chinese data, the threshold is set to 0.8. Our experiment integrates data denoising into the training process. The data size filtered by LaBSE is shown in table 1.

2.3 Multilingual Model

Johnson et al. (2017) proposes a simple solution that uses a single Neural Machine Translation (NMT) model to translate among multiple languages, and the model requires no change to the model architecture. Instead, the model introduces an artificial token at the beginning of the input sentence to specify the required target language. All languages use a shared vocabulary. There is no need to add more parameters. Surprisingly, experiments show that such model design can achieve better translation qualities across languages. In our experiment, we use two multilingual systems: forward model using ru2zh, en2zh, and ru2en data, and backward model using zh2ru and en2ru data.

2.4 Denoising Training

Wang et al. (2018) find that during training, dynamically adjusting noise data can boost system performance. The core idea is to train the model with noisy data at the initial stages and clearer data at later stages till the model converges. The quality of training data in this task is relatively poor as most of the data are crawled from website. We consider denoising training is suitable in this scenario. We simplify the denoising training process in our experiment, divide the training process into several stages.

For forward model, the training is divided into three steps: 1) Use all official provided data in three directions (ru2zh, en2zh, and ru2en) for training; 2) Use all clean data selected by LaBSE for incremental training; 3) Finally, use ru2zh clean data selected by LaBSE for incremental training.

For backward model, we only perform two steps: 1) Use all data (en2ru, zh2ru) for training; 2) Use zh2ru clean data selected by LaBSE for incremental training.

2.5 Fine-tuning and Ensemble

To achieve better results, fine-tuning with small-size in-domain data is necessary (Sun et al., 2019). An effective strategy for fine-tuning is to leverage the dev set available in this task. The fine-tuning strategies employed in our experiment include: 1) Add noise to the target side of the dev set to generate synthetic training data (Meng et al., 2020); 2) Use multiple models to generate synthetic data through beam search decoding, and then add synthetic data to the dev test for fine-tuning.

Model ensemble is also a widely used technique in previous WMT workshops (Garmash and Monz, 2016), which can boost the performance by combining the predictions of several models at each decoding step. We selected the best four models from the six we trained for ensemble.

3 Settings

3.1 Experiment Settings

We use the open-source fairseq (Ott et al., 2019) for training, and use sacreBLEU (Post, 2018) to measure system performances instead of the BLEU script mentioned in the task. The main parameters are as follows: Each model is trained using 8 GPUs. The size of each batch is set as 2048, parameter update frequency as 32, learning rate as 5e-4 (Vaswani et al., 2017) and label smoothing as 0.1 (Szegedy et al., 2016). The number of warmup steps is 4000, and the dropout is 0.1. We employ joint sentencepiece model (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation, with the size of the vocabulary set to 32k. Jieba tokenizer is used for Chinese word segmentation while Moses tokenizer for English and Russian word segmentation. The three languages share a vocabulary of 45K words. In the inference phase, we use the open-source marian (Junczys-Dowmunt et al., 2018) to perform decoding. The beam-size is 4 and the length penalty is set to 1.2.

System	BLEU
Data Filter	26.6
Multilingual model	29.3 (+2.7)
Full data denoise	30.0 (+0.7)
FTST + ru-zh denoise	31.9 (+1.9)
Ensemble	32.5 (+0.6)
2021 Final submit	27.7

Table 2: The experimental result of system

Training Strategy	Train Data	BLEU
Baseline	ru2zh	26.6
Enhanced target	+en2zh	28.7 (+2.1)
Enhanced target and source	+ru2en	29.3 (+0.6)
All Direction	+zh2ru zh2en en2ru	29.2 (-0.1)

Table 3: The experimental result of Multilingual Model

3.2 Training Process

We combine multi-stage denoising training with data augmentation methods. Figure 1 illustrates our training process:

1) We cleanse the training data using methods mentioned in 2.2.1 and train three forward models and one backward model.

2) We further denoise data using LaBSE (as mentioned in 2.2.3) and conduct denoising training until the model converge on the dev set.

3) We perform data augmentation as described in 2.2.2. We collect a total of 45M Russian monolingual data and split them into three sets, each with 15M sentences. We use three different forward models to generate three sets of training data. Hoping to add diversity to incremental training, we use the data synthesized by one model to train the other two models. For example, we use the synthetic data generated by forward model A to incremental train forward model B, C and so on. We also collect a total of 15M Chinese monolingual data and back translate the data using the backward model. We repeat back translation for three times and obtain three sets of back translation data. We incrementally train six models using the above synthetic data.

4) We average the last 5 checkpoints of each model and select the best four from the six models we trained for final ensemble.

4 Experiment Result

Our overall training strategy is to train a baseline model, conduct incremental training with techniques such as multilingual model, denoise training, data augmentation, and fine-tuning. Our submitted results come from ensembled models. Table 2 lists the results of our submission on dev set. Comparing with the baseline model, our final submission achieves an increase of 5.9 BLEU. Our baseline model is trained with data processed with methods mentioned in section 2.2.1. The BLEU score of the baseline model on the dev set is 26.6. Comparing with the baseline model, our multilingual strategy leads to a huge improvement of 2.7 BLEU. Our simplified denoising training strategy contributes to an increase of 0.7 BLEU. It should be noted that data augmentation techniques (FTST method and LaBSE denoising on ru2zh data) also result in a significant increase of 1.9 BLEU. Finally, an increase of 0.6 BLEU is gained via ensemble. Our submitted system gain 32.5 BLEU on the dev set, which demonstrate the effectiveness of our multiple strategies. According to the organizer's feedback, our submitted model gains 27.7 BLEU on the WMT21 test set.

5 Analysis

5.1 Multilingual Model and Model Performance

Our experiment results demonstrate that multilingual model has positive effects on system performance. We have experimented on different multilingual models and compare their results. Table 3 lists the results of different multilingual models. Compared with the baseline model, the multilingual model obtains 2.1 BLEU increase after adding en2zh data for training. A further 0.6 BLEU is achieved after adding the ru2en data, demonstrating that adding Russian data at the source side can optimize the encoder.

However, our experiment shows no improvement after adding data of other three directions. We adopt the enhanced target and source strategy for faster training, as training with all data might be considerably slow.

Training Strategy	BLEU
Baseline	26.6
+ru2zh denoise	28.0 (+1.4)
Enhanced target and source	29.3
+Full-data denoise	30.0 (+0.7)
+ru-zh denoise	30.5 (+0.5)

Table 4: The experimental result of denoising training

Training Strategy	BLEU
Enhanced target and source	29.3
Sampling BT	30.0
Beam BT	29.7
FT	29.7
Pivot FT	29.5
FTST	30.5

Table 5: The experimental result of data augmentation

5.2 Denoising Training and System Performance

Our experiment also demonstrates the contribution of denoising training to system performance. Table 4 compares the results of baseline and denoising training model, from which we can see an increase of 1.4 BLEU. We further compare the results measured at the three stages of denoising training. We use the enhanced target and source model to conduct simplified denoising training. Our experiment shows that full-data denoising training leads to an increase of 0.7 BLEU while ru2zh data denoising further leads to an increase of 0.5 BLEU. The experimental results show that the denoise strategy is effective and can lead to at least 1 BLEU improvement even after multilingual model enhancement.

5.3 Data Augmentation and System Performance

Data augmentation strategy also leads to huge BLEU improvements. We try multiple data augmentation strategies, including back translation (BT), forward translation (FT), FTST (2.2.2). Sampling BT means sampling from the model conditional distribution and beam BT means using beam search, when generating synthetic data. Table 5 shows the effects of different data enhancement methods. Our results show that sampling back translation can lead to better results (about 0.3 BLEU in our experiment). We also conduct two forward translation experiments: FT is translating Russian to Chinese directly, and Pivot FT is using English as the pivot language, which achieve only an undesirable result. We then using the FTST method and gain the best result with a BLEU score of 30.5. The experimental results show that the combination of sampling BT and FT data (FTST) can produce the best data augmentation effect.

6 Conclusion

This paper presents HW-TSC's submission to WMT21 Triangular Machine Translation Task. In general, we use Transformer architecture and explore multiple data filtering and selection methods. In terms of training and data processing strategies, multilingual model, denoising training, data augmentation, and FTST we used can effectively improve system performance. Our final result achieves an increase of 5.9 BLEU when comparing baseline model on the dev set and gain a BLEU score of 27.7 on the test which is the highest among all submissions.

References

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

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.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov.

2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, et al. 2020. Wechat neural machine translation systems for wmt20. *arXiv preprint arXiv:2010.00247*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

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

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

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.

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.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. *arXiv preprint arXiv:1809.00068*.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

DUTNLP Machine Translation System for WMT21 Triangular Translation Task

Huan Liu **Junpeng Liu** **Kaiyu Huang*** **Degen Huang**
School of Computer Science, Dalian University of Technology
{liuhuan4221,liujunpeng_nlp, kaiyuhuang}@mail.dlut.edu.cn
{huangdg}@dlut.edu.cn

Abstract

This paper describes DUT-NLP Lab's submission to the WMT-21 triangular machine translation shared task. The participants are not allowed to use other data and the translation direction of this task is Russian-to-Chinese. In this task, we use the Transformer as our baseline model, and integrate several techniques to enhance the performance of the baseline, including data filtering, data selection, fine-tuning, and post-editing. Further, to make use of the English resources, such as Russian/English and Chinese/English parallel data, the relationship triangle is constructed by multilingual neural machine translation systems. As a result, our submission achieves a BLEU score of 21.9 in Russian-to-Chinese.

1 Introduction

The WMT2021 Shared Task on translating sentences from Russian into Chinese provides a challenging mixed-genre test for machine translation systems and a triangular relationship for researchers to evaluate new techniques. The task focuses on translation between non-English languages and optimally mixing direct and indirect parallel resources. In this task, the participants must use only the provided parallel training data and use of other data is not allowed. The provided data is shown in Table 1, including parallel data in three directions. Given the language pair (Russian-to-Chinese), the bulk of previous NMT work has pursued one of two strategies that are illustrated in Figure 1:

Direct: Collect parallel Russian-to-Chinese data from the public resource, and train a Russian-to-Chinese translator.

Pivot: Collect parallel Russian-to-English and English-to-Chinese data (usually larger than direct data), train two translators (Russian-to-English + English-to-Chinese), and make a cascade translator from Russian to Chinese.

Direction	# SENT
Russian/Chinese	33,388,455
Russian/English	69,155,404
English/Chinese	28,528,290

Table 1: The provided training data in the constrained data track.

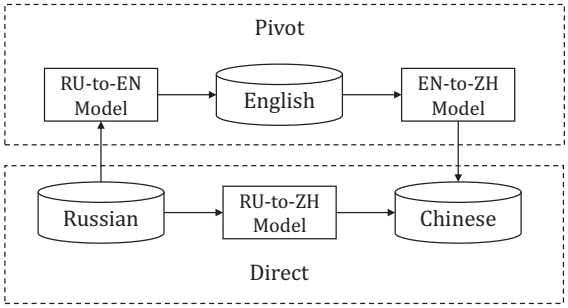


Figure 1: The illustration of two strategies for triangular machine translation.

The DUTNLP submission to the constrained data track is based on the mainstream architecture Transformer (Vaswani et al., 2017). According to the scale of datasets, this shared task should be considered as the high-resource translation direction. We use the Transformer-big setting for better performance, on the contrary, a low-resource translation task often utilizes the Transformer-base due to the limited parallel training data. Moreover, to enhance the baseline model and investigate the usage of triangular direction data, we utilize two pipelines for combining English related resources: 1) incorporates English-to-Chinese translator into direct translation process to normalize the translation of rare words. 2) adopts a multilingual training strategy to make use of English ↔ X parallel resources. Besides, some of the provided Russian ↔ Chinese parallel corpora are crawled from the

*Corresponding author

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 331–335
November 10–11, 2021.

web, which has many noise issues. We filter the training bilingual corpora with several techniques, including language model and constrained rules.

This paper is structured as follows: Section 2 describes variants of models we used in the shared task. In Section 3, we introduce the system overview using several techniques for model enhancement, including data pre-processing and filtering, triangular translation strategy and fine-tuning. In Section 4, this paper presents experimental settings, main results and analysis. Finally, in Section 5 we draw a brief conclusion of our work in the WMT2021 Triangular Translation Task.

2 Model

2.1 Transformer

Recent advances in Transformer (Vaswani et al., 2017) have led to significant improvement of Neural Machine Translation (NMT) and achieve human parity on Automatic Chinese to English News Translation (Hassan et al., 2018). The Transformer adopts a sequence-to-sequence structure, using stacked encoder and decoder layers of self-attention. Each encoder layer consists of a self-attention mechanism and a feed-forward network. Each decoder layer consists of a masked self-attention layer, a cross self-attention layer, and a feed-forward network layer. Moreover, the Transformer leverages positional embedding, residual connections and layer normalization for enhancement (Ba et al., 2016). In this paper, we adopt the Transformer-big as the baseline model, in which both the encoder and decoder have 6 layers, the hidden size is 1024, and the feed-forward inner size is 4096.

2.2 Multilingual Architecture

Multilingual neural machine translation (mNMT) handles the translation between multiple languages by joint training in a multi-task setup (Johnson et al., 2017), which greatly eases the model deployment. Previous works (Lakew et al., 2018; Tan et al., 2019) show that the mNMT model can facilitate cross-lingual knowledge transfer between languages. It also enables zero-shot translation between unseen language pairs (Johnson et al., 2017; Al-Shedivat and Parikh, 2019; Zhang et al., 2020). Following Johnson et al. (2017), we build our multilingual translation system based on the advanced Transformer model by adding a pretending language token to each source sentence, which indicates the language to be translated into. And the multilingual model is also fine-tuned on pre-trained mNMT models such as mBART(Liu et al., 2020) and mRASP(Lin et al., 2020).

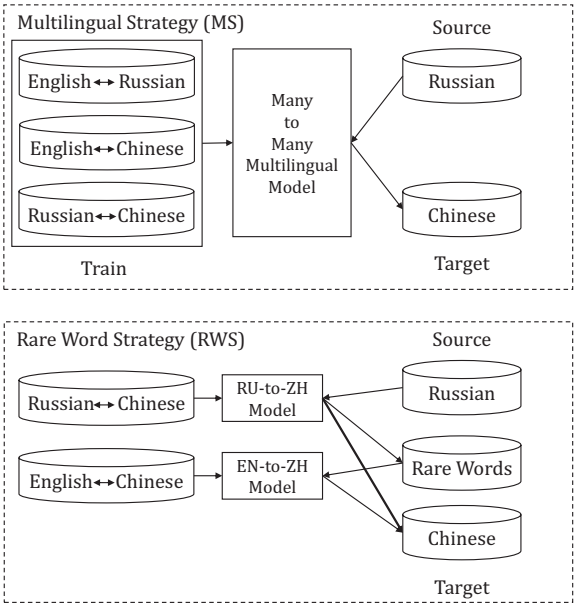


Figure 2: The illustration of two strategies for incorporating English resources into the baseline systems.

3 System Overview

3.1 Data Pre-processing and Filtering

To improve the quality of data, especially the Russian ↔ Chinese parallel data, we filter noisy data with several techniques. The flow of all training data pre-processing and filtering is set to step by step as follows:

- Punctuation normalization with Moses scripts (Koehn et al., 2007) for all language pairs.
- Chinese word segmentation using the open segmentation tool (Huang et al., 2020). Splitting the English and Russian words using clearly delimiter by Moses "tokenizer" script.
- True-casing. The uppercase letter may influence the generation of vocabulary dictionaries. We transfer the uppercase letter into lower case automatically by Moses scripts.
- Filtering out the sentence pairs longer than 256 or duplicated translation.
- Filtering out the sentences by the multilingual parallel data filter tools LASER [1].

[1]https://github.com/facebookresearch/LASER

- Filtering out the sentence according to their characteristics in terms of language identification and length ratios, in particular, the sentence pairs whose length ratio between the source and target are not in range of 1:2.5 and 2.5:1 are abandoned.

- The bilingual direct translator utilizes BPE to encode text into sub-word unit (Sennrich et al., 2016). In the multilingual translator, the system applies sub-word processing using SentencePiece tool (Kudo and Richardson, 2018).

3.2 Triangular Translation Strategy

In this task, we exploit two strategies to incorporate English resources into the baseline systems, which are shown in Figure 2.

Rare Word Strategy According to the official provided data, the performance of the direct translator is better than the pivot. And the provided parallel Russian ↔ Chinese training data can be considered as the high-resource. The baseline model can be trained successfully with the bilingual training data and can achieve competitive performance in most cases. However, the translations of rare words are always terrible by the direct translator, for example, most Russian names will be translated into English. We utilize an English ↔ Chinese translator to alleviate this issue. The external translator only works in the specific case to reduce the error propagation and redundant computational cost. This strategy is to complement the direct baseline model and improve the performance in an interpretable way.

Multilingual Strategy The mNMT models are effective in low resource settings due to knowledge transfer. To make use of all provided parallel data, we train a multilingual many-to-many models with 3 language pairs(i.e., 6 directed translation direction). However, it is expensive to train different parameter sets to get the best translation result in the target direction. According to Lin et al. (2020), mRASP can obtain more improvements with rich-resource language pairs than multilingual many-to-many models, so multilingual strategy (MS) is based on fine-tuning large-scale pre-training models. We fine-tune on two pre-trained models respectively: mBART(Liu et al., 2020) and mRASP(Lin et al., 2020). In particular, we use MS-mBART and MS-mRASP for the two methods in section 4. We use mBART to continue training on the filtered parallel data subset, and select the appropriate checkpoints according to the performance on the validation set. The mRASP model pre-trained on a dataset contains 32 English-centric language pairs, including English-Russian and English-Chinese, and Russian-Chinese are not the direct training objective of it. We stop the fine-tune process when the loss on validation does not decrease for 5 consecutive steps (measured every 50 updates). The experiment shows that fine-tuning on mRASP with filtered parallel data achieves anticipated improvements.

Models	VALID.	TEST.
Direct	20.2	17.0
Baseline	24.92	20.4
+RWS	26.40	21.8
+MS-mBART	19.47	-
+MS-mRASP	25.21	21.7

Table 2: The BLEU-4 scores in the constrained data track.

4 Experiment

4.1 Experimental Settings

The implementation of our models is based on Fairseq (Ott et al., 2019). All the models are carried out on 2 NVIDIA 3090 GPUs each of which has 24 GB of memory. The parameters of Transformer-Big, mBART, and mRASP are all followed by the architectures themselves. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The batch size is set to 4096 tokens and the "update-freq" parameter in Fairseq is set to 2. In particular, for pre-trained language models settings (i.e., mBART and mRASP), the batch size is 2048 and "update-freq" is 4. The initial learning rate is set to $5e^{-4}$ for training and $3e^{-5}$ for fine-tuning. The learning scheduler is inverse_sqrt and all the dropout probabilities are set to 0.1. We select the checkpoint with the average of the top 5 sacreBLEU scores on the development set as the final checkpoint in each training. We calculate the BLEU-4 score for all experiments, which is officially recommended.

4.2 Main Results

Table 2 shows the Triangular translation results on validation and test set. We train multiple models in each setting and report the best scores in Table

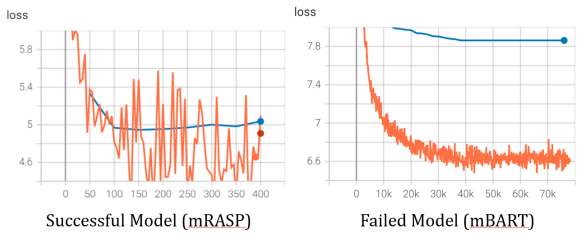


Figure 3: Loss curve of the multilingual models and the local minima.

Ref: 按照历史学家的说法，**阿尔巴津圣母圣像**是1492年尼科季姆修士所绘。
Baseline: 根据历史学家的推测，Albazinskaya是在1492年由尼克科姆修道院写的。
RWS: 根据历史学家的推测，**阿尔巴津圣母像**是在1492年由尼科季姆修道院写的。
MS-mRASP: 根据历史的推测，**阿尔巴津圣母神像**是在1492年由尼古迪姆修道院写成的。
Ref: **坦波夫地区**是农业地区，但该地区在种子生产领域也面临严重问题。
Baseline: Tambovshchyna是一个农业地区，但在种子生产方面也面临严重问题。
RWS: **坦波夫地区**是一个农业地区，但在种子生产方面也面临严重问题。
MS-mRASP: **坦布夫地区**是一个农业地区，但面临着严重的农业问题。

Figure 4: The results by different NMT systems. The rare words are bold.

2. In particular, the direct model is trained with all available data, and the Baseline utilizes the clean parallel data after filtering to train. It improves the direct method by 4.72 and 3.4 BLEU scores on validation and test set, respectively. Moreover, to make use of the English resources, we propose two triangular training strategies to investigate the effect of the triangle relationship. The rare word strategy (RWS) can effectively improve the baseline from 24.92 to 26.40 in terms of BLEU scores. And the multilingual strategy (MS) improves the baseline from 24.92 to 25.21 in terms of BLEU scores.

4.3 Triangular Translation Analysis

The official results do not open the reference on the test set. So we investigate the experiments further on the validation set.

Multilingual Models In this task, we utilize the multilingual training strategy which is fine-tuned on mBART and mRASP. It is surprising to find that the model trains failed on this dataset, which is fine-tuned on mBART, shown in Figure 3. However, the mRASP worked in this scenario. Follow by Lin et al. (2020), the mRASP is beneficial to fine-tune the high-resource language pairs. And this method (MS-mRASP) achieves the best BLEU-4 scores on the validation set. Although the training curve of MS-mBART is more stable, the overall loss has remained relatively high level and cannot be effectively declined.

Results Discussion Figure 4 shows the results from different models. The two multilingual translation strategies can effectively improve the performance of baseline model, especially for the rare words. In the baseline model, some rare words are translated into English, and most of the words are translated correctly depends on the direct training method. It is worth mentioning the two multilingual translation strategies can alleviate this issue effectively.

5 Conclusion

This paper presents the DUTNLP Translation systems for WMT2021 Russian-to-Chinese triangular translation tasks. We investigate various neural architectures and data filtering to build strong baseline systems. Then the two triangular translation strategies are used to improve the baselines. We also prove that in-domain finetuning is very effective for this translation task. Finally, we discuss the results carefully and analyze the influence of different triangular strategies for further improvement. A number of advanced technologies reported in this paper focus on alleviating the issue of triangle translation. As a result, our system outperforms the strong baseline by 1.48 and 1.4 BLEU scores on the validation set and test set, respectively. In the future, we will investigate the technologies in the low-resource scenario and continue to improve the performance of this task through post-evaluation submissions.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and suggestions to improve the quality of the paper. The authors gratefully acknowledge the financial support provided by the National Key Research and Development Program of China (2020AAA0108004) and the National Natural Science Foundation of China under(No.U1936109).

References

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*,

pages 1184–1197, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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*.

Kaiyu Huang, Degen Huang, Zhuang Liu, and Fengran Mo. 2020. A joint multiple criteria model in transfer learning for cross-domain chinese word segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3873–3882.

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 association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

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.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. 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.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Pivot based Transfer Learning for Neural Machine Translation: CFILT IITB @ WMT 2021 Triangular MT

Shivam Mhaskar, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
{shivammhaskar, pb}@cse.iitb.ac.in

Abstract

In this paper, we discuss the various techniques that we used to implement the Russian-Chinese machine translation system for the Triangular MT task at WMT 2021. Neural Machine translation systems based on transformer architecture have an encoder-decoder architecture, which are trained end-to-end and require a large amount of parallel corpus to produce good quality translations. This is the reason why neural machine translation systems are referred to as *data hungry*. Such a large amount of parallel corpus is majorly available for language pairs which include English and not for non-English language pairs. This is a major problem in building neural machine translation systems for non-English language pairs. We try to utilize the resources of the English language to improve the translation of non-English language pairs. We use the pivot language, that is English, to leverage transfer learning to improve the quality of Russian-Chinese translation. Compared to the baseline transformer-based neural machine translation system, we observe that the pivot language-based transfer learning technique gives a higher BLEU score.

1 Introduction

The aim of this work is to improve the quality of Machine Translation (MT) for low-resource, distant and non-English language pairs. One of the major requirements for the good performance of the Neural Machine Translation (NMT) systems is the availability of a large parallel corpus. Such large parallel corpus of good quality is not available for low-resource, distant and non-English language pairs but mostly available for language pairs containing English. This poses a major challenge in developing good quality Machine Translation systems for non-English and distant language pairs. As a result there is a need to come up with additional resources by augmenting parallel corpora or by using knowledge from other tasks using transfer learning for translation of non-English language pairs. In this paper, we focus on leveraging the knowledge from other tasks using transfer learning to improve the performance of NMT systems for low resource language pairs.

In our pivot based transfer learning experiments we try to utilize the resources of English language, that is English-Chinese and English-Russian parallel corpora to improve the quality of Russian-Chinese translation. We implement techniques which efficiently use the resources of the English language for the task of Russian-Chinese translation.

2 Related Work

Recurrent Neural Network (RNN) based encoder decoder architectures (Bahdanau et al., 2014; Cho et al., 2014; Sutskever et al., 2014) were initially used in NMT systems. Transformer (Vaswani et al., 2017) architecture improved the performance of NMT systems. In order to enable translation between distant and non-English language pairs for which a large amount of parallel corpus is not available, a cascade method can be used. In the cascade method, two models are trained, a source language to English and a English to target language model. Then to translate a source sentence to target sentence, the source sentence is passed through the two models. (Zoph et al., 2016) introduced a transfer learning technique in which a parent model is trained on high resource language pairs, which is then used to initialize the the parameters of a child model which is then trained on low resource language pair data. (Kim et al., 2019) introduced pivot language-based transfer learning techniques in which the encoder and decoder of the model for low resource language pair is initialized using the encoder and decoder of different models trained on high resource language pairs, and this model is then finetuned on low resource language

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 336–340

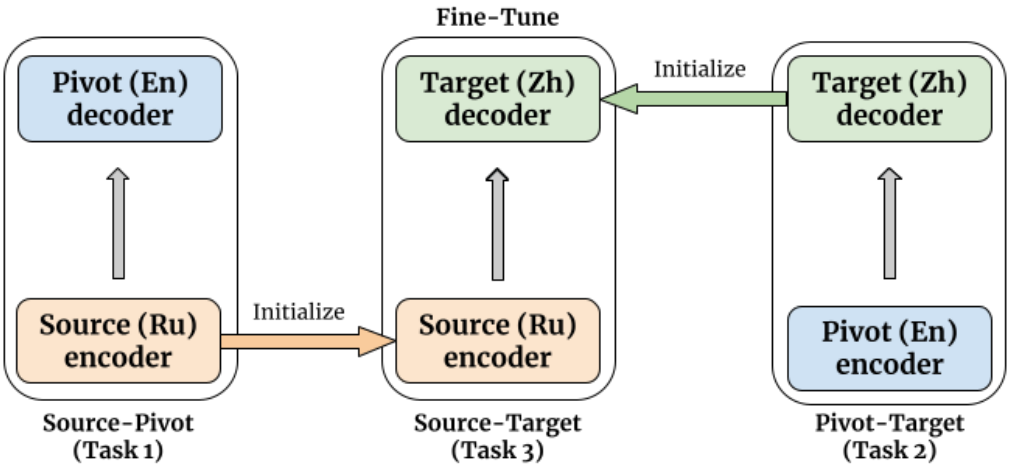


Figure 1: Direct Pivoting (En:English, Ru:Russian, Zh:Chinese)

pair data. Multi-lingual NMT systems (Zoph and Knight, 2016; Firat et al., 2016; Johnson et al., 2017) can also be used to improve the performance of low resource language pair translation as knowledge is transferred from various languages which helps the task of low resource language pair translation.

3 Approaches

In this section, we discuss the various approaches we used to build a Russian-Chinese MT system. We mainly focus on pivot-based transfer learning techniques, in which we use the resources of English to improve the quality of Russian-Chinese translation.

3.1 Baseline

The baseline Russian-Chinese model is a NMT model based on Transformer architecture. The model is trained on Russian-Chinese parallel data.

3.2 Cascade Model

The cascade model makes use of the resources of English language to train a Russian-Chinese MT system. In this approach, we train two NMT models, a source to pivot (Russian-English) model and a pivot to target (English-Chinese) model. The source Russian sentence is first translated into English using the Russian-English model. Then this English sentence is translated into Chinese using the English-Chinese model. In this way, the cascade model translates the Russian sentence to Chinese by passing it through the two NMT models. There are a few disadvantages in this cascade model based approach,

1. The source sentence is passed through two different NMT models to produce the target sentence. This doubles the decoding time for the generation of the output sentence which is very inefficient.

2. The errors in translation are propagated from first (source-pivot) model to the second (pivot-target) model.

These disadvantages of the cascade model approach make it an undesirable approach to utilize the resources of the pivot language. In order to overcome these disadvantages, we need to train a single source-target model which utilizes the resources of the pivot language. In the following pivot language-based transfer learning technique, direct pivoting, we overcome these disadvantages. In this technique, we train a single source-target model while utilizing the resources of the pivot language.

3.3 Direct Pivoting

In this technique, we first train two separate NMT models, a source-pivot model and a pivot-target model. As demonstrated in Figure 1, we first separately train a Russian-English (source-pivot) model (task 1) and a English-Chinese (pivot-target) model (task 2) on their respective parallel corpus. Then we use the encoder of the Russian-English (source-pivot) model and the decoder of

the English-Chinese (pivot-target) model to initialize the encoder and decoder of the Russian-Chinese (source-target) model respectively. Finally, we fine-tune the Russian-Chinese (source-target) model on the Russian-Chinese parallel corpus.

As in this technique we are training a single source-target (Russian-Chinese) model, there is no problem of double decoding time. The parameters of the encoder and decoder of the source-target (Russian-Chinese) model are not randomly initialized, they are trained on the source-pivot and pivot-target translation task respectively. The initialized encoder and decoder of the source-target (Russian-Chinese) model have already learned some representation or knowledge from the previous tasks. This knowledge helps in the source-target (Russian-Chinese) translation task. In this way this approach utilizes the resources of the pivot (English) language which assists in the translation task from source to target (Russian-to-Chinese).

4 Experiments

In this section, we discuss the details of all the experiments that we carried out to implement the Russian-Chinese MT system.

4.1 Dataset

The NMT systems were trained on the parallel corpora provided by the WMT 2021 organizers. We used the Russian-Chinese, Russian-English, and the Chinese-English parallel corpus. We used a subset of the provided parallel corpora for training the models. Byte Pair Encoding (BPE) (Sennrich et al., 2015) is used as a segmentation technique. The words in the data are broken down into sub-words using the BPE technique. For the baseline model the number of BPE merge operations used were 16000 for the source and target data. For the direct pivoting model, the source and target vocabulary are combined English-Russian and English-Chinese vocabulary, respectively. So, the BPE codes are computed by combining the source side Russian and English data for source and the target side English and Chinese data for target. The number of BPE merge operations used were 32000 for the source and target data. The detailed corpora statistics are mentioned in Table 1.

Language pair	Number of sentences
Russian-Chinese	10M
Russian-English	10M
English-Chinese	10M

Table 1: Corpora statistics of all the language pairs

4.2 Models

For all the experiments, Transformer architecture was used. The encoder of the Transformer consisted of 6 encoder layers and 8 encoder attention heads. The encoder used embeddings of dimension 512. The decoder of the Transformer consisted of 6 decoder layers and 8 decoder attention heads. For the implementation of all models, fairseq (Ott et al., 2019) library was used.

4.3 Training Setup

For all experiments, the transformer model from fairseq library was used. The optimizer used was adam with betas (0.9, 0.98). The inverse square root learning rate scheduler was used with an initial learning rate of 5e-4 and 4000 warm-up updates. The criterion used was label smoothed cross entropy with label smoothing of 0.1. The dropout probability value used was 0.3 for all layers. For the baseline model, the size of source (Russian) and target (Chinese) vocabulary is 16876 and 29500, respectively. For the direct pivoting model, the size of source (combined Russian-English) and target (combined English-Chinese) vocabulary is 34020 and 47052, respectively. The best model for all the techniques was chosen by calculating the BLEU (Papineni et al., 2002) scores on the development set provided by the WMT 2021 organizers and the choosing the model with best BLEU score.

4.4 Baseline

The baseline model is a transformer model trained on Russian-Chinese (source-target) parallel corpus.

4.5 Cascade Model

The cascade model consists of two NMT models trained separately. The first model is a Russian-English model trained on Russian-English parallel corpus. The second model is a English-Chinese model trained on English-Chinese parallel corpus. For translating a Russian sentence to Chinese, the sentence is passed through two models.

4.6 Direct Pivoting

The direct pivoting model uses a shared vocabulary of Russian-English (source-pivot) on the encoder side and English-Chinese (pivot-target) on the decoder side. This is done to ensure that the

Model	BLEU score
Baseline	18.2
Cascade	17.2
Direct Pivoting	18.8

Table 2: BLEU scores of Russian-Chinese NMT system using different techniques

encoder and decoder parameters are transferable as transformers are fixed vocabulary models. The Russian-English (source-pivot) model is trained on Russian-English parallel data and the English-Chinese (pivot-target) model is trained on English-Chinese parallel data. Then the encoder of Russian-English model and decoder of English-Chinese model is used to initialize the encoder and decoder of Russian-Chinese (source-target) model. Finally, we fine-tune the Russian-Chinese model on the Russian-Chinese parallel data.

5 Results and Analysis

The evaluation of the models were performed on the basis of the BLEU scores. These BLEU scores were calculated and provided by the WMT 2021 organizers. The BLEU scores were calculated on a test set provided by WMT 2021 organizers, which consisted of 1751 sentences. Table 2 shows the BLEU scores of all the models. The baseline Russian-Chinese model produced a BLEU score of 18.2. The cascade model in which the Russian sentence is first translated to English using Russian-English model and then the English sentence is translated to Chinese using the English-Chinese model, produced a BLEU score of 17.2. The possible reason for this decrease in BLEU score is that the errors made by the Russian-English model are propagated to the English-Chinese model, which further introduced its own errors. As the source sentence is passed through the two model each model introduces its own errors, which decreases the BLEU score.

The direct pivoting model produced a BLEU score of 18.8 which improved the BLEU score by 0.6 points over the baseline model. This increase in BLEU score is because the encoder and decoder of the Russian-Chinese model are not randomly initialized; but they are initialized from the encoder and decoder of Russian-English and English-Chinese model respectively. Then the model is fine-tuned on Russian-Chinese parallel corpus. The encoder and decoder have already learnt some representations which helps in the task of Russian-Chinese translation. Also as this is a single NMT model, there is no problem of propagation of errors or double decoding time.

6 Conclusion and Future Work

In this work, we implement and compared pivot language-based transfer learning technique to improve the task of translation between non-English language pair, that is Russian-Chinese. We observe that pivot language-based transfer learning technique improves the BLEU score over the baseline model and is an efficient way to use the resources of the pivot language. We also observe that the pivot language-based transfer learning technique mitigates the problems of double decoding time and error propagation present in simple cascade-based models.

In future, we plan to explore various data augmentation techniques that can make use of the resources of the English language to augment data for the task of translation of non-English language pair translation. We also plan to use various language model pretraining techniques like Masked Sequence to Sequence Pre-training (MASS) to pretrain the encoder and decoder before using them for the downstream task of translation.

References

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. *CoRR*, abs/1409.1259.

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.

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.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. *arXiv preprint arXiv:1909.09524*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

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.

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. *CoRR*, abs/1409.3215.

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.

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.

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.

Papago's Submissions to the WMT21 triangular Translation Task

Jeonghyeok Park*
Papago, Naver Corp.
1170339900l1@sjtu.edu.cn

Hyunjoong Kim
Papago, Naver Corp.
soy.lovit@navercorp.com

Hyunchang Cho
Papago, Naver Corp.
hyunchang.cho@navercorp.com

Abstract

This paper describes Naver Papago's submission to the WMT21 shared triangular MT task to enhance the non-English MT system with tri-language parallel data. The provided parallel data are Russian-Chinese (direct), Russian-English (indirect), and English-Chinese (indirect) data. This task aims to improve the quality of the Russian-to-Chinese MT system by exploiting the direct and indirect parallel resources. The direct parallel data is noisy data crawled from the web. To alleviate the issue, we conduct extensive experiments to find effective data filtering methods. With the empirical knowledge that the performance of bilingual MT is better than multi-lingual MT and related experiment results, we approach this task as bilingual MT, where the two indirect data are transformed to direct data. In addition, we use the Transformer, a robust translation model, as our baseline and integrate several techniques, averaging checkpoints, model ensemble, and re-ranking. Our final system provides a 12.7 BLEU points improvement over a baseline system on the WMT21 triangular MT development set. In the official evaluation of the test set, ours is ranked 2nd in terms of BLEU scores.

1 Introduction

We participate in the WMT21 triangular machine translation task, using the direct and indirect parallel data to improve Russian-to-Chinese machine translation. The provided data consists of one noisy web corpus (Russian-Chinese, direct translation) and two combined bitexts from several public resources (English-Chinese/Russian, indirect). Such cases frequently occur in both actual translation services and research. In particular, this task is crucial in scenarios where we need to improve the performance of non-English translations or low-resource languages with high-resource parallel data.

Previous works deal with the triangular MT using several methods such as pivot-translation (Cheng et al., 2017), transfer learning (Kim et al., 2019), pre-trained multi-lingual MT (Liu et al., 2020; Tang et al., 2020) and so on.

In this paper, we explore existing novel techniques to integrate them for the triangular MT tasks. The original direct parallel degrades the translation quality of the model due to the noisy parts containing not well-aligned sentence pairs, erroneous characters, or the wrong language ID. To discard the noise parts of the noisy web corpus, we filter out data with sequence length, length ratio, language ID, de-duplication, and the sentence similarity computed with pre-trained multi-lingual language model (LaBSE, Feng et al., 2020). In preliminary experiments, we approached this task in three main ways: bilingual MT, multi-lingual MT, and fine-tuning the pre-trained multi-lingual translation model (i.e., mBART). As shown in Section 4, we found that the bilingual MT outperforms the others. Thus, to augment the Russian-to-Chinese corpus, we conduct two types of data augmentation: (1) back-translation on the discarded monolingual Chinese data from noise-refining steps and (2) translation using English as pivot language on two indirect bilingual data (e.g., feed English of English-Chinese data to English-to-Russian translation model to augment Russian-Chinese data). In detail, we generate the synthetic data by using different decoding methods such as beam search, sampling, and adding noise to beam search outputs. Our submission systems use 12-layer Transformer architecture. Furthermore, we exploit ensemble, averaging checkpoints, and noisy-channel re-ranking techniques to mitigate the over-fitting problem or improve the generalization capability in the test set.

To find suitable methods for triangular MT, we conduct extensive experiments, where all neural machine translation (NMT) systems are evaluated against the development set released in the WMT21

* Work done during internship at Naver Corp.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 341–346

triangular MT shared task. Our final submission improves about 12.7 and 8.9 BLEU points compared to the organizer's baseline system on the development set and test set, respectively.

2 Approaches

2.1 Data Pre-processing

On all three corpora, we apply data normalization such as unifying punctuation marks and parentheses. For Chinese, we convert the traditional Chinese to Simplified Chinese using the open-source toolkit HanziConv[1] For all languages, we apply a language-specific tokenizer as a pre-tokenization step. We use NLTK[2] for English and Russian, jieba[3] for Chinese. And then, we apply joint multilingual Byte-Pair Encoding (BPE, Sennrich et al. 2016) to the pre-tokenized corpus with 75K merge-operations and 10K character limitation using the open-source toolkit Transformers[4].

2.2 Data Filtering

The provided parallel corpus contains a certain amount of noisy parts, which affects the translation quality. Thus, we eliminate noisy parts with the following heuristics rules:

- *Filtering out sentence pairs containing more than 256 tokens.
- *Filtering out sentence pairs consisting of characters of other languages than a predefined threshold. For this sake, we use an in-house language detector. We determine the threshold experimentally.
- *Filtering out sentence pairs with source/target length ratio exceeding 1.5 (Ott et al., 2019).
- Filtering out duplication in corpora (Khayrallah and Koehn, 2018; Ott et al., 2019). There are 4 options as follows: filtering out (1) duplicate sentence pairs (It is called Pair-dedup. in Table 3); (2) duplicate source sentences (Src-only-dedup.); (3) duplicate target sentences (Tgt-only-dedup.); (4) duplicate source and duplicate target sentences (Src&Tgt-dedup.).

[1]https://github.com/berniey/hanziconv
[2]https://www.nltk.org
[3]https://github.com/fxsjy/jieba
[4]https://github.com/huggingface/transformers

Systems	RU-ZH	EN-ZH	RU-EN
Original	33M	28M	69M
+ Basic Filter	22M	19M	50M
+ De-duplicate	18M	15M	42M
+ LaBSE Filter	13M	12.7M	39.3M

Table 1: The amount of the sentence pairs

- Filtering out sentence pairs according to the cosine similarity of the sentence pair. To this end, we feed the sentence pair to LaBSE (Feng et al., 2020) and calculate the cosine similarity score of the sentence pair. Then, we discard sentence pairs whose cosine similarity score falls below a certain threshold. From here on, it is called LaBSE filtering,

where the filtering methods marked with * are basic filtering methods.

We conducted experiments on the de-duplication and LaBSE filtering to find an optimal combination of them in subsection 4.1. Based on the results, we remove the duplicate sentence pairs and set the threshold of LaBSE filtering to 0.5 in our experiment. Table 1 shows the amount of the sentence pairs after filtering.

2.3 Data Augmentation

To augment the direct bilingual data (Russian-to-Chinese), we generate synthetic bilingual sentence pairs on three data: one monolingual data (Chinese), two indirect parallel data (English-Chinese, and Russian-English). The Chinese monolingual corpora filtered out in the filtering step are translated back to Russian by the Chinese-to-Russian translation model (back-translation). To utilize indirect parallel data, we first train English-to-Chinese and English-to-Russian translation systems using provided corpora. Then we acquire synthetic Russian-Chinese pairs translating English sentences of English-Chinese data to Russian sentences using the trained English-to-Russian MT system (back-translated synthetic corpus). In the same way as before, we also acquire synthetic Russian-Chinese pairs translating English sentences of Russian-English data to Chinese using the English-to-Chinese MT system (forward-translated synthetic corpus). In this paper, we use the Transformer-Big model to augment the direct bilingual. In the future, we would thoroughly ex-

plore several methods to improve further the quality of augmented data, such as using a bigger model and iterative back-translation Hoang et al. (2018).

Following Edunov et al. (2018), we use various decoding strategies , including maximum a-posteriori (MAP) and non-MAP methods for effective data augmentation. In detail, there are five decoding methods: (1) beam search; (2) sampling; (3) sampling top 10; (4) noising beam outputs; (5) noising beam outputs only with sentences longer than 5. The sampling top 10 method is to restrict the sampling method to the k highest-scoring outputs at every decoding step. The noising beam outputs method denotes to add three types of noise such as random permutation over tokens, deleting some tokens, and masking some tokens. Edunov et al. (2018) demonstrated that the non-MAP decoding methods such as (2)-(5) outperform pure beam search.

In our experiments, we generate the five synthetic bilingual data using the different decoding methods. We train five models with each synthetic data embracing data variation. Performance of each way is described in Table 4 In the final submission, we choose a combination of (1) beam search, (2) sampling, and (3) restricted noising beam outputs experimentally. After generating the synthetic bilingual data, we apply the data filtering schemes described in section 2.2 to them. We upsample bitext data to maintain a 1-to-1 ratio of real to synthetic bitext during the training phase.

2.4 Model

In our experiments, we adopt three Transformer architectures.

- **Transformer-Base** with a 6-layers encoder-decoder and a model dimension of 512 as used in Vaswani et al. (2017).
- **Transformer-Big** with a 6-layers encoder-decoder and a model dimension of 1024 as used in Vaswani et al. (2017).
- **Transformer-Large** is similar to Transformer-Big model except that it uses a 12-layers encoder-decoder with pre-norm (Wang et al., 2019).

To boost the performance of the translation model, we average the parameters acquired from various epochs obtained in a training phase and then ensemble the averaged checkpoints involving various variations in terms of data. Moreover, we perform a grid search for decoding hyper-parameters such as length penalty and beam size to find the best performance. We conduct preliminary experiments using the Transformer-Big model to find (sub)optimal configurations in data filtering, data augmentation, hyper-parameters, and so on. Then, based on the observations, we apply the (sub)optimal configurations to the Transformer-Large model.

2.5 Noisy-Channel Re-ranking

The noisy channel re-ranking (Yee et al., 2019) applies Bayes' rule to decoding:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}, \tag{1}$$

where x is source sequence and y is hypothesis sequence in translation task. Since $p(x)$ is constant for all y, re-ranking score for each hypothesis candidate can be reconstruct as follows:

$$\frac{\lambda_1 \log p(y|x) + \lambda_2 \log p(x|y) + \lambda_3 \log p(y)}{|y|^\alpha}, \tag{2}$$

where λ, α are tunable weights, $|y|$ is length of hypothesis sequence, and $p(y|x)$, $p(x|y)$, $p(y)$ denote score of forward model, backward model, and language model, respectively.

In a preliminary experiment, we used several publicly released Chinese language models [5] and found that they caused performance degradation. In addition, we used another scoring metric (inverse document frequency similar to BARTScore (Yuan et al., 2021)) when re-ranking, but this did not give any performance gain. Due to time and resource constraints, we could not fully explore our own Chinese language model.

3 Experiments and Results

3.1 Experiment Setup

Our base system is based on the Transformer-Large with an embedding size of 1024, 12 encoder and decoder layers, 12 attention heads, shared source and target embedding, the sinusoidal positional embedding, and pre-norm. We train with a batch size of 3584 tokens and optimize the model parameters using Adam optimizer with a learning rate 1e-3 β_1= 0.9 and β_2 = 0.98, learning rate warm-up

[5] https://huggingface.co

Systems	#Sentence	BLEU
Organizer's Systems		
Direct	33M	20.2
Pivot	(69+28)M	19.7
Our Systems		
Transformer-Base	22M*	26.4
Transformer-Big	22M*	28.7
Transformer-Large	22M*	28.9
+ De-duplication	18M	29.2
+ LaBSE Filter (0.5)	13M	29.8
+ Augmented data	(13+45)M	31.3
+ Averaging	-	31.9
+ Ensemble	-	33.0
+ Re-ranking**	-	33.0

Table 2: Performances on the WMT21 triangular MT Russian-to-Chinese development set in Transformer-Large. The asterisk(*) mark is a basic filter, and the double-asterisk(**) denotes our submitted system.

Systems	#Sentence	BLEU
Basic filter	22M	28.7
LaBSE filter (0.5)	17M	29.5 (+0.8)
Pair-dedup.	18M	28.7 (+0.0)
+ LaBSE filter	13M	29.7 (+0.9)
Src-only-dedup.	12.6M	28.5 (-0.3)
Tgt-only-dedup.	13M	28.5 (-0.2)
Src&Tgt-dedup.	11M	28.9 (+0.1)
+ LaBSE filter	10.6M	29.8 (+1.0)

Table 3: Further experiment results with different data filtering methods in Transformer-Big. The Basic filter contains filtering the sentences by sequence length, language ratio, and length ratio. The others are described in section 2.2. The plus marks denote that the filtering method is applied additionally. For example, the "+ LaBSE filter" in fifth row means that both Pair-dedup and LaBSE filter are applied.

over the first 16k steps. Additionally, we apply label smoothing with a factor of 0.1. In the training phase, the dropout is set to 0.1, and the attention dropout is set to 0.3. We apply the early stopping technique using the WMT21 triangular MT development set, and all models are trained for a minimum of 30 and a maximum of 50 epochs. We trained all our models using FAIRSEQ [6] (Ott et al., 2019) on 8 NVIDIA Tesla V100 GPUs.

3.2 Experimental Results

As shown in Table 2, our final model outperforms about +12.7 BLEU compared to the organizer's systems. In detail, we got the most significant performance improvement in scaling up the Transformer model. Through the data filtering process, our model achieved an improvement of about 1 BLEU. By augmenting the Russian-to-Chinese corpus, our model obtained a gain of about 1.5 BLEU. When model-level methods such as averaging parameters, ensemble, and re-ranking were applied, the BLEU score could be raised again by 1.5.

4 Discussions

4.1 Analysis of Data Filtering

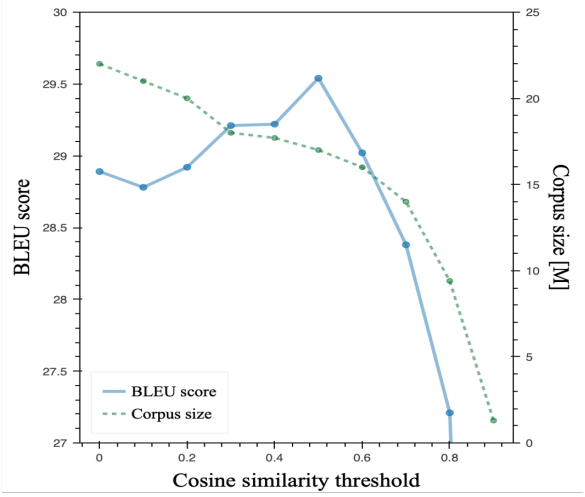
In order to verify the impact of various data filtering methods on translation performance, we conduct experiments on the direct parallel corpus (i.e., Russian-to-Chinese). As shown in Table 3, we can see that the performance improves even though we have removed more than half of the data, which means that the original data is quite noisy.

To find the best threshold value for the LaSBE filtering, we executed an additional experiment in which the threshold range is set from 0.0 to 0.9. The threshold 0.0 denotes that the filtering is not applied, and threshold 0.9 means filtering out the sentence pairs whose cosine similarity score falls below 0.9. As can be seen from the results in Figure 1a, we set the threshold value of LaBSE filtering to 0.5 in our final system. Figure 1b shows the distribution of cosine similarity scores on training data. In contrast to the distribution of the train data, that of the WMT21 triangular MT development set is clustered around 0.8. It means that the train data contain many noisy sentence pairs (nearby cosine similarity score 0.2) in terms of LaBSE sentence similarity.

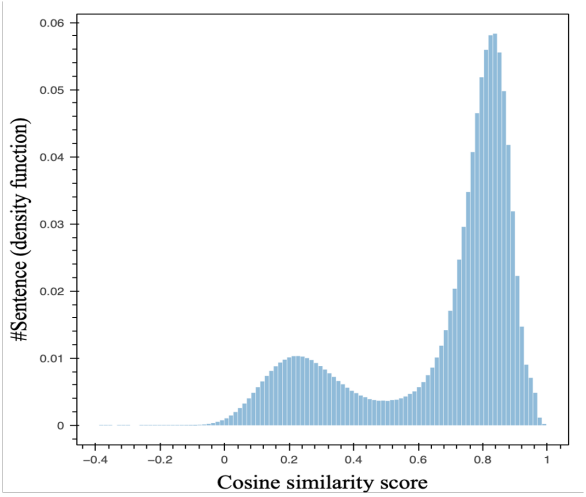
4.2 Analysis of Data Augmentation

We evaluate the impact of the different decoding methods for data augmentation. Table 4 shows the experiment results, which are consistent with Edunov et al. (2018). We observed that sampling and noise beam search are more effective than vanilla beam search. In particular, it is more effective to limit adding noise only to sentences longer than 5 (Noising beam*). As shown in the Table 4, none of the decoding strategies demonstrates superior performance. Therefore we ensemble models

[6] https://github.com/pytorch/fairseq



(a) Performances with different threshold values



(b) Cosine similarity scores on train data

Figure 1: The LaBSE filtering methods.

Systems	#Sentence	BLEU
Before augment.	13M	29.6
Beam	(13+45)M	30.2 (+0.5)
Sampling	(13+54)M	30.7 (+1.0)
Sampling top 10	(13+47)M	30.5 (+0.9)
Noising beam	(13+45)M	30.3 (+0.6)
Noising beam*	(13+45)M	30.8 (+1.1)

Table 4: Further experiment results with different decoding methods for data augmentation in Transformer-Big. The Before augment. denotes applying the basic filtering, de-duplication (pair), and LaSBE filtering to data. The asterisk(*) mark denotes the restriction to sentences with the length of tokens longer than 5.

Systems	Data	BLEU
Transformer-Large	RU2ZH*	31.3
mBART50	RU2ZH	30.3 (-1.0)
mBART50	RU2ZH*	30.9 (-0.4)
mBART50	M2ZH	29.4 (-1.9)
mBART50	M2M	30.3 (-1.1)

Table 5: Comparison between Transformer trained from scratch and fine-tuned mBART50 in an aspect of BLEU score. The asterisk(*) mark denotes augmentation with noising beam search. The M2ZH and M2M indicate RU2ZH&EN2ZH and RU2ZH&EN2ZH&RU2ZH, respectively.

trained with the different decoding methods. As a result, the ensemble model performs better, as seen in the Table 2. In an additional experiment, we also find that the performance of the augmentation (back-translation) models has a significant impact on the performance of the forward model as suggested in Hoang et al. (2018)

4.3 Bilingual MT vs Multi-lingual MT

We experimented with two ways to fully utilize the triangular MT data: to transform the indirect parallel data into direct parallel data and use them for bilingual MT as described in subsection 2.3; the another is to use the all provided data for multi-lingual MT. From the experiment result, we observed that the bilingual MT outperforms the multi-lingual MT by 1 BLEU point, and the multi-lingual MT requires more training time due to upsampling specific direction data.

4.4 Pre-trained Multi-lingual Language Model

Recently, fine-tuning pre-trained multi-lingual MT models (Liu et al., 2020; Tang et al., 2020) showed remarkable performance in multi-lingual translation scenarios. To explore the effectiveness of fine-tuning a pre-trained multi-lingual translation model for triangular MT, in the Table 5, we conducted experiments using mBART50 (Tang et al., 2020) on several datasets: (1) the augmented RU2ZH with beam search; (2) the augmented RU2ZH with noising beam search; (3) the RU-ZH and EN-ZH data; (4) the RU-ZH, EN-ZH, and RU-EN data. We use the Transformer-Large equal to mBART50 in model size as a baseline model for a fair compar-

ison. For fine-tuning mBART, the WMT21 triangular MT development set is used to compute the stopping criterion, and the models are fine-tuned for a minimum of 10 and a maximum of 20 epochs. In general, mBART transfer learning is known to be effective in low-resource language data. Fine-tuning mBART does not work well when large enough data are available. As can be seen from the experimental results, it is more effective to train the model from scratch after data augmentation.

5 Conclusion

This paper depicts Papago's submissions to the WMT21 triangular MT shared task. We have conducted extensive experiments using various techniques such as data filtering, data augmentation, model ensembling, and re-ranking in the triangular MT scenario. Except for existing techniques, we also have tried to apply data filtering with LaBSE sentence score and data augmentation using pivot language and demonstrated their effectiveness in translation performance. As a result, our system achieves the second record according to the released official results.

References

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3974–3980.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. pages 18–24.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *CoRR*, abs/1805.12282.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. *CoRR*, abs/1909.09524.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *CoRR*, abs/1904.01038.

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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

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.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. *CoRR*, abs/1906.01787.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *CoRR*, abs/2106.11520.

Machine Translation of Low-Resource Indo-European Languages

Wei-Rui Chen **Muhammad Abdul-Mageed**
Natural Language Processing Lab
The University of British Columbia
{weirui.chen,muhammad.mageed}@ubc.ca

Abstract

In this work, we investigate methods for the challenging task of translating between low-resource language pairs that exhibit some level of similarity. In particular, we consider the utility of transfer learning for translating between several Indo-European low-resource languages from the Germanic and Romance language families. In particular, we build two main classes of transfer-based systems to study how relatedness can benefit the translation performance. The *primary* system fine-tunes a model pre-trained on a *related* language pair and the *contrastive* system fine-tunes one pre-trained on an *unrelated* language pair. Our experiments show that although relatedness is not necessary for transfer learning to work, it does benefit model performance.

1 Introduction

Machine translation (MT) is currently one of the hot application areas of deep learning, with neural machine translation (NMT) achieving outstanding performance where large amounts of parallel data are available (Koehn and Knowles, 2017; Ranathunga et al., 2021; Luong et al., 2015a; Nguyen and Chiang, 2017). In low-resource settings, transfer learning methods have proven useful for improving system performance (Zoph et al., 2016; Nguyen and Chiang, 2017; Kocmi and Bojar, 2018). In this work, we focus on studying NMT, including in the low-resource scenario. In particular, we focus our attention on investigating the effect of ***language relatedness*** on the transfer process. We define relatedness of a pair of languages based on belonging to the same language family. That is, by 'related' we mean 'within the *same* language family' whereas by 'unrelated' we mean 'belong to two *different* language families'. For example, we call English and Swedish related since they belong to the Germanic language family but English and French *not* related since the latter belongs to the Romance language family. As an analogy to human learning, we would like to ask: *if there are two translators (pre-trained models), for example one Catalan → Spanish translator and one Catalan → English translator, will they (after extra training, i.e., fine-tuning/transfer learning process) have different abilities to translate from Catalan into Occitan?* If the Catalan-Spanish translator proves to perform better Catalan → Occitan, we may attribute this to Spanish and Occitan being members of the Romance language family while English being a member of the, Germanic, different family.

Of particular interest to us are two sets of languages belonging to two different language families, one set to Romance and the other set to Germanic. For the former set, we take Catalan (ca), Italian (it), Occitan (oc), Romanian (ro), and Spanish (es); and we take English (en) for the latter set. We note that both Romance and Germanic are two branches of the larger Indo-European language family, and hence there are some level of relatedness between all the languages we study in this work. Nevertheless, languages in Romance and Germanic differ in some syntactic structures. For example, the position of attributive adjectives in Germanic languages is before the noun while it is after the noun for Romance languages (Van de Velde et al., 2014). Despite differences, the writing system of all languages in this work is the Latin script. This can be beneficial to transfer learning because these languages can potentially share common lexical items or morphemes, which may facilitate the transfer learning process.

As mentioned, we adopt transfer learning since it has been shown to improve translation quality for low-resource language pairs. For example, Zoph et al. (2016) sequentially build a *parent* model and a *child* model where each is trained, respectively, on high-resource and low-resource language pairs with the child model retaining parameters from

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 347–353
November 10–11, 2021.

Lang. Pair	Primary	Contrastive
ca-it	ca-it/ca-es	ca-en
ca-oc	ca-es	ca-en
ca-ro	ca-es	ca-en

Table 1: Pre-trained model choices for our primary and contrastive NMT systems.

the parent model. In addition, Nguyen and Chiang (2017) successfully transfer knowledge from a parent model to a child model where both models are trained on low-resource but related language pairs. Kocmi and Bojar (2018) also adopt a similar approach to these previous works, but based their work on the Transformer architecture (Vaswani et al., 2017) instead of using a recurrent encoder-decoder network with attention (Bahdanau et al., 2015; Luong et al., 2015b).

Our work builds on these studies. Namely, we empirically apply transfer learning under different conditions to members of various language families. The outcomes of our work are similar to those of Zoph et al. (2016); Kocmi and Bojar (2018). That is, while we find relatedness to be beneficial, a positive transfer between an unrelated language pair can still be possible (although with a potentially diminished performance).

The rest of this paper is organized as follows: In Section 2, we overview related work. We introduce our datasets and experiments in Section 3. In Section 4, we present and analyze our results. We conclude in Section 5.

2 Background

2.1 Transfer learning

Transfer learning is a machine learning approach that aims at transferring the knowledge of one task to another. As an analogy to human learning, one who masters the skills to ride a bicycle may transfer the knowledge to riding a motorcycle because these two tasks share common abilities such as maintaining balance on a two-wheel moving vehicle (Pan and Yang, 2009; Weiss et al., 2016). We employ transfer learning to port knowledge from a model trained on one pair of languages to another. We now discuss transfer learning in NMT.

2.2 Transfer learning in Machine Translation

Zoph et al. (2016) design a framework where a parent model is trained on a high-resource language pair while retaining model parameters for the child model to start fine-tuning with. Using this method, Zoph et al. (2016) improve system performance by an average of 5.6 BLEU points. The improvement is realized by transferring what is learnt in the high-resource language pair to the low-resource language pair. The Uzbek-English model obtains 10.7 BLEU score without the parent model and improves to 15.0 with the French-English parent model. The Spanish-English model has 16.4 BLEU score without the parent model and 31.0 with the French-English parent model. These results show that applying transfer learning contributes 4.3 and 14.6 BLEU points gain. Based on results from Zoph et al. (2016), the closer the two source languages, the more performance gain acquired. Due to the relatedness between Spanish and French (both are members of the Roman language family), performance gain is higher for this pair.

Following previous work, Nguyen and Chiang (2017) design a paradigm similar to that of Zoph et al. (2016) but maintain one major difference. In particular, Nguyen and Chiang (2017) try to make use of relatedness between the parent and child models at the vocabulary level: instead of randomly mapping tokens in the parent and child vocabulary, they retain the parent tokens for the child model if these tokens exist in child language pair. This approach is based on two assumptions - (i) the lexicons of the parent and child language pair have at least some partial overlap and (ii) these identical tokens have similar meaning. Instead of the word-level tokenization in Zoph et al. (2016), Nguyen and Chiang (2017) use Byte Pair Encoding (BPE) (Gage, 1994; Sennrich et al., 2016) to obtain subword tokens which may increase the number of overlapped tokens between the parent and child models. Improvement of 0.8 and 4.3 in BLEU score were obtained for the Turkish-English and Uyghur-English child models as transferred from an Uzbek-English parent model.

Following the previous two works, Kocmi and Bojar (2018) take a similar approach but use the Transformer architecture. They obtain an improvement of 3.38 BLEU for an English-Estonian child model transferred from an English-Czech parent model. Similarly, Neubig and Hu (2018) add a second language related to the added low-resource language to avoid overfitting when fine-tuning. This mechanism has shown to be effective. Other works have investigated NMT approaches

to similar languages by pre-training new language models on the low-resource languages (Nagoudi et al., 2021) or without necessarily applying transfer learning (Przystupa and Abdul-Mageed, 2019; Adebara et al., 2020; Barrault et al., 2019, 2020), and there are several works on low resource languages (Adebara et al., 2021). We now introduce our experimental settings.

3 Experimental Settings

3.1 Languages & Settings

We carry out experiments on three language pairs: *ca-it*, *ca-oc*, and *ca-ro*. The number of parallel sentences of each dataset is shown in Table 4. Training data are from OPUS (Tiedemann, 2012), particularly version 1 of the WikiMatrix datasets (Schwenk et al., 2021). They are data of child language pair and are used to fine-tune pre-trained model. Development and test data are provided by organizers of the Multilingual Low-Resource Translation for Indo-European Languages shared task. The shared task is hosted in EMNLP 2021 Sixth Conference on Machine Translation (WMT21).

We build two systems: a ***Primary system*** and a ***Contrastive system***. The primary system fine-tunes pre-trained *ca-es* and *ca-it* models, while the contrastive system fine-tunes a pre-trained *ca-en* model as shown in Table 1. The primary and contrastive systems serve as context for studying the role of language ***relatedness*** in transfer learning in NMT. We submitted predictions of the two systems to the WMT2021 shared task, and evaluation was based on blind test sets with a number of metrics as run by shared task organizers. An exception is the *ca-it* language pair fine-tuned on top of the *ca-es* pre-trained model–we train the model of this pair post shared task formal evaluation.

3.2 Model Architecture

We leverage publicly accessible pre-trained models on Huggingface (Wolf et al., 2020) from Helsinki-NLP (Tiedemann and Thottingal, 2020). The pre-trained MT models released by Helsinki-NLP are trained on OPUS. These models are Transformer-based implemented in the Marian-NMT framework (Junczys-Dowmunt et al., 2018). Each model has six self-attention layers in both the encoder and decoder sides, and each layer has eight attention heads. The tokenization method is SentencePiece (Kudo and Richardson, 2018) which produces vocabulary of size $49,621$, $21,528$ and $55,255$ for *ca-es, ca-it*, and *ca-en* models, respectively.

3.3 Approach

The pre-trained models are chosen based on the degree of *relatedness* of the original target language on which the model is trained and the new target language on which the model is fine-tuned. *Primary system* takes related languages while contrastive system takes unrelated languages. Since Catalan, Italian, Occitan, Romanian, and Spanish are all members of the Roman language family, we take **ca-es** as our pre-trained MT model for transfer learning. As English is a member of the Germanic language family, we use a **ca-en** pre-trained model for our transfer learning. Our model choices are summarized in Table 1.

Without modifying the architecture of the MT pre-trained models, all architecture-related hyperparameters are identical to the original edition. As for hyperparameters related to fine-tuning, the number of beams for beam search is modified from four for pre-training to six for fine-tuning. The batch size is set to be 25. Pre-trained models are further fine-tuned for $30,000$ steps on OPUS bitext. The checkpoint with the lowest validation loss is then selected as our best model for prediction.

Similar to Zoph et al. (2016); Nguyen and Chiang (2017); Kocmi and Bojar (2018), to achieve transfer learning, we retain the parameters of the parent model when fine-tuning the child model. Besides, parent and child models share a common vocabulary. That is, we do not build distinct vocabularies for the parent model and child models. A shared vocabulary can contribute to better transfer learning since all our language pairs employ the same Latin writing system. We suspect a shared vocabulary is more influential when the two languages are related to each other since the languages may have common morphemes, lexical items, or syntactical structure. For unrelated languages, a shared vocabulary may not hurt since the token embeddings are not frozen throughout the fine-tuning process. That is, token embeddings can still be updated to attain better representations during training.

3.4 Baseline Models

To demonstrate the effectiveness of our transfer learning approach, we provide a baseline model for each language pair that is simply a parent model (a

		Baseline		Our models				
Pre-trained Model	**Lang. Pairs**	**BLEU**	**chrF**	**BLEU**	**chrF**	**TER**	**COMET**	**BertScore**
ca-it	ca-it	29.31	0.583	**35.06**	**0.622**	0.477	0.391	0.886
ca-es	ca-it	7.07	0.370	**33.13**	**0.602**	0.499	-	-
ca-es	ca-oc	12.56	0.472	**59.93**	**0.787**	0.254	0.538	0.928
ca-es	ca-ro	4.43	0.266	**11.24**	**0.354**	0.855	-0.908	0.749

Table 2: Primary system results. We did not submit the ca-it language pair fine-tuned on the ca-es pre-trained model to the WMT2021 shared task, and hence the results are calculated by ourselves with Sacrebleu.

		Baseline		Our models				
Pre-trained Model	**Lang. Pairs**	**BLEU**	**chrF**	**BLEU**	**chrF**	**TER**	**COMET**	**BertScore**
ca-en	ca-it	1.97	0.249	**25.46**	**0.539**	0.574	-0.263	0.844
ca-en	ca-oc	2.16	0.258	**51.46**	**0.736**	0.316	0.259	0.905
ca-en	ca-ro	1.59	0.209	**8.61**	**0.311**	0.884	-1.119	0.725

Table 3: Contrastive system reuslts

Lang. Pairs	**Train**	**Dev**	**Test**
ca-it	1,143,531	1269	1743
ca-oc	138,743	1269	1743
ca-ro	490,064	1269	1743

Table 4: Distribution of dataset

pre-trained model) without any fine-tuning on data of the child language pair.

3.5 Evaluation

The adopted metrics are BLEU (Papineni et al., 2002), chrF (Popović, 2015), TER (Olive, 2005), COMET (Rei et al., 2020), and BERTScore (Zhang et al., 2019). BLEU, chrF and TER are measured with the implementation of *Sacrebleu* (Post, 2018)[1].

4 Results and Analysis

4.1 Primary and Contrastive Systems

As can be seen in the rightmost five columns in Table 2 and Table 3, primary system outperforms contrastive system across all metrics. We believe that **ca-es** pre-trained MT model performs better transfer learning because Spanish is closer to Italian, Occitan, and Romanian than English is to these languages. These results, as such, indicate that transfer learning between related language pairs can produce better performance than between unrelated language pairs.

4.2 Baseline and Fine-tuned Models

Our results in Table 2 and Table 3 show the effectiveness of transfer learning for both related and *unrelated* language pairs. This is the case since both systems experience a performance gain after fine-tuning.

As an interesting observation, it seems counter-intuitive to have the unrelated language pairs experience slightly higher performance gain. For example, regarding **ca-oc** language pair, the transfer learning provides 47.37 BLEU score improvement transferring from **ca-es** parent model but 49.3 BLEU score improvement transferring from **ca-en** parent model. We suspect this is because in our work, when fine-tuning, we fix source language and alter the target language.

Unlike multilingual MT models which requires target language label to be prepended at the beginning of a source sentence (Johnson et al., 2017) or notifying the model what target language is for this forward propagation (Liu et al., 2020), the pre-trained models we use in this work are bilingual models which lack a mechanism to provide the model any information about current target language. Therefore, the **ca-en** pre-trained model does not know it should now be translating Catalan to Occitan instead of English. Due to producing pre-

[1] https://github.com/mjpost/sacrebleu

diction in an incorrect target language, the metrics will be very poor. After fine-tuning the parent models on data of the child language pairs, the models are likely abler to produce prediction in the correct target language. Due to baseline metrics being too low, the difference in metric values between non-fine-tuned (baseline) and fine-tuned models are large and that is why the performance gain can be higher in contrastive system than in primary system.

5 Conclusion

In this work, we confirm previous works showing that transfer learning benefits NMT. Besides, an empirical comparison between transferring from related and unrelated languages shows that relatedness is not strictly required for knowledge transfer, but it does result in higher performance than transferring with unrelated languages.

Acknowledgements

We appreciate the support from the Natural Sciences and Engineering Research Council of Canada, the Social Sciences and Humanities Research Council of Canada, Canadian Foundation for Innovation, Compute Canada (www.computecanada.ca), and UBC ARC–Sockeye (https://doi.org/10.14288/SOCKEYE).

References

Ife Adebara, Muhammad Abdul-Mageed, and Miikka Silfverberg. 2021. Translating the unseen? yoruba-english mt in low-resource, morphologically-unmarked settings. *arXiv preprint arXiv:2103.04225*.

Ife Adebara, El Moatez Billah Nagoudi, and Muhammad Abdul Mageed. 2020. Translating similar languages: Role of mutual intelligibility in multilingual transformers. In *Proceedings of the Fifth Conference on Machine Translation*, pages 381–386, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Loïc Barrault, Magdalena Biesialska, Ondrej Bojar, M. Costa-jussà, C. Federmann, Yvette Graham, Roman Grundkiewicz, B. Haddow, M. Huck, E. Joanis, Tom Kocmi, Philipp Koehn, Chi kiu Lo, Nikola Ljubesic, Christof Monz, Makoto Morishita, M. Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *WMT@EMNLP*.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussa, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

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.

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.

Tom Kocmi and Ondrej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. *CoRR*, abs/1809.00357.

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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Minh-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.

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, Lisbon, Portugal. Association for Computational Linguistics.

El Moatez Billah Nagoudi, Wei-Rui Chen, Muhammad Abdul-Mageed, and Hasan Cavusogl. 2021. Indt5: A text-to-text transformer for 10 indigenous languages. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 265–271.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. *CoRR*, abs/1708.09803.

Joseph Olive. 2005. Global autonomous language exploitation (gale). *DARPA/IPTO Proposer Information Pamphlet*.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

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.

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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Michael Przystupa and Muhammad Abdul-Mageed. 2019. Neural machine translation of low-resource and similar languages with backtranslation. In *Proceedings of the 4th Conference on MT (Volume 3: Shared Task Papers, Day 2)*, pages 224–235.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2021. Neural machine translation for low-resource languages: A survey.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. 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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Freek Van de Velde, Petra Sleeman, and Harry Perridon. 2014. The adjective in germanic and romance. *Adjectives in Germanic and Romance*, 212:1.

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*, volume 30. Curran Associates, Inc.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

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.

CUNI systems for WMT21: Multilingual Low-Resource Translation for Indo-European Languages Shared Task

Josef Jon and **Michal Novák** and **João Paulo Aires** and **Dušan Variš** and **Ondřej Bojar**
Charles University
{jon,mnovak,aires,varis,bojar}@ufal.mff.cuni.cz

Abstract

This paper describes Charles University submission for Multilingual Low-Resource Translation for Indo-European Languages shared task at WMT21. We competed in translation from Catalan into Romanian, Italian and Occitan. Our systems are based on shared multilingual model. We show that using joint model for multiple similar language pairs improves upon translation quality in each pair. We also demonstrate that chararacter-level bilingual models are competitive for very similar language pairs (Catalan-Occitan) but less so for more distant pairs. We also describe our experiments with multi-task learning, where aside from a textual translation, the models are also trained to perform grapheme-to-phoneme conversion.

1 Introduction

The goal of the task was to translate text from Catalan into Occitan, Italian and Romanian. Additionally, use of parallel corpora which combine the evaluated languages with English, French, Portuguese and Spanish was permitted. The choice of the languages from the same family invites to explore how to take advantage of similarities between the languages.

One way to exploit similarities between the languages translated by an NMT model is to train a single joint model for multiple languages. This way, parameters representing rules and features which are common for multiple languages can be shared and better estimated due to a larger amount of training examples related to them.

Another approach which can be effective when source and target languages are very similar is character-level processing of the text. Since most of the differences between Catalan and Occitan are straightforward orthographic variations, we hypothesize that the translation model would benefit from being able to manipulate the text at character level instead of larger subwords.

We also explore making use of language similarity in spoken form, aside from written form. Languages from the same language group may be more mutually intelligible in their spoken form rather than in the written form. For instance, based on our anecdotal observations, native speakers of Czech report better understanding of spoken rather than written Polish. This is mainly due to Polish orthography, which is regular but uses various digraphs, making Polish texts less comprehensible for common Czech speakers. Phonemic representations may be even more helpful for languages with irregular spelling.

Instead of using automatically acquired phonemic representation as one of the inputs, we rather focus on strengthening robustness of our translation models by teaching them to produce this representation as an additional task. Some of our models are thus trained to provide machine translation as well as grapheme-to-phoneme conversion (G2P) of the source.

2 Main features of our approach

The core of our approach lies in leveraging multilingual training data, various subword granularity and phonemic representation of texts by multi-task learning.

All our models are instances of the Transformer architecture (Vaswani et al., 2017) as implemented in the MarianNMT (Junczys-Dowmunt et al., 2018). For the final submissions, we trained several models in multiple stages and tuned the decoding hyperparameters. Moreover, we applied character-level rescoring for the Catalan-Occitan submissions.

2.1 Data preparation

In this section we describe our preprocessing steps, the relevant code is available at https://github.com/ufal/bergamot.git/wmt21-multi-low-res

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 354–361
November 10–11, 2021.

Mulilinguality. It has been shown (e.g. by Zhang et al. (2020); Fan et al. (2020); Firat et al. (2016); Tan et al. (2019); Arivazhagan et al. (2019); Lakew et al. (2018)) that combining multiple translation directions into one model may be beneficial for the translation quality (especially for related languages) in the low-resource scenarios due to knowledge transfer between the translation directions, as it allows the model to get better estimates of the parameters that represent principles which are shared between the languages.

For our multilingual systems, we use the vanilla Transformer (single encoder, single decoder), concatenate the training data and insert a special token at the start of each source sentence to mark the desired target language, e.g. for translation from Catalan into Occitan: `<oc> Tres dels seus costats tenen porxada.`

Subwords granularity and character-level translation. It has been shown (Sennrich and Zhang, 2019) that granularity of subword segmentation and thus the resulting vocabulary size has a large effect on translation quality in low-resource scenarios. For mid- and high-resource language pairs, vocabulary size of around 32k subwords is the usual choice. However, for smaller corpora, this size causes sparsity problems, since the vocabulary contains many subwords that were seen too few times to estimate sufficiently good embeddings for them. The solution is to split the words into smaller subwords or even into single characters. Moreover, we suspected that for similar languages, like Catalan and Occitan, small subword or character level translation may be beneficial because large part of the differences between the translations are merely orthographic variations and the ability to work on character level will allow the model to learn to perform these variations more easily.

Grapheme-to-phoneme conversion as an extra task. We hypothesize that teaching the model both to translate and to perform G2P may increase the model's robustness and consequently its performance. Multi-task learning (Caruana, 1997) has been successfully shown in NMT to either incorporate linguistic knowledge (Luong et al., 2016; Eriguchi et al., 2017; Kiperwasser and Ballesteros, 2018) or to exploit monolingual data (Wang et al., 2020). Although it has been also used in G2P (Prabhu and Kann, 2020), the two tasks has not been to the best of our knowledge modelled jointly so far.

Using a G2P tool, we prepare phonemic representation of the source side of the training data and combine it with the text data in two possible ways.

Vertical combination is an analogy of how multiple translation directions are combined. We concatenate the bitext with the data that consist of the same source side and its phonemic representation as the target side. Furthermore, we use a special token at the start of each source sentence to indicate the G2P task, e.g. `<ca_p>` for Catalan phonemization.

In *horizontal combination*, we attempt to mimic multi-output learning (Xu et al., 2019), i.e. producing outputs for multiple tasks at the same time. We thus enrich each target sentence with the phonemic representation of the source sentence. The two are separated by a special symbol `<sep>`. To evaluate the MT output, we need to strip off the phonemic part first.

2.2 Model training and decoding

Learning stages. Some of the models submitted to the shared task are a result of learning in two consecutive stages, each utilizing a different dataset. In the pre-training stage, we build a general multilingual model, leveraging most of the available data sources. In the fine-tuning stage, we continue training only on selected languages, possibly in conjunction with learning to convert graphemes to phonemes.

Decoding. During the beam search, we normalize the scores of each hypothesis by its length (the score is divided by $length^n$). We performed grid search over the n coefficient and beams size for our primary submission and we obtained values $n = 1.0$ and $b = 8$. We used these values for all the systems.

Character-level rescoring. For Catalan-Occitan, we found character-level models to be competitive with subword models, but after manual inspection, we see some of the translations produced by these models included superfluous repetitions of groups of characters. For this reason, we decided to use the character-level model only for rescoring hypotheses produced by the subword-level models.

	ca	en	fr	it	oc	ro
ca	-	1305	2501	1756	57	1106
en	-	-	-	6434	37	1445
fr	-	-	-	21721	124	4815

Table 1: Number of lines (in thousands) in corpora for each language pair used in our systems.

3 Datasets

Apart from the Catalan, Occitan, Romanian and Italian data, we take advantage of the data in other languages allowed by the Shared Task organizers: Spanish, French and English (we did not use Portuguese corpora). We used datasets specified by the task organizers, namely ParaCrawl, GlobalVoices, EuroParl, JW300, WikiMatrix, MultiCCaligned, Opus100, Books and Bible. Table 1 shows number of lines for each language pairs used in our experiments.

4 Results

In this section, we report BLEU (Papineni et al., 2002) and ChrF2 citepopovic-2015-chrf scores on development and test sets provided by the organizers. We did not rerun test set evaluations for all the models, so for a small number of configurations we only show scores on the development sets.

4.1 Tools

We break the input text into subwords using SentencePiece (Kudo and Richardson, 2018). We use MarianNMT (Junczys-Dowmunt et al., 2018) to train the models and the BLEU and ChrF scores are computed using SacreBLEU (Post, 2018). For experiments involving G2P conversion, we used `phonemizer` wrapper script[1] around Espeak-ng speech synthesizer[2] to produce phonemic representation of the texts.

4.2 Baselines

We used publicly available services and models as external baselines, and traditional bilingual Transformer models trained on provided corpora as our own baselines. We use SentencePiece preprocessing with 8k subword models for our bilingual baselines. We also trained models to translate from Catalan to English and from English to the target languages to be able to do pivoted translation. The external baselines include Google Translate (for Romanian and Italian), Romance multilingual model[3] from Opus-MT project (Tiedemann and Thottingal, 2020) and Apertium rule-based machine translation system (Forcada et al., 2011), which was chosen since we suspected that the rule-based approach might work better than NMT for very low resource, but very similar language pairs, like Catalan-Occitan (and also Apertium is especially focused on languages of that region). Results on dev and test sets are presented in Tables 2 and 3, respectively.

System	BLEU			ChrF		
	it	ro	oc	it	ro	oc
Opus-MT	32.4	-	16.7	0.608	-	0.545
Google	32.3	28.7	-	0.609	0.554	-
Apertium	32.1	14.9	67.0	0.619	0.461	0.834
Bilingual	42.1	29.8	59.2	0.674	0.559	0.789
Pivot	37.7	20.3	0.6	0.636	0.505	0.082

Table 2: Results of the baseline system evalutation, development set.

System	BLEU			ChrF		
	it	ro	oc	it	ro	oc
Opus-MT	33.7	-	17.3	0.612	-	0.544
Apertium	34	13.3	67.5	0.624	0.408	0.834
Bilingual	44.9	26.7	59.4	0.687	0.497	0.787

Table 3: Results of the baseline system evaluation, test set.

We see that even our bilingual baselines outperform all other baselines aside from Apertium on Catalan-Occitan. We were unable to train functional English-Occitan model on the provided data (only 37k noisy sentence pairs), so the pivoted approach was not feasible in this direction.

4.3 Improving bilingual models

Before working on multilingual models, we focused on improving the bilingual systems to be sure our baselines are sufficiently strong.

First, we add backtranslated data. We trained a joint multilingual model for translation from the target languages into Catalan. For Romanian and Italian, we used this model to translate Wikipedia,[4]

[1] https://github.com/bootphon/phonemizer

[2] https://github.com/espeak-ng/espeak-ng

[3] https://github.com/Helsinki-NLP/OPUS-MT-train/tree/master/models/ca+es+fr+ga+it+la+oc+pt_br+pt-ca+es+fr+ga+it+la+oc+pt_br+pt

[4] We obtained the most recent dumps from https://dumps.wikimedia.org/

BT	BLEU			ChrF		
	it	ro	oc	it	ro	oc
none	42.1	29.8	59.2	0.674	0.559	0.789
w, scr.	43.5	32.7	64.3	0.680	0.584	0.818
w, finet.	-	-	62.5	-	-	0.810
g, scr.	-	-	63.4	-	-	0.815
g, finet.	-	-	61.4	-	-	0.803
w(c)	-	-	64.7	-	-	0.819
w(c) big	-	-	65.2	-	-	0.821

Table 4: Adding backtranslation, development set. *w* denotes backtranslated data originating from Wikipedia dumps, *g* denotes general texts, *scr.* denotes a system that was trained from scratch, *finet.* denotes a system that was initialized by a baseline model trained on parallel data and finetuned, *(c)* means character-level model and *big* means that transfomer-big model was used instead of base.

BT	BLEU			ChrF		
	it	ro	oc	it	ro	oc
none	44.9	26.7	59.4	0.687	0.497	0.787
w, scr.	45.8	28.4	64.3	0.690	0.511	0.815
w, finet.	-	-	62.4	-	-	0.805
g, scr.	-	-	63.6	-	-	0.813
g, finet.	-	-	61.6	-	-	0.801
w(c)	-	-	64.8	-	-	0.818
w(c) big	-	-	65.2	-	-	0.821

Table 5: Adding backtranslation, test set. Meaning of the rows is descirbed in previous table.

Vocab	BLEU			ChrF		
	it	ro	oc	it	ro	oc
8k	42.1	29.8	59.2	0.674	0.559	0.789
2k	42.4	30.3	59	0.676	0.565	0.792
char	38.8	28.6	62.6	0.652	0.555	0.808
char-f	41.2	28.3	62.1	0.669	0.554	0.808

Table 6: Results with varying vocabulary size, development set. *Char-f* models are the original 8k models subsequently finetuned one character-level data.

Vocab	BLEU			ChrF		
	it	ro	oc	it	ro	oc
8k	45	26.7	59.6	0.687	0.497	0.787
2k	44.5	26.1	59.1	0.685	0.495	0.788
char	40.9	24.6	63.5	0.665	0.487	0.812
char-f	43.5	24.8	62.3	0.678	0.489	0.806

Table 7: Results with varying vocabulary size, test set. *Char-f* models are the original 8k models subsequently finetuned one character-level data.

for Occitan, we utilized Apertium and aside from Wikipedia, we also translated Occitan sides of all the other provided parallel corpora. The results are presented in Tables 4 and 5. We see that backtranslation improves results for all the language pairs, and that for Occitan, wiki translation (rows marked as *w*) works better than general corpora backtranslation obtained from Occitan sides of other parallel corpora (En-Oc, Fr-Oc and Es-Oc). We also observe that the performance is better when training with parallel and BT data from the beginning (*scr.*), opposed to finetuning parallel-only trained model on parallel-BT mix (*finet.*).

We also tried to improve the results by choosing a correct subword granularity. We compared baseline models, which use SentencePiece vocabulary with 8k tokens, with 2k tokens and character level translation (see Tables 6 and 7). Based on observations by Libovický and Fraser (2020), we trained character level models both from scratch (row *char*) and by finetuning the subword models (row *char-f*). We see that the character-level training works best for Catalan to Occitan translation. We suppose it partially stems from the lack of resources for the language pair and partially from the relative similarity of the two languages.

We combined the backtranslation and character level processing for Occitan to see if the improvements are orthogonal (Tables 4 and 5). We also trained transformer-big models on the same data for comparison with larger models introduced in the next section.

4.4 Multilingual models

Our final submission is based on multilingual models. We combined the datasets allowed for the task and included a special language tag at the beginning of the source sentence to indicate the target language. The results on dev and test sets are presented in Tables 8 and 9. We use 32k vocabulary for the multilingual models.

Firstly, we trained a model only on the languages that were evaluated (system 1). We see that just by using the joint model, we obtained improved results for all language pairs. We also trained transformer-big model on the same data, as increasing model capacity usually improves performance especially for multilingual settings (system 2), but we observed same or worse results than with a base model.

Next, we added corpora with the other allowed translation directions which contain the evaluated

		BLEU			ChrF		
i	**Description**	**it**	**ro**	**oc**	**it**	**ro**	**oc**
1	ca-oc,ro,it	43.7	33.2	63.8	0.681	0.582	0.816
2	1 + transformer-big	43.1	34.0	63.5	0.681	0.585	0.815
3	ca,fr,es,en-oc,ro,it	42.8	33.7	54.5	0.675	0.584	0.761
4	3 + balanced	41.7	33.6	62.9	0.667	0.583	0.806
5	3 + balanced, bt	41.8	33.0	60.3	0.672	0.585	0.789
6	3 + transformer-big	44.7	35.1	57.4	0.688	0.594	0.778
7	3 + transformer-bigger	42.6	33.7	52.1	0.672	0.582	0.749
8	3 + ca-es, ca-fr, ca-en	44.5	34.6	55.5	0.686	0.591	0.769
9	8 + big	46.7	37.1	59.1	0.700	0.607	0.792
10	8 + bigger (430k updates)*	47.1[1]	38.0[1]	59.8	0.702	0.613	0.794
11	8 + bigger (2.1M updates, converged)	48.5	39.2	62.7	0.714	0.624	0.808
12	10 + bt	46.3[2]	36.5[2]	59.2	0.701	0.608	0.792
13	10 + finetuning for lang pair + bt	44.6	34.4	65.6	0.689	0.597	0.824
14	13 + char-level rescoring	-	-	67.1[1]	-	-	0.833
15	9 + ca-it,oc; vert. multi-task	45.2	-	65.3	0.690	-	0.823
16	9 + ca-it,oc; balanced vert. multi-task	42.9	-	65.7	0.675	-	0.825
17	16 + char-level rescoring	-	-	66.8[2]	-	-	0.832

Table 8: Results of our multilingual models, dev set. [1] marks our primary submissions, [2] is our secondary submission.

		BLEU			ChrF		
i	**Description**	**it**	**ro**	**oc**	**it**	**ro**	**oc**
1	ca-oc,ro,it	45.9	29.2	63.9	0.692	0.513	0.814
2	1 + transformer-big	45.7	29.0	63.2	0.691	0.511	0.808
3	ca,fr,es,en-oc,ro,it	46.0	29.3	55.1	0.690	0.513	0.760
4	3 + balanced	45.0	29.1	63.3	0.684	0.511	0.803
5	3 + balanced, bt	44.3	28.9	60.8	0.685	0.515	0.788
6	3 + transformer-big	47.7	30.6	58.0	0.701	0.522	0.778
7	3 + transformer-bigger	46.7	30.1	54.8	0.693	0.517	0.759
8	3 + ca-es, ca-fr, ca-en	47.4	29.8	55.5	0.699	0.517	0.764
9	8 + big	49.1	31.7	59.5	0.710	0.531	0.788
10	8 + bigger (430k updates)	50.5[1]	32.8[1]	60.3	0.717	0.533	0.792
11	8 + bigger (2.1M updates, converged)	51.1	33.9	62.6	0.722	0.544	0.804
12	10 + bt	49.5[2]	31.8[2]	59.9	0.713	0.533	0.792
13	10 + finetuning for language pair + bt	47.3		66.6	0.702		0.825
14	13 + char-level rescoring	-	-	66.9[1]	-	-	0.829
15	9 + ca-it,oc; vert. multi-task	48.6	-	65.2	0.706	-	0.819
16	9 + ca-it,oc; balanced vert. multi-task	45.3	-	65.5	0.687	-	0.820
17	16 + char-level rescoring	-	-	67.1[2]	-	-	0.832

Table 9: Multilingual models, test set. [1] marks our primary submissions, [2] is our secondary submission.

languages on their target side, i.e. French, Spanish and English into Occitan, Romanian and Italian (system 3). At the first glance, including additional related languages did not improve the performance (and even hurts the performance for Catalan-Occitan), but we suspected that this might be a model capacity and data balancing problem. After oversampling the smaller training corpora to have the same number of sentences as the largest one, we see that performance of the model for this pair (4) reaches the levels of the previous model. Interestingly, adding backtranslated Wikipedia results in worse scores, even though backtranslation helped in bilingual models (5). To see whether increasing the model capacity while using larger amount and more diverse training data is beneficial, we trained transformer-big (6) and transformer-big with 12-layer encoder instead of 6-layers, which we call transformer-bigger (7). For transformer-bigger, we used depth-scaled initialization proposed by Zhang et al. (2019). We see that in fact, after adding more data, larger model capacity helps, but the 12-layered encoder transformer-big performs worse than the 6-layered one. We believe this is caused by instability of the training for the deeper models as in the next paragraph, we see improvements with the deeper model.

Until now, our goal was to mainly improve the target language generation by including other corpora with evaluated languages at the target side. We also tried to improve source-side Catalan encoding by adding corpora with Catalan on the source side, namely Catalan to French, English and Spanish (8). Resulting model shows improvements compared to the other language combinations, and again, increasing the model size ((9), (10) and (11[5])) has even larger effect than for the previous models due to the amount and diversity of the training data. We hypothesize that increasing depth of the encoder helps in this case compared to the previous model because we added more data with Catalan source side and the increased encoder capacity could be used to learn more Catalan-specific features and rules.

Our primary submissions for Romanian and Italian are simply translations produced by the largest multilingual model (10). The training has not fully converged at the time of the submission and further training brought improvements in the range of 1-3 BLEU. Our secondary submissions for these two languages were the same models, however, we also included the backtranslated Wikipedia (12) in the training dataset. Surprisingly, this approach lead to decrease in performance in terms of BLEU and ChrF2. On the other hand, BERT and COMET scores in the official evaluation are same or slightly better for the models trained with backtranslation.

	ca2it		ca2oc	
	z-score	**raw**	**z-score**	**raw**
HUMAN	0.8±0.4	4.8±0.6	0.8±0.7	4.0±1.0
CUNI-Primary	**0.5±0.7**	**4.4±0.9**	**0.5±0.8**	**3.6±1.1**
M2M-100	0.4±0.7	4.2±1.0	-0.7±0.8	2.0±1.0
TenTrans-Primary	0.0±0.8	3.8±1.1	0.3±0.8	3.4±1.2
BSC-Primary	-0.1±0.8	3.7±1.1	0.3±0.9	3.4±1.2
UBCNLP-Primary	-0.5±1.0	3.1±1.3	0.0±0.9	3.0±1.2
mT5-devFinetuned	-1.2±0.9	2.3±1.2	-1.0±0.7	1.7±0.9

Table 10: Results of human evaluation performed by the organizers.

Due to the data imbalance, even the largest model underperforms in Catalan-Occitan. Because of the time constraints, we did not try oversampling Occitan corpora and training with balanced data, instead we fine-tuned the multilingual model for specific language pairs (13[6]). Finally, we produced 20 best hypotheses for each sentence and rescored them by the character level Catalan-Occitan transformer-big introduced earlier (Table 4), leading to a 1.5 BLEU increase on the dev set. This is our primary system for Catalan-Occitan.

Our submissions were ranked first in all directions with respect to all metrics except for the Catalan-Romanian BLEU score, where the M2M model was 0.2 points better (but after finishing the training, our model outperforms it by 0.8 BLEU).

For translation into Occitan and Italian, the organizers also performed human direct assessment evaluation. Translations produced by different systems were scored from 1 to 5 (on sentence-level, but document-level context was provided to the annotators). The results are shown in Table 10.

4.5 Multi-task models

In our experiments with multi-task learning, we trained the models to be able to both translate and perform G2P conversion of the source. Using the `phonemizer` script, we automatically acquired phonemic representations of the Catalan sides in the Catalan-Italian, Catalan-Romanian and Catalan-Occitan data. We then combined them with the

[5]Model available at http://hdl.handle.net/11234/1-3769

[6]Catalan-Occitan model available at http://hdl.handle.net/11234/1-3770

Description	BLEU it	BLEU ro	BLEU oc	ChrF it	ChrF ro	ChrF oc
tgt horiz.	43.2	31.0	62.5	0.680	0.568	0.811
tgt vert.	43.2	31.6	63.6	0.679	0.573	0.817
it,oc horiz.	43.3	–	63.9	0.681	–	0.818
it,oc vert.	42.9	–	64.5	0.678	–	0.821
it,ro,oc horiz.	42.5	32.8	63.4	0.675	0.578	0.814
it,ro,oc vert.	43.1	32.8	63.4	0.678	0.579	0.815

Table 11: Results of multi-task models on dev set. The source side always consists of Catalan texts. The top part shows bilingual models, while the models in the bottom part are multilingual.

original bitexts as proposed in Section 2.1.

As shown in Table 11, we started with training multi-task transformer-base models from scratch using vocabularies of 32k tokens.[7] Apart from translation to Italian, multilingual models (in the bottom part) outperform the bilingual models (in the bottom). In addition, vertical combination of texts and phonemes appears to perform better than the horizontal one.

Comparison of Tables 11 and 8 suggests that even though trained from scratch multi-task learning seems to achieve competitive results for Catalan-Occitan. We thus focus on this language pair in the following steps. Interestingly, best scores for Occitan are achieved with a multilingual model that excludes Romanian. We suppose Occitan is too distant from Romanian to benefit from it. Therefore, we took the best-performing multilingual model at the time (system 9 in Tables 8 and 9) and fine-tuned it with the Catalan-Italian and Catalan-Occitan training sets vertically combined with Catalan phonemes for these datasets (15). As data balancing in multilingual models proved to be beneficial for Occitan, we also applied it before the fine-tuning, which results to even better performance for Occitan (16[8]). Finally, we rescored 20 best hypotheses by char-level Catalan-Occitan model as in the system 14, resulting in our contrastive submission for Catalan-Occitan (17). Within all submitted Catalan-Occitan systems, our submission was ranked first in all metrics.

5 Conclusion

We described our submission to the shared task, which ranked first according to the majority of the used metrics for all languages. We used multilingual transformer models and we present results showing that combining all the languages into single model improves upon bilingual baseline by a large margin. We also present our findings about using multi-task learning, where aside from translation of the source, the model also learns to convert the source sentence from graphemes to its phonemic form.

[7] Except for Occitan bilingual model, which uses a vocabulary of 8k tokens.

[8] Catalan-Occitan model available at http://hdl.handle.net/11234/1-3772

Acknowledgements

Our work is supported by the grants H2020-ICT-2018-2-825303 (Bergamot) of the European Union, 19-26934X (NEUREM3) of the Czech Science Foundation and SVV 260 575. Our work has also been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure, supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *ArXiv*, abs/2010.11125.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

M. Forcada, Mireia Ginestí-Rosell, J. Nordfalk, Jimmy O'Regan, Sergio Ortiz Rojas, Juan Antonio Pérez-Ortiz, F. Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25:127–144.

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.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2020. Towards reasonably-sized character-level transformer NMT by finetuning subword systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2572–2579, Online. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nikhil Prabhu and Katharina Kann. 2020. Frustratingly easy multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 123–127, Online. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

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.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034, Online. Association for Computational Linguistics.

Donna Xu, Yaxin Shi, Ivor W. Tsang, Yew-Soon Ong, Chen Gong, and Xiaobo Shen. 2019. A survey on multi-output learning. *CoRR*, abs/1901.00248.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Transfer Learning with Shallow Decoders: BSC at WMT2021's Multilingual Low-Resource Translation for Indo-European Languages Shared Task

Ksenia Kharitonova, Ona de Gibert Bonet, Jordi Armengol-Estapé, Mar Rodríguez i Alvarez, Maite Melero
Text Mining Unit, Barcelona Supercomputing Center
{ksenia.kharitonova,ona.degibert,jordi.armengol,
mar.rodriguez1,maite.melero}@bsc.es

Abstract

This paper describes the participation of the BSC team in the WMT2021's Multilingual Low-Resource Translation for Indo-European Languages Shared Task. The system aims to solve the Subtask 2: Wikipedia cultural heritage articles, which involves translation in four Romance languages: Catalan, Italian, Occitan and Romanian.

The submitted system is a multilingual semi-supervised machine translation model. It is based on a pre-trained language model, namely XLM-RoBERTa, that is later fine-tuned with parallel data obtained mostly from OPUS. Unlike other works, we only use XLM to initialize the encoder and randomly initialize a shallow decoder. The reported results are robust and perform well for all tested languages.

1 Introduction

We present the work carried out by the BSC Team in the context of WMT2021's first edition of the Multilingual Low-Resource Translation Shared Task. The tasks addresses the issue of multilinguality in machine translation (MT) for low-resource languages, focusing on two language families: North Germanic and Romance. We take part in the Subtask 2, which involves translation in four Romance languages: Catalan, Italian, Occitan and Romanian.

2 Background

Machine translation for low-resource languages is characterised by the lack of sufficient parallel data of a given language pair, either because the combination is infrequent or because the languages involved are themselves low-resource. Several works have attempted to overcome this pitfall, using different techniques. A common solution is to employ back-translation (Sennrich et al., 2016), while other research focuses on using other languages as pivots to compensate for the lack of data (Firat et al., 2016; Zoph et al., 2016). Artetxe et al. (2018); Lample et al. (2018) make use of monolingual data only.

Our approach is based on multilinguality. Previous works such as Vergés Boncompte and Ruiz Costa-Jussà (2020); Tubay and Costa-Jussa (2018) have shown that the use of multilingual MT is beneficial, as it generalizes better by sharing parameters among all the languages involved, especially if the languages belong to the same linguistic family. At the same time, training of multilingual MT models from scratch usually requires large parallel corpora and may not be feasible in a low-resource and zero-resource translation scenarios.

Pre-training of large language models from scratch on monolingual data and then fine-tuning them for the specific downstream tasks, has proved to be an extremely successful approach for many natural language processing problems. Cross-lingual language models such as XLM and XLM-RoBERTa (Conneau and Lample, 2019; Conneau et al., 2020), that combine unsupervised (monolingual data) and supervised (parallel data) training objectives, perform especially well both on cross-lingual NLU tasks and in machine translation. The idea of combining the power of pre-trained cross-lingual language models with a multilingual machine translation setting naturally follows from there.

This idea was explored in (Liu et al., 2020) where a denoising seq2seq auto-encoder (mBART) based on BART (Lewis et al., 2020) was pre-trained on extensive monolingual corpora in many languages. A similar approach is implemented in (Lin et al., 2020) where alignment information is used to pre-train a multilingual MT transformer on existing public parallel datasets. Both approaches require either a computationally intensive pre-training on monolingual data or access to extensive large-scale

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 362–367

parallel data.

Initializing an encoder and decoder of a bilingual MT seq2seq transformer with a pre-trained cross-lingual language model was famously proposed in (Conneau and Lample, 2019). A natural next step is to initialize a multilingual MT seq2seq transformer with a shared encoder and a shared decoder by the XLM-like language encoder which was first performed in (Ma et al., 2020).

We reuse this idea by initializing the encoder with a pre-trained XLM-Roberta (Conneau et al., 2020), as in (Ma et al., 2020). However, unlike (Ma et al., 2020), we only initialize the encoder, with the motivation of being able to instantiate a shallower decoder, following previous works that for a given compute budget suggest that it is more efficient to use deeper encoders and shallower decoders (Kasai et al., 2020). The encoder-only initialization was already implemented in Fairseq (Ott et al., 2019).[1]

3 Experimental Framework

3.1 Fine-Tuning Data

To train our MT system, we use all parallel data available in OPUS[2] for the targeted language pairs, namely ca-it, ca-oc, ca-ro.

We further include a small dataset ca-oc, the Catalan - Occitan Gencat Crawling, specifically obtained for the occasion, by leveraging parallel data from a crawling of the Catalan Government Internet domains and subdomains. We use the CorpusCleaner[3] pipeline to process the WARC files obtained from the crawling. This allows us to maintain the metadata and retrieve the original url per each document. We then extract the content of the same URLs in both languages and align them at document level using vecalign[4]. The final dataset of 503 sentences was obtained by manually reviewing 1,237 automatically aligned sentences. Although smaller than expected, one motivation to crawl this brand new dataset is to contribute to the development of MT resources for Occitan, which is a severely under-resourced language. We are publicly releasing this new dataset with an open license.[5]

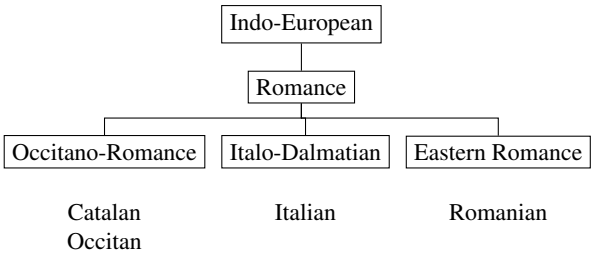


Figure 1: Family tree of Romance languages showing only the languages targeted by the Shared Task.

The resulting statistics of the corpora used to train our system can be seen in Table 1. As expected, the number of aligned sentences is much larger for Italian and Romanian as target languages, since Occitan is such a low-resource language. Nonetheless, we must bear in mind that Catalan and Occitan belong to the same sub-branch in the family tree of the Romance languages, as shown in Figure 1. Thus, their considerable typological closeness makes up for the reduced amount of aligned sentences available for this language pair.

3.2 Preprocessing

We start by preprocessing our data with a filtering and a tokenization step.

To ensure that there is no train-test overlap, we filter all of our training data by removing all sentences from the validation and test sets present in our train set.

To build our system, we use SentencePiece BPE tokenization with the original shared vocabulary of 250,000 tokens of XLM-R model (Conneau et al., 2020; Kudo and Richardson, 2018), and we only keep sentences with a maximum size of 512 tokens.

The final number of parallel sentences used for training is shown in Table 1.

3.3 System Description

We base our system on XLM-RoBERTa (Conneau et al., 2020) and then fine-tune it with the collected parallel data. As described earlier, the seq2seq multilingual transformer with shared encoders, shared decoders and shared embedding tables is initialized by XLM-R BASE pretrained language model on the encoder side, whereas a shallow decoder of 3 layers is initialized randomly. Sharing of embedding tables for all directions (ca-it, it-ca, ca-ro, ro-ca, ca-oc, oc-ca) was initially implemented due

[1] https://github.com/pytorch/fairseq/tree/v0.9.0

[2] https://opus.nlpl.eu/

[3] https://github.com/TeMU-BSC/corpus-cleaner-acl

[4] https://github.com/thompsonb/vecalign

[5] https://github.com/TeMU-BSC/wmt2021-indoeuropean/tree/master/gencat_crawling_ca-oc

Corpus Name	ca-it	ca-oc	ca-ro
EUbookshop v2	2,933	-	769
GlobalVoices v2018q4	6,036	-	468
GNOME v1	2,584	76	2,147
KDE4 v2	140,541	35,416	86,518
MultiCCAligned v1.1	1,335,785	-	890,155
OpenSubtitles v2018	359,798	-	387,044
QED v2.0a	61,013	245	57,279
Tatoeba v2021-03-10	296	-	2
TED2020 v1	49,674	33	46,978
Ubuntu v14.10	6,884	5,764	6,813
WikiMatrix v1	316,208	57,689	110,612
wikimedia v20210402	6,974	11,763	1,064
XLEnt v1.1	590,170	83,982	476,738
Catalan Government	-	503	-
Total	2,878,896	195,471	2,066,587
Cleaned	2,878,422	195,430	2,066,273
Tokenized	2,876,680	195,340	2,064,987

Table 1: Number of aligned sentences per corpus. The last row shows the final number of aligned sentences after the cleaning and tokenization steps.

Language	Tokens (M)
ca	1,752
it	4,983
oc	-
ro	10,354

Table 2: Number of million tokens per language present in the training corpus of XLM-R.

to memory constraints but eventually turned out to work well.

The token indicating the required target language is prepended to the target sentences, thus the model is aware to what language it has to translate to, as in (Wu et al., 2016).

It is important to note that the data used to train XLM-RoBERTa does not contain any Occitan text, as can be seen in Table 2. Thus the only knowledge that the multilingual transformer has about the language directions including Occitan comes from the XLM-R language model being pre-trained on text in related languages, such as Catalan.

We use default Fairseq parameters for fine-tuning, first of all, the Adam optimizer (Kingma and Ba, 2017) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The polynomial decay learning rate schedule starts from $5\mathrm{e}{-04}$, warmed up to over 1000 updates and gradually decays to 0 over around 60k updates. The model was fine-tuned for 2 days on 4 NVIDIA V100 GPUs. The final learning rate was around $3\mathrm{e}{-04}$ with a batch size of $3,072$ sentences.

During inference we use the beam search generation algorithm with a beam size of 5. Since the languages between which we are translating are typologically close, we do not assign any length penalty, and we use the best checkpoint for generating.

4 Results

Here we report the official evaluation.[6] We submitted our results a bit later than the deadline due to some bottlenecks in our in-house computational resources. Table 3 reports the results obtained by our system on the evaluation test set, together with the two official baselines provided by the organisers (M2M-100 and mt5-devFinetuned).

Out of 7 competing systems and 2 baselines, our system was ranked 5th in Average, 3rd in the Catalan-to-Occitan direction, 4th in the Catalan-to-Romanian direction and 6th in the Catalan-to-Italian direction.

4.1 Human Evaluation

The organizers of the workshop have recently released the results of a human evaluation for the

[6] http://statmt.org/wmt21/multilingualHeritage-translation-results.html

Model	ca-it	ca-oc	ca-ro	Avg.
Ours	42.00	**57.10**	24.90	**49.77**
M2M-100	**46.75**	40.24	**33.06**	40.02
mT5-dev-ft.	30.38	40.14	17.33	29.28

Table 3: Official BLEU scores for the evaluation of the final test set

language pairs ca-it and ca-oc.[7] A sentence level evaluation has been performed taking into account the document as context. Each sentence is evaluated in a Likert-like scale [1,5] answering the question of direct assessments. A second human evaluation is performed where 60 selected terms (mostly named entities, dates and locations) are annotated as being either well translated, not translated or mistranslated, by majority voting among the annotators. The results can be found in Tables 4 and 5, respectively.

5 Discussion

5.1 Little sisters over big cousins

As seen in Table 3 the average results of our system are above both baselines, although it is the results for Catalan-Occitan that give the greater leverage, because in the other two scenarios M2M has a higher BLEU. Actually, the score for Catalan-Occitan is substantially higher than the score obtained for the other two pairs, although the fine-tuning data used in this model is, at least, ten times smaller than the data used in the other two models. These results are replicated in most of the other competing systems[8]. The reason for this apparent anomaly is clearly due to the linguistic similarity between the Catalan and Occitan, which in medieval times were practically one and the same language. This result confirms the intuition that when two languages are similar enough, less data is needed.

That said, we also hypothesize a positive impact of the curated dataset (Catalan - Occitan Gencat Crawling) added to the rest of parallel data obtained in the OPUS repository, but there is no definitive proof of it. Furthermore, we can also hypothesize that the presence of Spanish in the multilingual corpus, being a high-resource language and also linguistically close to Catalan and Occitan (more so than to the other two Romance languages involved in the task), has a beneficial impact on the results. Indeed, low-resource languages can greatly benefit from their similarity to other languages present in the multilingual training. In such scenarios, less data can lead to satisfactory results, and with a smaller carbon print, since the models use less computational power for training.

5.2 Human Evaluation results

The human evaluation on two of the test sets shown in in table 4 validates the relative position of our system in the global ranking. Interestingly, human scores correlate well with BLEU for Catalan-Italian, and less well for Catalan-Occitan. In the latter case, human scores tend to be lower than the corresponding BLEU. The reason for this may again have to do with linguistic similarity between Catalan and Occitan: "Catalanish" Occitan may be deemed acceptable by subword-based BLEU, but not by human evaluators. The performance of our system as evaluated for term translation, shown in in table 5 is consistent with the other evaluations regarding the position of the system in the overall ranking.

5.3 Vocabulary

One of the shortcomings of our approach is the big vocabulary size (250k tokens), inherited from XLM. This big vocabulary size was required by XLM to cover a very diverse set of languages. However, this makes it sometimes challenging to fit the embedding tables in memory, which is especially inefficient taking into account that a large proportion of tokens are not used (since we focus on a tiny subset of languages). Thus, either pruning the vocabulary, or using pre-trained models specifically trained for the Romance languages family (with a reduced vocabulary size) would be better alternatives.

5.4 Shallow decoders and transfer learning

While recent works have suggested that allocating more computation to (deeper) encoders (Kasai et al., 2020) at the expense of allocating less computation to (shallower) decoders is more efficient, this approach is not yet standard in the machine translation literature, especially when applying transfer learning. This method has the advantage of not reusing pre-trained weights for the decoder, although a middle ground is perhaps worth exploring.

[7] http://www.statmt.org/wmt21/multilingualHeritage-translation-manual.html

[8] http://statmt.org/wmt21/multilingualHeritage-translation-results.html

	ca-it		ca-oc	
Model	z-score	raw	z-score	raw
Human	0.8±0.4	4.8±0.6	0.8±0.7	4.0±1.0
Ours	-0.1±0.8	3.7±1.1	0.3±0.9	3.4±1.2
M2M-100	0.4±0.7	4.2±1.0	-0.7±0.8	2.0±1.0
mT5-dev-ft.	-1.2±0.9	2.3±1.2	-1.0±0.7	1.7±0.9

Table 4: Official human evaluation scores at sentence level

	ca-it				ca-oc			
Model	well	mis	no	$\sum$	well	mis	no	$\sum$
Human	53	0	3	56	40	0	2	42
Ours	27	7	5	39	33	4	0	37
M2M-100	33	2	6	41	26	9	0	35
mT5-dev-ft.	20	17	10	47	25	11	4	40

Table 5: Official human evaluation scores for 60 selected terms

Namely, use just some of the pre-trained weights to initialize the decoder layers. For example reuse the first N layers of XLM in the decoder, even if there is no 1-to-1 mapping between layers because there are less in the fine-tuned model.

6 Conclusions

We have showed that our approach is a simple, yet effective method for multilingual machine translation between linguistically similar languages. The encoder-only initialization allows for having a shallow decoder, which is computationally wise. As future work, we plan to further explore transfer learning techniques in the context of shallow decoders as well as applying different vocabulary pruning techniques.

Code availability

We release[9] with an open license the scripts used for this work for the sake of reproducibility.

Acknowledgements

This work was funded by the MT4All CEF project.[10]

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7059–7069.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2020. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

[9] https://github.com/TeMU-BSC/wmt2021-indoeuropean

[10] https://ec.europa.eu/inea/en/connecting-europe-facility/cef-telecom/2019-eu-ia-0031

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8(0):726–742.

Shuming Ma, Jian Yang, Haoyang Huang, Zewen Chi, Li Dong, Dongdong Zhang, Hany Hassan Awadalla, Alexandre Muzio, Akiko Eriguchi, Saksham Singhal, Xia Song, Arul Menezes, and Furu Wei. 2020. Xlm-t: Scaling up multilingual machine translation with pretrained cross-lingual transformer encoders.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

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.

Brian Tubay and Marta R Costa-Jussa. 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: Shared Task Papers*, pages 667–670.

Pere Vergés Boncompte and Marta Ruiz Costa-Jussà. 2020. Multilingual neural machine translation: Case-study for catalan, spanish and portuguese romance languages. In *EMNLP 2020, Fifth Conference on Machine Translation: November 19-20, 2020, online: proceedings of the conference*, pages 447–450. 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.

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.

EdinSaar@WMT21: North-Germanic Low-Resource Multilingual NMT

Svetlana Tchistiakova[1], **Jesujoba O. Alabi**[2],
Koel Dutta Chowdhury[2], **Sourav Dutta**[3], **Dana Ruiter**[2]
[1]The University of Edinburgh, Edinburgh, Scotland
[2]Saarland University, Saarbrücken, Germany
[3]Technical University of Kaiserslautern, Kaiserslautern, Germany
Corresponding author: stchisti@ed.ac.uk

Abstract

We describe the EdinSaar submission to the shared task of Multilingual Low-Resource Translation for North Germanic Languages at the Sixth Conference on Machine Translation (WMT2021). We submit multilingual translation models for translations to/from Icelandic (is), Norwegian-Bokmål (nb), and Swedish (sv). We employ various experimental approaches, including multilingual pre-training, back-translation, fine-tuning, and ensembling. In most translation directions, our models outperform other submitted systems.

1 Introduction

This paper presents the neural machine translation (NMT) systems jointly submitted by The University of Edinburgh and Saarland University to the WMT2021 Multilingual Low-Resource Translation for Indo-European Languages task, describing both primary and contrastive systems which translate to/from the three North Germanic languages, Icelandic (is), Norwegian-Bokmål (nb), and Swedish (sv). Our contrastive system, submitted as "edinsaarContrastive" outperforms the other submissions across all evaluation metrics except for BLEU, for which our "edinsaarPrimary" system performs best.

Although low-resource MT has recently gained much attention, there is little prior work on North Germanic languages. We contribute to this space by experimenting with both training a multilingual system from scratch and exploiting model adaptation from a large pre-trained language model. We fine-tune our initial translation models to the target languages, and then experiment with further in-domain fine-tuning. Data is sourced from openly available data sets in accordance with the corpora allowed in the shared task. We use parallel data sets pairing our target languages with each other and with the allowed high-resource languages, and monolingual data from Wikipedia.

The rest of the paper is structured as follows: we review related work in Section 2, we introduce the methods and experimental settings including data and model architecture in Section 3, we evaluate model performance in Section 4, and, finally, we draw conclusions and suggest avenues for future work in Section 5.

2 Related Work

Recent work in NMT for North Germanic languages is limited; however, OPUS-MT (Tiedemann and Thottingal, 2020), which contains over 1,000 pre-trained, ready-to-use neural MT models including models for Danish, Norwegian, and Swedish, is a notable exception.

Due to the scarcity of parallel data for low-resource languages, recent work leverages monolingual data, including pivoting from high-resource languages (Currey and Heafield, 2019; Kim et al., 2019), and using back-translation (Sennrich et al., 2016a; Edunov et al., 2018) to generate pseudo-parallel data with synthetic sources from monolingual data. Since the little parallel data that is available often comes from noisy web crawls, parallel corpus filtering is used to develop better translation models (Koehn et al., 2020). Additional methods for boosting the performance of low-resource pairs include transfer learning from models trained on higher-resource pairs (Zoph et al., 2016; Kocmi and Bojar, 2018), and developing multilingual systems to allow models to take advantage of linguistic relatedness. Multilingual systems can employ either separate encoders or decoders for each language (Dong et al., 2015; Firat et al., 2016), or shared encoders/decoders, and can additionally make zero-shot MT possible (Johnson et al., 2017; Ha et al., 2016), while scaling to hundreds of language pairs (Aharoni et al., 2019; Fan et al., 2020). Sampling language pairs in proportion to their prevalence in the training data can ensure that all directions get enough coverage by the model (Arivazhagan et al.,

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 368–375
November 10–11, 2021.

2019; Fan et al., 2020). Further fine-tuning multilingual systems on target language directions can improve performance of low-resource pairs (Neubig and Hu, 2018; Lakew et al., 2019). Adapting a multilingual pre-trained language model to the translation task has led to improvements in translation quality (Clinchant et al., 2019; Chen et al., 2020). Finally, combining multiple MT system checkpoints together by ensembling improves performance of the final system (Sennrich et al., 2017).

3 Method

Given a set of primary languages L_p and secondary languages L_s, we train a multilingual MT system on the parallel data between all the language combinations $\{L_p, L_s\} \leftrightarrow \{L_p, L_s\}$. This is our **baseline**. We extend this approach with a combination of the following methods:

Pre-training: We initialize a base model using a highly multilingual pre-trained model, in order to transfer the learned parameters to the translation task. This is our **primary** system.
Back-translation: We use the baseline model to back-translate monolingual corpora in L_p into all other languages in L_p to obtain a training data set of back-translations D_{BT}.
Fine-tuning: We fine-tune the baseline model on the subset of languages $\{L_p, L_s\} \leftrightarrow L_p$, on both parallel and back-translated data D_{BT}. Our **contrastive** system is an ensemble of the last four checkpoints of this model.

3.1 Data

For training our models, we include data from the target primary low-resource languages, Icelandic (`is`), Norwegian-Bokmål (`nb`), and Swedish (`sv`), and the related secondary languages Danish (`da`), German (`de`), English (`en`).

We use data for all translation directions involving `da`, `de`, `en`, `is`, `nb`, `sv` from the following **parallel** corpora from Opus: Bible (Christodouloupoulos and Steedman, 2014), Books (Tiedemann, 2012), Europarl (Koehn, 2005), GlobalVoices (Tiedemann, 2012), JW300 (Agić and Vulić, 2019), MultiCCAligned (El-Kishky et al., 2020), Paracrawl (Esplà et al., 2019), TED2020 (Reimers and Gurevych, 2020), and WikiMatrix (Schwenk et al., 2019). We also use all corpora from ELRC[1] that include these directions (a total of 159 corpora, retrieved in May 2021). These corpora include all corpora allowed by the shared task, with the exception of the Opus-100 data set, which we avoided as it had many duplicate sentences with the above corpora.

We use **monolingual** data from Wikipedia for `is` and `nb` to augment our data set with back-translations (Sennrich et al., 2016a). Because the Wikipedia data for `sv` was created in large part by a bot[2] and consisted of many stub articles and tables, we use the `sv` portion of our training data as monolingual data for back-translation instead.

Our final data includes 30 language directions:

(a) $L_p \leftrightarrow L_p$: {`is`,`nb`,`sv`} $\leftrightarrow$ {`is`,`nb`,`sv`}
(b) $L_p \leftrightarrow L_s$: {`is`,`nb`,`sv`} $\leftrightarrow$ {`da`,`de`,`en`}
(c) $L_s \leftrightarrow L_s$: {`da`,`de`,`en`} $\leftrightarrow$ {`da`,`de`,`en`}
(d) $L_{p\_bt} \rightarrow L_p$: {`is`,`nb`,`sv`} $\rightarrow$ {`is`,`nb`,`sv`}

where L_{p_bt} is created from the monolingual target side back-translated data D_{BT}.

Parallel Data Filtering We filter the parallel data using rule-based heuristics borrowed from the Bifixer/Bicleaner tools (Sánchez-Cartagena et al., 2018; Ramírez-Sánchez et al., 2020) and language identification using FastText (Joulin et al., 2016, 2017). This repairs common orthographic errors, including fixing failed renderings of glyphs due to encoding errors, replacing characters from the wrong alphabet with correct ones, and un-escaping html. It also removes any translation pairs where: the pair is a duplicate, the source and target are identical, the source or target language is not the intended language, one side is more than 2x the length of the other, one side is empty, one side is longer than 5000 characters, one side is shorter than 3 words, or one side contains primarily URLs and symbols rather than text.

Filtering reduces our parallel data to 77% of its original total size. This data is then reversed in order to train our multilingual model in all translation directions, resulting in a total of 421,656,410 parallel sentence pairs in all 30 language directions. Table 1 lists the filtered data counts and the percentage of the original data that these counts represent.

Monolingual In-Domain Data Filtering The validation set provided by the shared task organizers, containing thesis abstracts and descriptions, is dissimilar to our available parallel corpora. Therefore, we filter the Wikipedia monolingual `is` and

[1] https://elrc-share.eu/

[2] https://en.wikipedia.org/wiki/Lsjbot

	de		en		is		nb		sv	
da	6921831	(48)	20604309	(77)	797806	(68)	10654	(89)	5590356	(65)
de			144890166	(80)	456054	(62)	24963	(91)	5119372	(59)
en					3766342	(78)	279370	(46)	21906032	(78)
is					351833	(60)	597	(89)	446106	(46)
nb							2943733	(44)	14247	(89)

Table 1: Number of sentences after filtering (with % of total raw data remaining after filtering) in each language direction from source (left) to target (top) from all corpora and for additional monolingual data from Wikipedia. The parallel data was mirrored in the reverse directions to create 30 total language directions for training.

nb data for similarity to this validation set to create in-domain monolingual data for use in back-translation. We identify in-domain monolingual instances in our data by calculating the cosine similarity between each sentence in a given language in the monolingual data to each of the sentences in the shared task validation data for that language. When a training instance has a similarity of $>= \theta$ with at least one validation instance, it is added to the in-domain fine-tuning corpus. We set $\theta = 0.9$ and use LASER (Artetxe and Schwenk, 2019) to extract vector representations of sentences for calculating similarity.

Validation and Test Data We split off 2000 sentence pairs from each language pair in our parallel data to use as an **internal test set**. For is-nb directions, we use the few parallel sentences available for this, meaning that no parallel data is left for the training or validation corpus. Therefore, translating between these directions is a zero-shot task for our models.

We also split off 2000 sentence pairs from each language pair in our parallel data for **internal validation**. For validation of our primary model, we use the entire collection of 2000 validation sentence pairs in each language direction. For the baseline system, we cut this down to a total of $\sim$ 2000 sentences, because performing validation is quicker on smaller data. Therefore, we use a subset of 72 validation sentences in each $\{L_p, L_s\} \leftrightarrow \{L_p, L_s\}$, except is-nb, resulting in 2016 sentences. For the contrastive model, we use the same sentences in only $\{L_p, L_s\} \leftrightarrow \{L_p\}$, to which we add 72 sentences from the back-translated data in the is-nb directions, resulting in a total of 1728 sentences.

We use the **shared task validation** set, to compare performance between our systems, and do not use it during model training or fine-tuning. We additionally report results Section 4 on the **shared task test set**, which was provided to the teams after the completion of the shared task. These test sets contain approximately 500 sentences in each language direction.

	is	nb	sv
is		2564234 (87)	10123 (99)
nb	279818 (80)		344583 (78)
sv	299277 (85)	2521823 (86)	

Table 2: Number of back-translated filtered sentences (with % of total data remaining after filtering) between synthetic source (left) to original target (top).

Back-translation We use the baseline system (Section 3.3) to create back-translations of our monolingual in-domain filtered Wikipedia data. This generates synthetic sources from is to {nb, sv} and from nb to {is, sv}. We additionally back-translate the sv side of our parallel nb-sv corpus into is and our is-sv corpus into nb. After creating the back-translations, we filter the new synthetic parallel data sets again using the parallel data filtering steps (Section 3.1), in order to remove sentences that consisted primarily of model errors or hallucinations. The final counts of filtered back-translated data are in Table 2, as well as the percentage of the original total in-domain data that these counts represent.

3.2 Byte-pair Encoding

To create a vocabulary for our baseline and contrastive systems, we train a shared byte-pair encoding (BPE) (Sennrich et al., 2016b) model using SentencePiece (Kudo and Richardson, 2018). We sample 10 million monolingual sentences from our parallel training data, based on the amount of monolingual data available for each language. Following the idea of Arivazhagan et al. (2019), we use temperature sampling, where the probability of sampling any particular data set D in language ℓ out of the n total data sets is defined as $p_\ell = (\frac{D_\ell}{\sum_i^n D_i})^{\frac{1}{T}}$, where we set $T = 5$. The goal of sampling in this way is to provide a compromise that allows the BPE model to view a larger portion of lower resource

language tokens (unlike sampling according to the original distribution would), while still providing extra space in the model for the larger variety of tokens coming from high-resource corpora (unlike sampling uniformly would). We use a vocabulary of $32,000$ tokens. When BPE-ing our training data, we use BPE-dropout (Provilkov et al., 2020) with a probability of 0.1.

3.3 Models

Baseline Our baseline system is trained on a concatenation of data sets (a), (b), and (c) (see Section 3.1). The data is pre-processed using byte-pair encoding as described in Section 3.2. Following the method of Johnson et al. (2017), we jointly train the model to translate in all our language directions, pre-pending a token `<2xx>` to the source side to inform the model which target language to translate into. The system is comprised of a transformer base model trained using Marian (Junczys-Dowmunt et al., 2018) with cross-entropy loss, following the method of (Vaswani et al., 2017) and the default Marian `transformer` configuration.

We differ from the default configuration in the following ways. We fit our mini-batch to a workspace of 6144 MB, set the learning rate to 0.0003 with a warm-up increasing linearly for 16000 batches and decaying by $\frac{16000}{\sqrt{no.\ batches}}$ afterwards. We train on multiple GPUs using Adam (Kingma and Ba, 2014) with synchronous updates for optimization, setting $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 1e-09$. We set transformer dropout between layers to 0.01. We use a maximum sentence length of 200 tokens, a maximum target length as source length factor of 2, and a label smoothing of 0.01. During validation, we use a beam size of 6 and normalize the translation score by $translation\_length^{0.6}$. We check translation quality on our internal validation set (Section 3.1) every 5000 model updates and stop training when performance doesn't improve for 15 checkpoints. The model was trained for approximately 66 hours on four NVIDIA GeForce RTX 3090 GPUs.

Contrastive Our contrastive model fine-tunes the baseline model directly, using a concatenation of all data sets that incorporate our target languages, including parallel and back-translated data (the data sets (a), (b), and (d) described in Section 3.1). The fine-tuned model uses the same architecture, training settings, and stopping criterion as the original baseline model, essentially allowing us to continue training further from the original baseline. The final submitted system is an ensemble of the last four checkpoints of this model. The model was trained for approximately 54 hours on two NVIDIA GeForce RTX 2080 TI GPUs.

Primary For the primary system, we adapt mT5 (Xue et al., 2020), a multilingual pre-trained transformer language model, to the translation task. We use mt5 because of its state-of-the-art performance and its coverage of all of our target North Germanic languages. We use the SimpleTransformers[3] framework which extends HuggingFace (Wolf et al., 2019), with the default parameters. Since our model is initialized from the parameters of the `mt5-base` system, including the embedding layers, we use the same byte-pair encoded vocabulary as the original model. Due to resource constraints, we sample a total of 100k parallel sentences from data sets (a) and (b) (described in Section 3.1). We pre-pend a string to the source side to indicate to the model which target language to translate into, and adapt the model for 5 epochs. We further fine-tune this model on data that includes our target languages (sets (a) and (b) from Section 3.1) to create our Primary system. The model was trained for approximately 46 hours on a single NVIDIA A100 SXM4 GPU.

4 Evaluation

Table 3 reports results on detokenized SacreBLEU on each of our internal test set, the shared task validation set, and the shared task test set[4]. Comparing results on the internal test set and shared task validation sets show that our models fail to generalize well to the shared task domain. The `mt5_base_ada_ft` performance drops by an average of -4.2 BLEU points between the internal test set and the shared task validation set, while the `marian_ft_esmb` model performance drops by an average of -1.0 BLEU points. Performance on the shared task test set suffers the most on the least represented languages (in particular on `is`) causing the `marian_ft_esmb` to lose an additional -1.7 average BLEU points and the `mt5_base_ada_ft` model to lose an additional -1.8 average BLEU points. In future work, we would like to experiment with different sampling

[3] https://github.com/ThilinaRajapakse/simpletransformers

[4] BLEU+case.mixed+numrefs.1+smooth.exp +tok.13a+version.1.4.14

	Model	is → nb	is → sv	nb → is	nb → sv	sv → is	sv → nb	Avg.
Internal test	marian	12.5	33.3	11.8	26.7	27.8	18.7	21.8
	marian_ft	19.1	41.7	16.1	31.6	38.4	30.3	29.5
	marian_ft_esmb	19.3	42.2	16.4	31.6	39.2	30.3	29.8
	mT5_base_ada	23.1	42.3	19.4	33.7	42.8	33.9	32.5
	mT5_base_ada_ft	**26.5**	**42.9**	**20.0**	**33.9**	**43.3**	**34.2**	**33.5**
Shared valid	marian	10.9	13.5	15.1	41.3	12.2	24.9	19.7
	marian_ft	13.0	18.0	22.9	50.0	19.4	45.9	28.2
	marian_ft_esmb	13.9	18.2	23.6	**50.6**	20.1	**46.7**	28.8
	mT5_base_ada	14.6	**19.2**	25.8	46.6	20.6	43.2	28.3
	mT5_base_ada_ft	**17.4**	18.7	**26.5**	47.9	**20.8**	44.2	**29.3**
Shared test	marian_ft_esmb	13.0	17.3	18.3	**45.4**	20.2	**48.2**	27.1
	marian_base_ada_ft	**16.3**	**18.8**	**19.5**	42.9	**22.4**	45.4	**27.5**

Table 3: SacreBLEU (detokenized) results on the internal test set and the shared task validation and test sets.

methods to boost the performance of the least represented directions.

Comparing results between models, our primary `mt5_base_ada` system outperforms the `marian` model trained from scratch by an average of +10.7 and +8.6 BLEU points on the internal and shared task validation sets, respectively. The further fine-tuned variant `mt5_base_ada_ft` leads to an additional average improvement of just under +1 BLEU point on both sets, showing that the mt5 model already learned a good amount about our target task and languages from our initial adaptation step. The `marian` model is also outperformed by the fine-tuned variant `marian_ft`, resulting in an average improvement of +7.7 BLEU points on the internal test set and +8.5 BLEU points on the shared task validation set.

Both the `mt5_base_ada_ft` and `marian_ft` models are exposed to similar language data; however, the mt5 language model we adapted from (`mt5-base`) is much larger than our `marian` model (580 million vs 44 million parameters), and was trained on more language data (750 GB vs 46 GB), so it had a much stronger base to start from. Ensembling the last 4 checkpoints of the fine-tuned marian model for `marian_ft_esmb` boosts performance by +0.3 and +0.6 average BLEU on the internal and shared task validation sets over `marian_ft`; however, the `mt5_base_ada_ft` model still outperforms the `marian_ft_esmb` model by +3.7 and +0.5 average BLEU on the internal test set and the shared task validation set, respectively. Therefore, we submitted the `mt5_base_ada_ft` model as our primary system to the shared task; however, our contrastive system, the `marian_ft_esmb` model, won in the shared task rankings.

In the global automated evaluations of the shared task, our contrastive system is the best-performing submitted system[5], outperforming the official mT5 baseline by approximately +8.5 BLEU. We hypothesize that the mt5 baseline, while being pre-trained on massive amounts of partially noisy monolingual data, has learned the translation task via training on the development set only, so it has less informative parallel data available than our models. The M2M-100 (Fan et al., 2020) baseline outperforms all submitted systems, despite having been trained on noisy parallel data only. We hypothesize that the highly-multilingual nature of the M2M-100 model allows the target languages to benefit from the supervisory signals between related language combinations.

5 Conclusion and Future Work

We contribute to the growing space of NMT for North Germanic languages. We explore multilingualism by training a transformer with a shared encoder and decoder for all language pairs from scratch, as well as adapting a pre-trained multilingual language model. Fine-tuning these models to our low-resource language pairs was a key component in our success in the task, and we additionally confirm that employing popular techniques in machine translation, such as data filtering, back-translation, and model ensembling are beneficial for improving performance on low-resource directions. In future work, we would like to experiment with fine-tuning additional pre-trained models such as the M2M-100, incorporating iterative back-translation, and trying different sampling methods during training to boost lower performing low-resource language pairs.

[5]Only our primary model was submitted for manual evaluation, where it outranked the other submissions. Official rankings are available at: http://statmt.org/wmt21/multilingualHeritage-translation-task.html

Acknowledgements

The authors thank the University of Edinburgh and the German Research Center for Artificial Intelligence (DFKI GmbH) for providing the necessary infrastructure and resources to run the experiments.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online. Association for Computational Linguistics.

Christos Christodouloupoulos and Mark Steedman. 2014. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of BERT for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 108–117, Hong Kong. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 99–107, Hong Kong. Association for Computational Linguistics.

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. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint*.

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.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

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*, pages 427–431, Valencia, Spain. 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.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-English languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 866–876, Hong Kong, China. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Surafel Melaku Lakew, Alina Karakanta, Marcello Federico, Matteo Negri, and Marco Turchi. 2019. Adapting multilingual neural machine translation to unseen languages. *CoRR*, abs/1910.13998.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. 2018. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

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*, 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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480.

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*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

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.

TenTrans Multilingual Low-Resource Translation System for WMT21 Indo-European Languages Task

Han Yang[1,2]†, Bojie Hu[2], Wanying Xie[2], Ambyer Han[2], Pan Liu[2], Jinan Xu[1]*, Qi Ju[2]*
[1]Beijing Jiaotong University
[2]TencentMT Oteam
{19120421, jaxu}@bjtu.edu.cn
{bojiehu, wanyingxie, ambyera, galeliu, damonju}@tencent.com

Abstract

This paper describes TenTrans' submission to WMT21 Multilingual Low-Resource Translation shared task for the Romance language pairs. This task focuses on improving translation quality from Catalan to Occitan, Romanian and Italian, with the assistance of related high-resource languages. We mainly utilize back-translation, pivot-based methods, multilingual models, pre-trained model fine-tuning, and in-domain knowledge transfer to improve the translation quality. On the test set, our best-submitted system achieves an average of 43.45 case-sensitive BLEU scores across all low-resource pairs. Our data, code, and pre-trained models used in this work are available in TenTrans evaluation examples[1].

1 Introduction

We participate in the WMT21 Multilingual Low-Resource Translation shared task. This task focuses on the multilinguality in the cultural heritage domain for two Indo-European language families: North-Germanic and Romance. We devote the research into translations among Romance languages, including Catalan→Occitan, Catalan→Romanian, Catalan→Italian. Additionally, this task explicitly encourages the use of data of four related high-resource languages (Spanish, French, Portuguese and English) in the same linguistic family.

For the model architecture, we adopt a universal encoder-decoder architecture that shares parameters across all languages (Johnson et al., 2017). And almost all of the subsequent experiments are based on Transformer *base* model (Vaswani et al., 2017).

To effectively exploit low and high resource data in the multilingual low-resource scenario, we explore several approaches, and each approach shows effectiveness. We employ back-translation (Sennrich et al., 2016a) and pivot-based methods to augment the training corpus. In terms of knowledge transfer, we explore the pre-trained model and the multilingual model that trained with both low and high resource language pairs. Moreover, we extract in-domain corpus by a domain classifier and adapt the model to the target domain by in-domain fine-tuning.

This paper is structured as follows: Section 2 introduces used datasets, data statistic and pre-processing pipeline. Section 3 describes the details of different approaches. In Section 4 we present experimental settings and results. Section 5 draws a brief conclusion of our work in the WMT21.

2 Data

2.1 Datasets

The training datasets are majorly provided by the publicly available OPUS (Tiedemann, 2012) repository. We use almost all available datasets provided in the task, including Europarl, JW300, WikiMatrix, MultiCCAligned, OPUS-100, Bible, ELRC, and 167.2K It-Ro pairs in TED talks as well as 15M/360K sentence pairs of En-It/En-Ro extracted from Wikipedia dumps. For datasets that can be found through the resources search form on the top-level website of OPUS, we use opus-tools[2] to extract low-resource language pairs. As for rest of the data, we download them in the usual way. Statistics of different datasets are showed in Table 1.

2.2 Data Pre-processing

Cleaning datasets is necessary when the datasets are noisy and of low quality. We partially refer to

†This work is done by the author as an intern at TencentMT Oteam.
*Corresponding author.
[1]https://github.com/TenTrans/TenTrans/blob/master/example/WMT21/WMT21-low-resource-MNMT.md
[2]https://pypi.org/project/opustools-pkg/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 376–382
November 10–11, 2021.

Bilingual	WikiMatrix	MultiCCAligned	Bible/Europarl	JW300	ELRC	OPUS
LRL-LRL	2.7M	11.5M	386.5K	1.2M	0.022K	-
HRL-LRL	8.9M	68.7M	5.9M	12.2M	2.7M	2.0M
Monolingual	**WikiMatrix**	**MultiCCAligned**	**Bible/Europarl**	**JW300**	**ELRC**	**OPUS**
Oc	342.3K	-	-	-	-	35.8K
Ro	3.8M	132.6M	470.1K	54.5M	1.1M	1M
It	9.1M	175.2M	23.3M	66.2M	1.6M	~~1M~~

Table 1: Number of sentences in different datasets. 'LRL-LRL' means the bilingual data between low resource languages, e.g. Ca-Ro. 'HRL-LRL' means the bilingual data between high-low resource languages, e.g. En-Ro. '~~1M~~' means we do not use that data though it is provided. Note that OPUS provides En-Oc/Ro/It bilingual pairs, but we also use the target side Oc/Ro as monolingual data due to lacking data.

	Ca-Oc	Ca-Ro	Ca-It
No filter	138.7K	2.2M	6.3M
Filtered	138.7K	2.1M	5.8M
	It-Ro	**It-Oc**	**Oc-Ro**
No filter	7.2M	122K	81K
Filtered	6.9M	122K	81K

Table 2: Number of sentences in low-resource bilingual data.

M2M-100[3] (Fan et al., 2020) data pre-processing procedures to filter bilingual sentences. We remove sentences with more than 50% punctuation, deduplicate training data and remove all instances of evaluation data from the bilingual training data.

We tokenize all data and normalize punctuation with the Moses tokenizer (Koehn et al., 2007). To enable open-vocabulary and share information among languages, we use joint Byte-Pair-Encoding (BPE) with 32K split operations for subword segmentation (Sennrich et al., 2016b). We also remove sentences longer than 512 as well as sentence pairs with a source/target length ratio exceeding 3.

For monolingual data, we still employ those rules except the length ratio filter. See Table 2 for the statistics of low-resource bilingual data, Table 3 for the statistics of high-low resource bilingual data and Table 4 for the statistics of low-resource monolingual data.

3 System Overview

3.1 Base Systems

In multilingual translation scenarios, one can employ multi-task learning framework using multiple encoders or multiple decoders (Luong et al., 2016; Dong et al., 2015; Firat et al., 2016). Either, one can employ a unified model consisting of a shared encoder and a shared decoder for all the language pairs (Johnson et al., 2017). We experiment with these two models and conduct the conclusion that a universal encoder-decoder model outperforms the model with multiple decoders. The unified architecture is adopt in subsequent experiments in this work. Parameters and vocabulary are shared among all language pairs and this helps the generalization across languages improving the translation for the low-resource language pairs (Aharoni et al., 2019). We also train three separate bilingual models to be regarded as contrastive model with multilingual model. Furthermore, we jointly train on Catalan, Occitan, Romanian, Italian four low-resource languages simultaneously to obtain a many-to-many multilingual model. Detailed results of base systems are shown in Table 6.

We use the Transformer (Vaswani et al., 2017) as our model architecture for all of our systems. We experiment with increasing network capacity but we find that deep and wide model architectures bring training hurdles. So almost all subsequent models are based on the Transformer *base* architecture (Vaswani et al., 2017) as implemented in TenTrans[4], expect for pre-trained model M2M-100 trained using FAIRSEQ[5] (Ott et al., 2019).

3.2 Back-translation

Back-translation (briefly, BT) (Sennrich et al., 2016a) is an effective and commonly used data augmentation technique to incorporate monolingual data into a translation system.

In this work, for translation direction with more than 5 million bilingual data such as Catalan→Italian, we train a dedicated bilingual BT

[3] https://github.com/pytorch/fairseq/tree/master/examples/m2m_100

[4] https://github.com/TenTrans/TenTrans

[5] https://github.com/pytorch/fairseq

		It	Oc	Ro	Ca
En	No filter	22.4M	73K	14.6M	7.1M
	Filtered	22.3M	59K	14.5M	7.0M
Es	No filter	4.4M	36K	6.4M	12.3M
	Filtered	4.3M	36K	4.2M	6.5M
Fr	No filter	4.8M	124K	1.6M	7.7M
	Filtered	4.7M	124K	1.5M	7.0M
Pt	No filter	24.3M	24K	5.7M	4.9M
	Filtered	15.6M	24K	5.6M	4.6M

Table 3: Number of sentences in high-low resource bilingual data.

	It	Oc	Ro
No filter	275M	378K	193.5M
Filtered	38.3M	225K	13.4M

Table 4: Number of sentences in low-resource monolingual data.

BT System	It-Ca	Oc-Ca	Ro-Ca
Bilingual model	37.74	-	-
Multilingual 4-to-4	31.41	23.72	51.02

Table 5: BLEU scores (%) for reverse models evaluated on the validation data.

model Italian→Catalan to translate Italian monolingual data into Catalan. For other translation directions with less than 5 million bilingual data, we use the jointly pre-trained many-to-many multilingual model with four low-resource languages as its source and target side (see Section 3.1) to back translate Occitan and Romanian monolingual data into Catalan. Beam search with beam size 5 is used when generating the synthetic sentences. Detailed results of reverse models are shown in Table 5.

3.3 Multilingual Model

Arivazhagan et al. (2019) shows that multilingual models can improve the translation performance of medium and low resource languages, as multilingual models are often trained on greater quantities of data compared to individual models. So we utilize high-low resource paired data such as English→Occitan in addition to low-resource bilingual data during training. Training on high-resource and low-resource language pairs together may bring knowledge transfer (Zoph et al., 2016), especially when languages are from the same linguistic family.

In the experiment, we train on four high-resource languages (Spanish, French, Portuguese and English) combined with four target-task low-resource languages together, resulting in an **8-to-4 multilingual** model with Ca, Oc, Ro, It as the target side. We randomly extract 2K sentence pairs from training data as the validation set for each high-low resource languages pairs. BPE codes and multilingual vocabulary are shared among all languages, but a shared multilingual vocabulary runs the risk of favoring high-resource languages over others, due to the imbalance of the dataset size the vocabulary is extracted. To reduce the effect of imbalanced dataset size, we apply a temperature sampling strategy named Vocabulary Sampling to construct a joined vocabulary. Following Arivazhagan et al. (2019), we set sampling temperature $\underline{T} = 5$.

Table 6 shows results on validation set of our baseline systems. Obviously, the universal encoder-decoder model outperforms the model with separate decoders for each target language by 7 BLEU on average. Compared to the bilingual baseline system, our universal multilingual 1-to-3 baseline system performs great improvement on low-resource languages, at the cost of sacrificing performance on relatively rich language Italian. However, the jointly trained multilingual 4-to-4 system shows performance degradation. We ascribe this phenomenon to multilingual model capacity is split for more translation directions, from 3 directions to 12 translation directions in this case.

3.4 Pivot-based Method

Pivot-based approaches are prevalent when addressing the data scarcity problem in machine translation, nonetheless, they suffer from cascaded translation errors: the mistakes made in the source-to-pivot translation will be propagated to the pivot-to-target translation (Dabre et al., 2020). Another pivot-based approach used in zero-resource transla-

Base System	Ca-Oc	Ca-Ro	Ca-It	Average BLEU
Bilingual models	30.02	-	-	28.48
	-	21.51	-	
	-	-	33.91	
Separate Decoders				
Multilingual 1-to-3	25.82	22.03	32.96	26.90
Universal Models				
Multilingual 1-to-3	43.40	24.16	32.92	**33.78**
Multilingual 4-to-4	41.44	22.81	31.01	31.75

Table 6: BLEU scores (%) for baseline systems evaluated in the validation data. And the numbers represent languages used in models, e.g. 1-to-3 means source side of model is Ca but target side consists of Oc, Ro, It, and 4-to-4 means both the source and target side of model consist of four languages Ca, Oc, Ro and It.

tion scenario is that the pivot side of the pivot-target parallel corpus is back-translated to the source language, creating a synthetic source-target parallel corpus (Lakew et al., 2018; Gu et al., 2019). In this work, we adopt the latter pivot-based method.

In practice, we consider four high-resource languages En, Es, Fr, Pt as pivot languages, thus we train a pivot-to-source multilingual model to back translate four pivot languages in pivot-to-target parallel data into source language. Owing to relatively rich data of Catalan-Italian, we only perform experiments on low-resource languages of Occitan and Romanian. To balance distribution between genuine parallel data and synthetic parallel data, we oversample genuine data to be of the same magnitude as synthetic data.

We can combine all synthetic parallel data generated from back-translation and pivot-based method with genuine parallel data to jointly train a multilingual model from scratch, which is named **Combine-All**. Source side of this model is comprised of four rich-resource and four low-resource languages, and target side of this model is comprised of four low-resource languages.

3.5 Pre-trained Model Fine-tuning

Because of the recent popularity of using large scale pre-training models to fine-tune specific languages and tasks, we employ the M2M-100, a true Many-to-Many multilingual translation model (Fan et al., 2020) that can translate between 100 languages which cover four task languages. Our experiments are based on the M2M-100 1.2B model due to its better performance than the 418M model. In the subsequent fine-tuning procedure, we follow the parameters setting in fine-tuning mBART (Liu et al., 2020). In three task directions, we try fine-tuning M2M-100 model with genuine bilingual data (Bilingual FT) and fine-tuning with genuine multilingual data (Multilingual FT). Moreover, we try fine-tuning the M2M-100 1.2B model using **Combine-All** data with four high-resource plus low-resource languages as the source side and four low-resource languages as the target side.

Unfortunately, M2M-100 model trains on SentencePiece(Kudo and Richardson, 2018) rather than Byte-Pair-Encoding so that the fine-tuned model can not be directly combined with the models that listed above for ensembling. We utilize synthetic Catalan-Occitan, Catalan-Romanian data generated through sentence-level knowledge distillation (Kim and Rush, 2016) to train a 'student' model so as to incorporate knowledge of 'teacher' model M2M-100 1.2B into 'student' model. Concretely, in Catalan→Occitan direction, we employ multilingual fine-tuning on M2M-100 1.2B model using Combine-All data for 200K updates (1.1M updates for each epoch), after that, we continue with bilingual fine-tuning using genuine Catalan-Occitan parallel data. As for Catalan→Romanian direction, we directly use the pre-trained model without fine-tuning. We continue to train on **8-to-4 multilingual** model (See Section 3.3) in three task translation directions with data obtained through knowledge distillation and finally get a new model named **M2M-KD**. We do not implement knowledge distillation in Catalan→Italian direction since we find other systems perform equivalently to the pre-trained model. If time permitted, we believe that more improvements will be observed.

3.6 Domain Adaptation

Domains of training data are various, whereas validation and hidden test data belong to the cultural

System	Ca-Oc	Ca-Ro	Ca-It	Average BLEU
Multilingual baseline	43.40	24.16	32.92	**33.78**
+ Back-translation	48.21	22.66	32.9	34.59
+ Pivot	26.98	26.33	34.2	29.17
Combine-All	26.75	29.59	37.49	31.28
M2M-100 418M				
w/o FT	31.04	26.72	34.18	30.65
+ Multilingual FT	40.71	25.26	33.92	33.30
+ Bilingual FT	49.42 - -	- 25.4 -	- - 35.19	36.67
M2M-100 1.2B				
w/o FT	34.70	32.21	38.37	35.09
+ Multilingual FT	42.09	28.11	36.62	35.61
+ Bilingual FT	49.79 - -	- 27.83 -	- - 37.09	38.24
+ Combine-All FT	37.15	30.54	37.28	34.99
+ Bilingual FT	49.86	-	-	-
Multilingual 8-to-4	51.49	29.11	38.26	39.62
+ In-domain-FT	56.60	28.30	38.74	41.21
+ M2M-KD	**65.18**	**32.85**	36.19	44.74
Ensemble				
†In-domain-FT + M2M-KD	64.70	32.85	39.41	45.65
*In-domain-FT + M2M-KD + Combine-All	64.02	32.63	**40.04**	45.56

Table 7: BLEU scores (%) for different systems on the validation data. The number 8 means source side of model consists of both four high-resource languages and four low-resource languages, 4 means target side of model consists of four low-resource languages Ca, Oc, Ro and It. '†' is the submitted primary system. '*' is the submitted contrastive system.

heritage domain. Owing to the domain discrepancy, adapting models to the cultural heritage domain (Luong et al., 2015) is required.

Due to the scarcity of in-domain data, we utilize pre-trained language model multilingual Bert [6] (Devlin et al., 2019) to train a domain classifier for extracting in-domain sentences from genuine bilingual data. To train the domain classifier, we consider validation data of three languages Ca, Ro, It as positive samples, and randomly sample the low-resource side of high-low resource bilingual data as negative samples. Then classifier is exploited to score the source sentences (Ca/Ro/It). We select sentence pairs whose source is predicted to be positive with a probability greater than threshold 0.7 to construct in-domain corpus. In the end, we pick out 60K Catalan-Occitan, 297K Catalan-Romanian and 815K Catalan-Italian data respectively as in-domain corpus. We fine-tune **8-to-4 multilingual** model on the in-domain corpus in three task translation directions and then get the **In-domain-FT** model. For the purpose of preventing overfitting, we set the max-tokens to be 2K with a learning rate of 3e-5 and we force fine-tuning to stop when finishing the first epoch. Note that we do not perform fine-tuning on the validation set.

[6] https://huggingface.co/bert-base-multilingual-cased

4 Experiments

4.1 Settings

Except that the pre-training experiments are trained on 4 NVIDIA V100 GPUs, the rest of our experiments are carried out with 8 NVIDIA P40 GPUs. Except for the pre-training experiments, the rest of our experiments use the following settings. Our models apply Adam (Kingma and Ba, 2015) as optimizer to update the parameters with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We set the label smoothing and dropout rate to 0.1. The initial learning rate is set to 5e-4

varied under a warm-up strategy with 4000 steps. In the training stage, batch size is 8K tokens per GPU.

We use uncased BLEU scores calculated with Moses multi-bleu.pl[7] toolkit as the evaluation metric. And we choose model checkpoints based on the BLEU score on average of the validation set.

4.2 Main Results

Table 7 shows that the translation quality is largely improved with different systems. Although minority systems encounter the problem of average performance degradation on the validation set, they contribute to at least one translation direction. Back-translation gives a solid improvement by nearly 0.8 BLEU on average. Pivot-based method offers 1∼2 BLEU in Catalan→Romanian, Catalan→Italian directions, however, pivot degrades in Catalan→Occitan direction. When we train an 8-to-4 multilingual model jointly with both the high and low resource languages, the model shows an absolute improvement in three task directions of 6 BLEU on average score. It can be explained by that a larger quantity of genuine data leads to robust encoder/decoder or knowledge can be transferred from high-resource into low-resource languages. As for the pre-trained model, we notice that M2M-100 1.2B model performs very well in Catalan→Romanian, Catalan→Italian directions without fine-tuning. And we find that average bilingual fine-tuning outperforms multilingual fine-tuning by about 2.6 BLEU. We also observe some systems hold a comparable performance with M2M-100 1.2B model in Catalan→Romanian and Catalan→Italian directions when training data is abundant.

Further experiments include the in-domain fine-tuning and M2M-KD based on the **multilingual 8-to-4** system. In-domain fine-tuning is restricted to in-domain data size, but we also obtain a solid improvement of 1.5 BLEU on average, especially in Catalan→Occitan direction. M2M-KD model yields a greater improvement that we get the best BLEU in Catalan→Occitan, Catalan→Romanian directions with 65.18, 32.85 respectively. Ultimately, to take advantages of multiple single models, two or three top performing models are ensembled to be the submitted systems.

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

5 Conclusions

In this paper, we present the system TenTrans submitted for the WMT21 Multilingual Low-Resource Translation for Indo-European Languages shared task. We focus on Romance languages, translating from Catalan to Occitan, Romanian and Italian. Back-translation, pivot-based method, multilingual model, knowledge distillation using pre-trained model, domain adaptation and ensembles are employed and proven effective in the experiments. Our best submitted system achieves an average of 43.45 case-sensitive BLEU score across all low-resource languages pairs.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3874–3884. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation. *CoRR*, abs/2001.01115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek,

Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. *CoRR*, abs/1906.01181.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Surafel Melaku Lakew, Quintino F. Lotito, Matteo Negri, Marco Turchi, and Marcello Federico. 2018. Improving zero-shot translation of low-resource languages. *CoRR*, abs/1811.01389.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Minh-Thang Luong, Christopher D Manning, et al. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the international workshop on spoken language translation*, IWSLT. Da Nang, Vietnam.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer 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, Istanbul, Turkey, May 23-25, 2012*, pages 2214–2218. 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575. The Association for Computational Linguistics.

The University of Maryland, College Park Submission to Large-Scale Multilingual Shared Task at WMT 2021

Saptarashmi Bandyopadhyay* **Tasnim Kabir*** **Zizhen Lian*** **Marine Carpuat**
Department of Computer Science
University of Maryland, College Park
{saptab1,tkabir1,zizhlian,marine}@umd.edu

Abstract

This paper describes the system submitted to Large-Scale Multilingual Shared Task (Small Task #2) at WMT 2021. It is based on the massively multilingual open-source model FLORES101_MM100 model, with selective fine-tuning. Our best-performing system reported a 15.72 average BLEU score for the task.

1 Introduction

Massively multilingual models such as Facebook's M2M-100 (Fan et al., 2020) model provide an attractive approach to scaling Machine Translation to many language pairs by sharing encoder-decoder parameters across languages. By not centering English in its training process, M2M-100 improves translation quality substantially (by over 10 BLEU points) compared to the best single systems of WMT before 2020 on the "large-scale Many-to-Many dataset for 100 languages" (Fan et al., 2020). However, translation quality for low-resource languages still leaves much room for improvement.

We address the Large-Scale Multilingual Machine Translation Shared Task (Small Track #2) at WMT 2021, by fine-tuning the FLORES101_MM100 model for the languages in the Shared task. We consider different fine-tuning configurations, with a goal to minimize the computational and data resources required. First, we consider the impact of finetuning on datasets of different sizes, and surprisingly show that finetuning with the smaller dataset gives better performance for some language pairs. Second, we consider selectively dropping layers during fine-tuning to reduce the computational cost of working with a Transformer model with millions of parameters. We adopt a structure dropout technique, *LayerDrop*, which has been shown to have a regularization effect and to effectively reduce model size for inference (Fan et al., 2019), as well as to reduce training time while preserving decoding quality (Zhang and He, 2020). We have used *LayerDrop* so that our model can run on large datasets for low resource language pairs.

Our best performing system is fine-tuned on the large MultiCCAligned training data and yields a sentence-piece BLEU score (the official Shared task metric) of 15.72 on the Shared task test set. However, a model fine-tuned on smaller amounts of data (bible-uedin) approaches that result, with a BLEU score of 15.10. This paper describes the submitted models, as well as experiments with *LayerDrop* configuration, which show that dropping the top layers does not help BLEU.

2 Shared Task Data

Training Our training data is provided by the Shared task organizers and is drawn from the publicly available open-source multilingual parallel corpus (OPUS) data repository for the languages of the Shared task (Tiedemann, 2012). It consists of the MultiCCAligned large dataset which supports 112 languages (El-Kishky et al., 2020) with English as the pivot language. The bible-uedin dataset (Christodouloupoulos and Steedman, 2015) is comparatively smaller than the MultiCCAligned dataset and is supported by 102 languages based on translations from the Bible. Table 1 reflects the statistics for the datasets from 23 different language pairs (3 from bible-uedin with a size of 125 MB and 15 from MultiCCAligned with a size of 16 GB) considering only the 6 languages in the Shared task which are Indonesian, Javanese, Tamil, Tagalog, Malay, and English. MultiCCAligned takes up more than 50% of the dataset while bible-uedin takes less than 0.2%.

We preprocess the data using the "SentencePiece" module (Kudo and Richardson, 2018) for tokenization and byte-pair encoding, and remove duplicate samples.

*These authors contributed equally to this work

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 383–386

Evaluation Sets The Shared task evaluates models on three distinct datasets: *dev*, *devtest* and *test*. They are all drawn the FLORES-101 benchmark for Many-to-Many multilingual translation (Goyal et al., 2021). It consists of 3001 sentences extracted from English Wikipedia and covering a variety of different topics and domains. These sentences have been translated into 101 languages by professional translators through a carefully controlled process. The Shared task uses a subset of six languages including English from FLORES-101. The languages are: Javanese (jav), Indonesian (ind), Malay(msa), Tagalog (tgl), Tamil (tam), and English (eng). The *dev* and *devtest* sets are both 2.8MB in size. These datasets were evaluation test set and were therefore held out from our fine-tuning experiments. The *test* set were inaccessible to the Shared Task participants.

source	lang_pair	lang	#lines	#words
bible-uedin	id-tl	id	29686	629304
		tl	29686	792379
	en-tl	en	62195	1550443
		tl	62195	1650384
	en-id	id	59363	1258405
		en	59363	1491576
MultiCCAligned	en-id	en	27005411	229031867
		id	27005411	219942614
	en-jv	jv	1513975	6736011
		en	1513975	6751212
	en-ms	en	5391811	75761505
		ms	5391811	73624832
	en-ta	en	880568	13561100
		ta	880568	11021555
	en-tl	tl	6593254	46368945
		en	6593254	45388545
	id-jv	id	756823	3144256
		jv	756823	3084732
	id-ms	id	2790866	37035615
		ms	2790866	38179211
	id-ta	id	406980	5326520
		ta	406980	4765008
	id-tl	tl	2673325	19793654
		id	2673325	17573455
	jv-ta	ta	64693	346766
		jv	64693	369599
	jv-ms	jv	431117	1909419
		ms	431117	2071297
	jv-tl	jv	814883	2747948
		tl	814883	2808677
	ms-ta	ms	260338	4340844
		ta	260338	3698516
	ms-tl	ms	1341969	12229073
		tl	1341969	13992119
	ta-tl	ta	557855	4203473
		tl	557855	5581043

Table 1: Split of the training datasets

3 Model Configurations

This section describes our base model and the various fine-tuning configurations considered.

3.1 Base Model

Figure 1 shows a FLORES101_MM100 model with the original encoder and decoder.

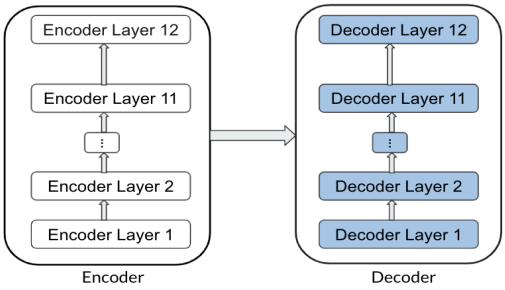


Figure 1: Baseline FLORES101_MM100 architecture

3.2 Finetuning Strategies

Hyper-parameters Table 2 gives the list of the hyper-parameter settings we use for all finetuning in our experiments. Since batch size and learning rate affect finetuning, we experimented with two different learning rates, $3e^{-5}$ and $3e^{-7}$, on the smaller dataset (bible-uedin). Changing the learning rate from $3e^{-5}$ to $3e^{-7}$ boosts the BLEU score of bible-uedin fine-tuned model to 15.10.

Batch Size	4
Loss Function	Label Smoothed Cross Entropy
Label Smoothing	0.2
Optimizer	Adagrad
Learning Rate	$3e^{-5}$ / $3e^{-7}$
LR Scheduler	Inverse Square Root
Warmup updates	2500
Dropout	0.3
Attention Dropout	0.1

Table 2: Hyperparameter setup

Data We compare the impact of using each of the datasets decribed in Section 2 to fine-tune the models: bible-uedin and MultiCCAligned.

Activation function In addition to using the standard ReLu activation function, we experiment with the GELU nonlinearity, which weighs inputs by their percentile, rather than gates inputs by their sign as in ReLUs. Compared to ReLU or leaky ReLU, GELU has the theoretical advantage of being differentiable for all values of x.

LayerDrop Fan et al. (2019) introduced a *LayerDrop* technique to generate shallow models from larger ones by dropping entire layers at inference time. These dropped layers have a regularization effect and reduce training time. Inspired by these results, we fine-tune our model with *LayerDrop*

Dataset	Dev	DevTest	Test
Baseline	12.39	11.78	12.11
Fine-tuned Models			
Bible-uedin	15.50	14.89	15.10
MultiCCAligned	**16.05**	**15.45**	**15.72**

Table 3: Impact of fine-tuning data on spBLEU: MultiCCAligned data yields the best scores, but Bible-uedin achieves close results despite being much smaller.

		Dev	DevTest	Test
Baseline		12.39	11.78	12.11
GeLu	no LD	15.19	14.61	14.83
ReLu	LD	7.35	6.94	7.34
ReLu	no LD	**16.05**	**15.45**	**15.72**

Table 4: Impact of activation function and LayerDrop (LD) on spBLEU: the standard settings with ReLu and without LD yield the best translation quality.

by selectively dropping the last three layers (9, 10, 11) in the encoder and the decoder. We compare this approach to fine-tuning all layers in our model without *LayerDrop*.

4 Results

Aggregate Results The Shared task evaluates the performance of models using a sentence-piece BLEU (spBLEU) score, aggregated across all language pairs tested. We report results using this metric and to it as BLEU in this section.

Table 3 reports the BLEU score of our models finetuned with the different datasets on the three Shared task evaluation sets. From Table 3, we can see that the model finetuned with MultiCCAligned obtains higher BLEU scores across the board compared to the model finetuned with bible-uedin. On the *test* set, it obtains a BLEU score of 15.72. However, the model fine-tuned on bible-uedin, is only about .6 BLEU point behind (15.10 BLEU), despite being only about $\frac{1}{340}$ in size comparing to the MultiCCAligned. These results suggest that amount of data is not the most important factor when selecting a dataset for fine-tuning.

Table 4 shows the BLEU scores obtained with fine-tuning configurations which vary in the activation function used and in the use of the *LayerDrop* technique for reducing model size. The best results are obtained with the standard settings: fine-tuning with the ReLU activation and no *LayerDrop*. *LayerDrop* degrades translation quality substantially, which suggests that it is not a promising strategy to reduce the computational cost of neural MT.

Per Language Results In addition to aggregate results, we report BLEU scores per language pair in Figure 2 for each of the main experimental conditions considered. Since our main motivation is to improve the performance of the model for low-resource languages, we would like to fill the gap between the languages with a higher score and the languages with a lower score, i.e. to see more dark blue squares in the Figure. Comparing the score break down of the MultiCCAligned model and the bible-uedin model, the latter one performs better on almost all translations to Tamil and Tagalog; for example, there is a 2.33 improvement on eng-tam and a 6.49 improvement on eng-tgl. Some translations from Tamil also show improvements, 1.3 on tam-eng, while the only improvement from Tagalog is 1.31 on tgl-tam. However, bible-uedin model performs worse on 19 out of all 30 language pairs and has a lower average.

5 Submitted System Configuration

The submitted system is fine-tuned with the MultiCCAligned dataset for all the language pairs mentioned in Table 1. The hyper-parameters are set as described in Table 2 with learning rate $3e^{-05}$. This system uses ReLu as the activation function and keeps all the original layers in the encoder and decoder. The fine-tuning is done for 10 epochs.

6 Conclusion

We described the University of Maryland submission to the Large-Scale Multilingual Shared Task (Small Task #2) at WMT 2021. We considered several fine-tuning configurations on top of the massively multilingual FLORES101_MM100, and find that using MultiCCAligned data and a standard model configuration give the best result. We also show that finetuning on the much smaller Bible-uedin dataset approaches our best result, with a BLEU score of 15.10. Selecting appropriate fine-tuning data thus plays a significant role in the quality of the final model, and the amount of data alone is a suboptimal selection criterion. Dropping the last three layers of the encoder and decoder decreased the translation quality. Future work is needed to determine how to reduce the computational needs of large-scale multilingual MT.

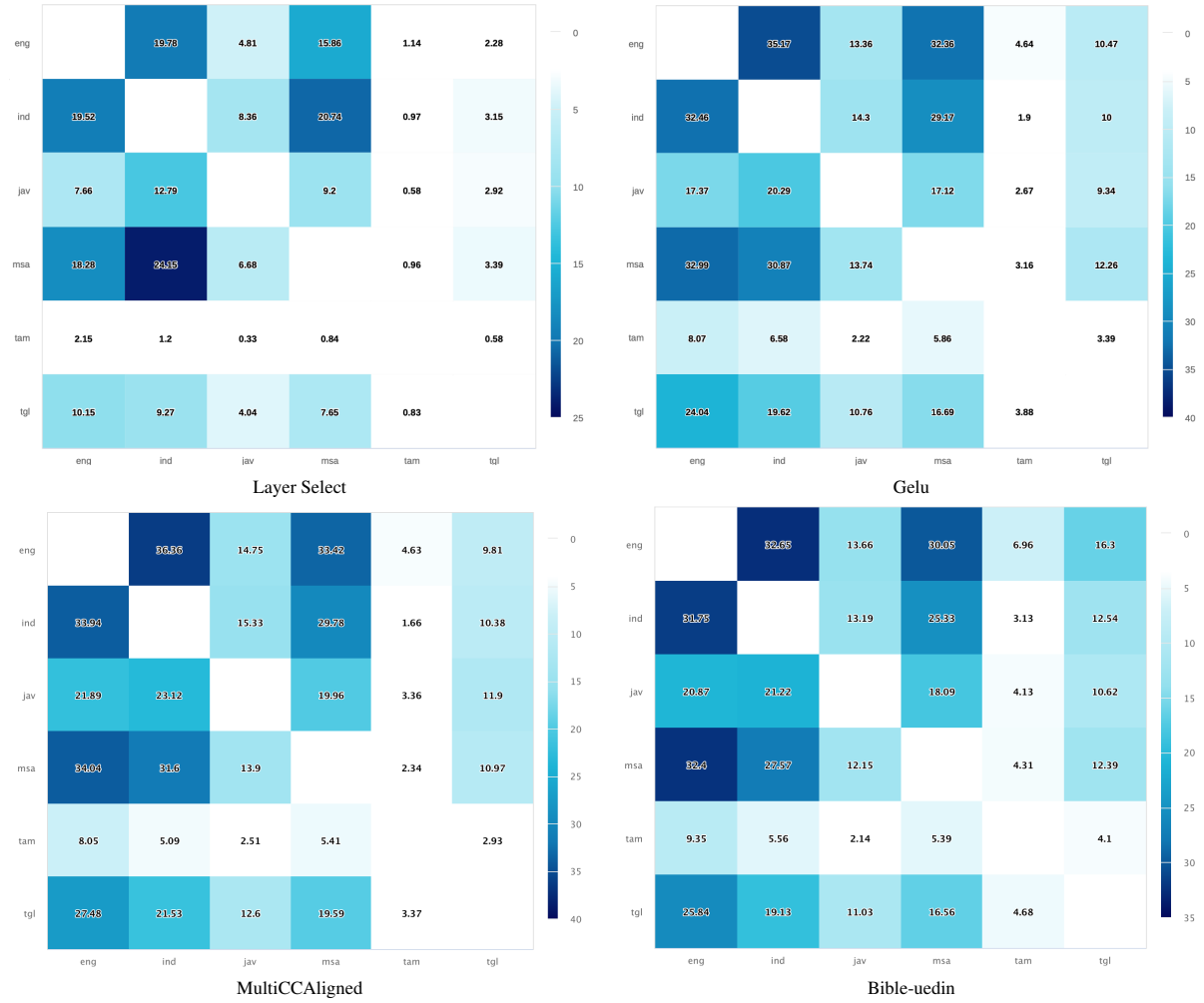


Figure 2: spBLEU score on the *test* setbreak down for each language pair

References

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: The bible in 100 languages. *Lang. Resour. Eval.*, 49(2):375–395.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint arXiv:2010.11125*.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. 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).

Minjia Zhang and Yuxiong He. 2020. Accelerating training of transformer-based language models with progressive layer dropping. *arXiv preprint arXiv:2010.13369*.

To optimize, or not to optimize, that is the question: TelU-KU models for WMT21 Large-Scale Multilingual Machine Translation

Sari Dewi Budiwati[1,2]*, **Tirana Noor Fatyanosa**[1]*, **Mahendra Data**[1,3], **Dedy Rahman Wijaya**[2], **Patrick Adolf Telnoni**[2], **Arie Ardiyanti Suryani**[4], **Agus Pratondo**[2], **Masayoshi Aritsugi**[5]

[1]Graduate School of Science and Technology, Kumamoto University, Japan
[2]School of Applied Science, Telkom University, Indonesia
[3]Faculty of Computer Science, Brawijaya University, Indonesia
[4]School of Informatics, Telkom University, Indonesia
[5]Faculty of Advanced Science and Technology, Kumamoto University, Japan
{saridewi, fatyanosa, mahendra.data}@dbms.cs.kumamoto-u.ac.jp
{saridewi, dedyrw, patrickadolf, ardiyanti, pratondo}@telkomuniversity.ac.id
mahendra.data@ub.ac.id, aritsugi@cs.kumamoto-u.ac.jp

Abstract

We describe TelU-KU models of large-scale multilingual machine translation for five Southeast Asian languages: Javanese, Indonesian, Malay, Tagalog, Tamil, and English. We explore a variation of hyperparameters of flores101_mm100_175M model using random search with 10% of datasets to improve BLEU scores of all thirty language pairs. We submitted two models, TelU-KU-175M and TelU-KU-175M_HPO, with average BLEU scores of 12.46 and 13.19, respectively. Our models show improvement in most language pairs after optimizing the hyperparameters. We also identified three language pairs that obtained a BLEU score of more than 15 while using less than 70 sentences of the training dataset: Indonesian-Tagalog, Tagalog-Indonesian, and Malay-Tagalog.

1 Introduction

This paper describes our participation in the WMT21 shared task of large-scale multilingual machine translation. Specifically, we chose small track #2, which involves thirty language pairs, and used the neural machine translation (NMT) method. We call our models TelU-KU (Telkom University - Kumamoto University) as we use our university name in our submissions.

*These authors contributed equally. Everyone contributed to writing this paper.

NMT has been widely used in machine translation research for many languages. Currently NMT has become the state of the art of machine translation with a large number of parallel corpus (Bojar et al., 2017; Nakazawa et al., 2017; Chu and Wang, 2018; Sutskever et al., 2014). Meanwhile, for low resources cases, the NMT tends to give poor translation results (Duh et al., 2013; Sennrich and Zhang, 2019; Zoph et al., 2016; Koehn and Knowles, 2017). In order to get better translation results of NMT for low resources languages, some approaches are applied, such as using a large number of monolingual corpora (Artetxe et al., 2018a,b; Lample et al., 2018b,a), applying transfer learning approach to share lexical and sentence level representation (Gu et al., 2018), using sub-word representation (Durrani et al., 2019), and hyperparameter optimization (HPO) (Sennrich and Zhang, 2019; Rubino et al., 2020).

HPO is an important part of building an NMT system in many real-world applications. In other words, selecting effective hyperparameters is critical to building a strong NMT system. However, in many cases, hyperparameters are often set manually based on intuition and heuristics mechanisms, tedious and error-prone processes that can lead to unreliable experimental results and poor performance of shared tasks or production systems.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 387–397
November 10–11, 2021.

This is because HPO requires rigorous testing and resources, which makes it a high-cost process. To deal with this problem, table-lookup has been proposed as a benchmark procedure (Zhang and Duh, 2020). Their study provides evaluation protocols and a benchmark dataset for comparing the HPO methods. Moreover, like other NMT models, transformers require setting various hyperparameters, but researchers often use default parameters, even when their data conditions differ substantially from the data conditions previously used to determine those default values.

In low-resource languages cases, the performance of the transformer is highly dependent on the hyperparameter settings (Araabi and Monz, 2020). The experimental results show that the best-suited combination of hyperparameters and regularization methods can produce substantial improvement for low-resource languages data. On the other side, grid search and manual search are the most frequently used strategies for HPO. However, according to the experiment, random search is actually better than grid search in several conditions (Bergstra and Bengio, 2012). Random searches are actually better suited to running on a cluster of computers than grid searches when a group of computers fails. Random search also allows the experimenter to change the "resolution" on the fly. In addition, they have advantages in high-dimensional searching spaces.

In this work, we experimented with two models, that is, TelU-KU-175M and TelU-KU-175M_HPO. Both models are based on a pre-trained model of flores101_mm100_175M. The TelU-KU-175M is our model that manually fine-tuning the hyperparameters, whereas the TelU-KU-175M_HPO is based on hyperparameter optimization. We used a random search method while using 10% of datasets to find the best hyperparameter optimization. In addition, we also included the M2M-100 175M model to compare with our results. This model uses the same pre-trained model as ours but without fine-tuning. Fine-tuning is a common practice in NLP to train a pre-trained model for several epochs on a downstream dataset and has proven to improve performance.

Our experimental results show improvement in most language pairs after optimizing the hyperparameters. The TelU-KU-175M is able to improves the average BLEU scores by 0.35-0.59 over M2M-100 175M. Meanwhile, the TelU-KU-175M_HPO improve the scores by 1.08-1.41 over the baseline. We also identified three language pairs that obtained a BLEU score of more than 15 while using less than 70 sentences of the training dataset: Id-Tl, Tl-Id, and Ms-Tl.

This paper is organized as follows. Section 2 explains the experiment. Section 3 shows the obtained results. Section 4 discusses the effect of HPO and the multilingual model. Section 5 provides the conclusion and future direction of this work.

2 Experiments

In this section, we first describe languages overview of the Southeast Asian language. Then, we discuss data and preprocessing. Finally, we discuss the model and architecture of our model submission.

2.1 Languages overview

We chose small track #2, which involves six languages from Southeast Asia, namely, Javanese (Jv), Indonesian (Id), Malay (Ms), Tagalog (Tl), Tamil (Ta), and English (En).

Indonesian and Malay are considered closely related languages due to being mutually intelligible in morphology, and both languages belong to the Malayo-Polynesian language family (Susanto et al., 2012). The base of formal Indonesian is from Malayo-Riau (Abas, 1987). The main difference is the influence of the vocabulary. Indonesian is largely influenced by Dutch, whereas English influences Malay. The Tagalog language has the same language family as Indonesian and Malay. However, it has different morphology characteristics, and the vocabulary is influenced by several countries, such as Spain, America, and Malay. Javanese is one of the Indonesian ethnic languages used by more than 42% of Indonesia's population, mostly from the central and eastern parts of Java (Novitasari et al., 2020). Javanese is also used in Suriname and New Caledonia. Currently, the Javanese is influenced by Indonesian. This is because Indonesian is used in

Language	Family	Alphabet
Indonesian	Malayo-Polynesian	Latin
Malay	Malayo-Polynesian	Latin
Tagalog	Malayo-Polynesian	Latin
Tamil	Dravidian	Tamil
English	Germanic	Latin
Javanese	Malayo-Polynesian	Latin

Table 1: Language family and writing system.

formal documents and as a daily conversation. Last, Tamil belongs to Dravidian, a unique family where the language is mostly spoken in a southern state (Tamil Nadu) of India (Kumar and Singh, 2019). Table 1 shows the general characteristic of the selected languages.

2.2 Data and preprocessing

We used a dataset provided by the WMT21 organizers. Thus, our system was considered a constrained system. We used three types of datasets, that is, training, evaluation, and hidden test datasets. The training dataset is a parallel corpus from Opus monolingual and Wikipedia, as shown in Table 2. The evaluation dataset is a parallel corpus from Flores101 (Goyal et al., 2021). The evaluation dataset consists of two evaluations, that is, *dev* and *devtest*, as much as, 997 and 1,012 sentences, respectively. Last, the hidden test dataset is an unknown parallel corpus provided by the organizer through the Dynabench leaderboard.[1]

We tokenized all the training and evaluation datasets by SentencePiece tokenizer (Sennrich et al., 2016). This tokenizer is an unsupervised text tokenizer and detokenizer, where the vocabulary size is predetermined prior to the neural model training. We preprocessed dataset according to the guideline,[2] that is, encode and binarize.

2.3 Models & Architectures

We use an NMT system with big Transformer architecture (Ng et al., 2019; Vaswani et al., 2017), i.e., *transformer_wmt_en_de_big*, as implemented in the Fairseq toolkit (Ott et al., 2019). We run experiments on Standard NC24 of Microsoft Azure virtual machine consisting of 4 NVidia Tesla K80 with 12 GB GPU memory.[3] We experimented with the following two models:

Language pairs	Sentences
En - Id	1,019,169
En - Jv	13,049
En - Ms	120,016
En - Ta	95,162
En - Tl	75,447
Id - Jv	42
Id - Ms	1,167
Id - Ta	24,648
Id - Tl	56
Jv - Ms	18
Jv - Ta	1,296
Jv - Tl	2,251
Ms - Ta	3,920
Ms - Tl	5
Ta - Tl	1,478

Table 2: Training datasets of each language pair.

- **TelU-KU-175M** is a pre-trained flores101_mm100_175M with fine-tuning. We manually tune the hyperparameters, as shown in Table 3, column 4.
- **TelU-KU-175M_HPO** is a pre-trained flores101_mm100_175M with HPO. The hyperparameters and their ranges are shown in Table 3. Some of these hyperparameters are based on (Ravikumar, 2020). Figure 1 shows the logical flow of our approach. We run 30 iterations of random searches for two epochs. Due to costly training, we only run the optimization using only 10% *training*, *dev*, and *devtest*. From those 30 models, we select the best model based on the results from the *devtest*. Then, we use the hyperparameter from the best models to fine-tune the flores101_mm100_175M model. The hyperparameter optimization results are shown in Table 3, column 5.

3 Results

We evaluate the generated texts of our models using the sentence-piece BLEU (spBLEU). The spBLEU uses a SentencePiece tokenizer with 256k tokens, and then the BLEU score is computed on the SentencePiece tokenized text. The results are shown in Table 4.

[1] https://www.dynabench.org/flores
[2] https://github.com/facebookresearch/flores
[3] The source code of our experiments is available at https://github.com/fatyanosa/WMT21

Hyper-parameter	Definition	Range	Hyperparameter value	
			TelU-KU-175M	TelU-KU-175M_HPO
BS	Batch size	Min: 8, Max: 128	128	15
LR	Learning rate	Min: 3e-05, Max: 3e-04	3e-05	0.000181463
BT1	Beta1	Min: 0.7, Max: 0.9999	0.9	0.745923
BT2	Beta2	Min: 0.7, Max: 0.9999	0.98	0.948909
EPS	Epsilon	Min: 9.98e-09, Max: 9.99e-06	1e-06	9.62e-06
WD	Weight Decay	Min: 0.0, Max: 0.018	0.0	0.00946414
AD	Attention Dropout	Min: 0.0, Max: 0.5	0.1	0.182843
DR	Dropout	Min: 0.0, Max: 0.5	0.3	0.0162452
SE	Seed	Min: 0, Max: 300	222	72

Table 3: Hyperparameter range for each hyperparameter and the values for each model.

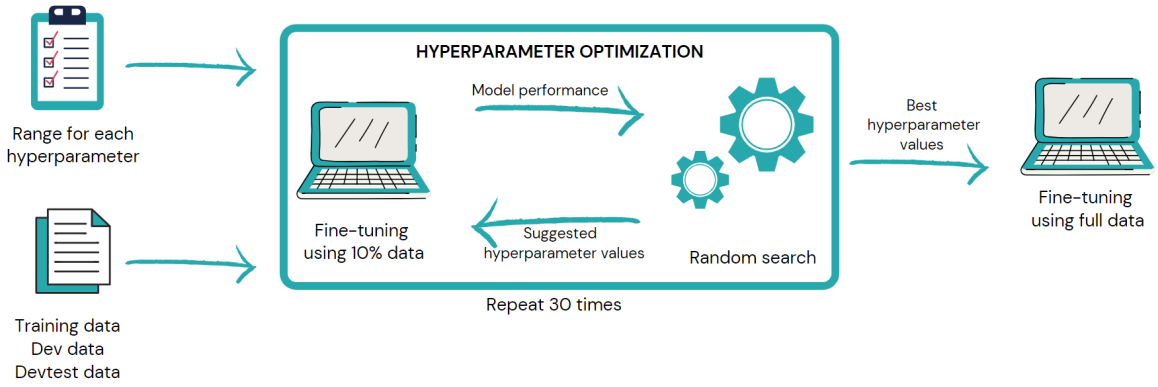


Figure 1: Logical flow of hyperparameter optimization.

We also compare our results with the pretrained model without fine-tuning (M2M-100 175M) as the baseline. Our models: the TelU-KU-175M and TelU-KU-175M_HPO, improved the BLEU scores on 16-17 language pairs over the M2M-100 175M model. In terms of the improvement in each language pair, TelU-KU-175M and TelU-KU-175M_HPO improved BLEU scores by 0.04-5.39 and 0.01-12.22, respectively. The BLEU scores also decreased on several language pairs between 0.1 to 3.25 and 0.02 to 6.5 for TelU-KU-175M and TelU-KU-175M_HPO, respectively.

4 Discussion

This section discusses the effect of HPO and NMT of the multilingual model against our models' evaluation results.

4.1 Effect of HPO

The main objective of HPO in this task is to explore a high-dimensional search space in NMT. As we mentioned in Section 2.3, we run HPO using random search for 30 iterations using 10% of the dataset. The best hyperparameter values were determined based on *devtest*. The best configurations based on the average BLEU scores for each iteration were saved for running the pretrained model, M2M-100 175M, using full datasets (TelU-KU-175M_HPO). The detail of the best configurations is shown in Table 3. The evaluation results were conducted by translating the *dev*, *devtest*, and *test* datasets. We show our detailed evaluation results in Table 4, while Table 5 is our average evaluation among other participants.

We found that fine-tuning of 10% dataset does not lead to the best results for all language pairs translation compared to the manual hyperparameter tuning (TelU-KU-175M) and fine-tuning using 100% dataset (TelU-KU-175M_HPO). The best HPO using 10% of datasets resulted in an average BLEU of 12.75 for dev, 12.33 for devtest, and 12.39 for test datasets. Nevertheless, these values are higher compared to the baseline (M2M-100 175M) in Table 5. This means that fine-tuning with only 10% dataset using a basic method, random search (Bergstra and Bengio, 2012), indeed increases the BLEU scores.

Lang pairs	Dev			Devtest			Test		
	M2M-100 175M	TelU-KU-175M	TelU-KU-175M_HPO	M2M-100 175M	TelU-KU-175M	TelU-KU-175M_HPO	M2M-100 175M	TelU-KU-175M	TelU-KU-175M_HPO
En-Id	28.13	33.36	39.58	28.25	33.34	40.47	28.82	32.97	40.33
Id-En	28.42	30.6	35.54	26.92	29.55	35.33	27.48	29.62	35.1
En-Jv	10.13	8.79	7.32	9.79	8.67	7.02	9.5	8.45	7.2
Jv-En	14.98	14.55	11.01	14.62	14.12	11.43	15.12	13.83	10.59
En-Ms	25.86	24.05	27.35	25.13	22.98	27.39	26.32	23.98	27.63
Ms-En	27.45	28.48	31.16	25.89	27.1	29.68	27.06	27.83	30.8
En-Ta	3.83	2.44	2.27	3.43	2.4	2.41	3.82	2.34	2.07
Ta-En	4.76	5.9	4.97	4.29	5.25	4.41	4.71	5.59	4.42
En-Tl	11.16	16.11	21.16	10.46	15.85	21.38	10.67	16.16	21.18
Tl-En	20.44	22.68	24.88	17.94	21.08	23.59	19.26	22.06	24.4
Id-Jv	12.42	10.72	7.24	12.24	10.92	7.79	11.65	10.1	6.7
Jv-Id	17.89	16.9	16.13	18.22	17	16.09	17.9	16.04	15.44
Id-Ms	26.33	23.53	25.21	25.85	22.92	22.32	26.61	23.36	24.08
Ms-Id	28.99	29.03	29.76	28.64	28.54	29.96	28.71	28.74	29.1
Id-Ta	1.02	2.23	2.35	0.89	2.19	2.23	1.02	2.01	2.31
Ta-Id	4.05	4.34	4.12	3.75	4.36	3.9	3.82	4.24	3.89
Id-Tl	8.11	12.98	18.43	7.41	12.4	17.95	7.75	12.78	17.94
Tl-Id	15.73	17.51	20.66	14.85	16.68	20.43	15.91	17.35	19.98
Jv-Ms	15.19	12.99	8.77	14.21	12.82	7.71	14.61	12.84	8.46
Ms-Jv	11.1	9.64	8.43	10.01	9.2	7.81	9.94	8.68	7.12
Jv-Ta	2.48	1.5	1.08	2.32	1.15	1.04	2.49	1.27	1.02
Ta-Jv	0.88	1.39	0.89	0.7	1.31	1.02	0.76	1.23	0.81
Jv-Tl	8.37	8.37	9.1	7.78	8.24	8.81	7.86	8.41	8.65
Tl-Jv	8.12	7.56	5.59	7.58	7.3	4.82	7.91	7.15	5.25
Ms-Ta	2.71	2.81	3.77	2.29	2.53	3.86	2.64	2.53	3.64
Ta-Ms	3.78	3.9	2.1	3.46	3.71	2.29	3.64	3.9	2.18
Ms-Tl	9.57	12.48	16.42	8.88	11.9	16.3	8.91	12.03	16.09
Tl-Ms	14.59	14.24	14.57	12.53	12.24	11.64	13.5	13.12	12.92
Ta-Tl	2.63	3.03	4.62	2.67	3.31	4.35	2.62	3.04	4.17
Tl-Ta	2.64	2.27	2.57	2.4	2.12	2.24	2.42	2.12	2.31

Table 4: Summary of results for all language pairs based on BLEU scores. Blue font means that there is an improvement over the M2M-100 175M model, while red font means a decline over the M2M-100 175M model.

The BLEU scores improved even more after using the full dataset with the same hyperparameter values (TelU-KU-175M_HPO). This means that the number of datasets influences the performance. We left for future work discussing the effect of the number of datasets in HPO for NMT.

We also study the hyperparameter importance of all optimized hyperparameters using Hyanova[4], a python implementation of a functional analysis of variance (fANOVA) algorithm (Hutter et al., 2014). The algorithm partitions the observed variation of a response value into components against its inputs (Klein and Hutter, 2019). In this study, the response value is the BLEU score, while hyperparameters are the inputs. The higher the fANOVA values, the more important the hyperparameter. Table 6 shows that LR is the most important hyperparameter, while BS is the least important. This means that the LR influence the achieved BLEU scores. From our observation, within our selected range in Table 3, the higher the learning rate, the higher the average BLEU score. Therefore, it is important to tune the LR within a higher range.

Furthermore, we investigate the statistical significance across language pairs from Table 4 using Wilcoxon signed-rank test with $\alpha = 0.05$. We show the p-values of all models in Table 7. Unfortunately, all the results demonstrate statistically non-significant as all of the p-values were more than 0.05. Although the average BLEU can be increased by optimizing the hyperparameter values, this finding shows that HPO might not contribute much to the performance.

One of the possible causes is the utilization of random search, which is categorized as an uninformed search. This category does not learn from previous results, and therefore, each solution is independent of the other. Moreover, uninformed search is proven to be inferior to the informed search, i.e.,

[4] https://pypi.org/project/hyanova/

	Model	Average Bleu Score		
		Dev	Devtest	Test
Other participants	DeltaLM+Zcode (Microsoft-Small)	34.09	33.94	33.89
	615m (Baohao Liao)	33.74	33.51	33.34
	TenTrans (Wanying Xie)	29.25	28.94	28.89
	adaavg (Danni Liu)	28.70	29.09	28.64
	huawei-tsc1 (huaweitsc)	28.64	28.34	28.40
	srph-large (jcblaisecruz02)	22.92	23.14	22.97
	finetune-saptarashmi (saptab)	16.05	15.45	15.72
	615m-new (zizhenlian)	15.50	14.89	15.10
Ours	TelU-KU-175M_HPO	13.57	13.19	13.19
	TelU-KU-175M	12.81	12.37	12.46
Baseline	M2M-100 175M	12.39	11.78	12.11

Table 5: Average BLEU scores from all submitted systems.

Hyper-parameter	Importance value
BS	0.94
LR	1.02
BT1	1.00
BT2	1.00
EPS	1.00
WD	1.00
AD	1.00
DR	1.00
SE	0.99

Table 6: Hyperparameter importance.

bayesian optimization or evolutionary algorithm (Fatyanosa and Aritsugi, 2020, 2021).

This work only calculates the statistical significance across language pairs and leaves the calculation per language pair for future studies.

4.2 Effect of multilingual model

The TelU-KU-175M and TelU-KU-175M_HPO models produced 16-17 language pairs that have higher BLEU scores compared to M2M-100 175M, as shown in Table 4 with blue colors. Among them, we identified seven language pairs that obtained a BLEU score below 10: Id-Ta, Ta-Id, Ta-Jv, Ms-Ta, Ta-Ms, Ta-Tl, and Jv-Tl. Most of these language pairs were related to the Tamil language. We found that most of the Tamil translation results had an English sentence as unknown words (see Tables 8 and 9 in appendix). The translation results leading to not the same as a reference file. As a result, these language pairs had a lower BLEU score.

Surprisingly, we identified three language pairs that obtained a BLEU score of more than 15: Id-Tl, Tl-Id, and Ms-Tl, while using less than 70 sentences of the training dataset. This could be because of the attention mechanism in NMT of multilingual models. The attention mechanism, which was initially called a soft-alignment model in (Bahdanau et al., 2015), aligns a source phrase to a target word. Training this attention-based model is done by maximizing the conditional log-likelihood. After training, the model can do translation from any of the source languages to any of the target languages included in the parallel training corpora (Firat et al., 2016).

The Id-Tl, for example, obtained a BLEU score of 17.94 using only 56 sentences of training datasets. The NMT system that trained with fewer training datasets, e.g., below 1M, usually obtained lower BLEU scores. However, the Id-Tl results indicated that this language pair obtained an advantage from the attention mechanism of the multilingual model using 30 language pairs. In this study, these 30 language pairs are considered as low-resource languages and mostly have the same language family. Table 2 shows that our models used a small number of training datasets, e.g., below 1M, in all language pairs. Whereas, Table 1 shows that most languages have the same language family: Malayo-Polynesian. Therefore, we argue that low-resource language with the same language family should be considered in the NMT of the multilingual model. For example, if we want to improve the Tamil language performance, we should considered to add other languages with the same (or closely) language family as Tamil, e.g., Kannada, Bengali, Hindi.

Model	Dev			Devtest			Test		
	M2M-100 175M	TelU-KU-175M	TelU-KU-175M_HPO	M2M-100 175M	TelU-KU-175M	TelU-KU-175M_HPO	M2M-100 175M	TelU-KU-175M	TelU-KU-175M_HPO
M2M-100 175M	x	0.567	0.399	x	0.360	0.289	x	0.734	0.572
TelU-KU-175M		x	0.229		x	0.329		x	0.360
TelU-KU-175M_HPO			x			x			x

Table 7: Results of Wilcoxon signed-rank test.

5 Conclusion

We described our team submission for WMT21. Our results show improvement in most language pairs after optimizing the hyperparameters.

Furthermore, we also found three language pairs that obtained a BLEU score of more than 15 while using less than 70 sentences of the training dataset. In this study, we used 30 language pairs that are considered as low-resource language and mostly have the same language family. This result indicated that low-resource language with the same language family should be considered in the NMT of the multilingual model.

As future work, we plan to use a more sophisticated optimization algorithm, specifically informed searches such as bayesian optimization or evolutionary algorithm. Additionally, we want to try other percentages of the optimized dataset to see the effect of the number of training data on the performance. We also plan to use a specific tokenizer for Tamil, e.g., Indic NLP library (Kunchukuttan, 2020), iNLTK (Arora, 2020). The Tamil language needs a particular pre-processing due to its writing system that differs from other languages. Last, we plan to clean the dataset in pre-processing steps, considering that the dataset used in this work is noisy. We expect this will maximize the attention mechanism in the NMT of a multilingual model. Therefore, our model could produce better translation results.

Acknowledgements

This project was funded by Compute Grants: Large-Scale Multilingual Machine Translation of Conference on Machine Translation (WMT) and Microsoft Azure. This project was also funded by the PDT research scheme, Telkom University.

References

Husen Abas. 1987. Indonesian as a unifying language of wider communication : a historical and sociolinguistic perspective.

Ali Araabi and Christof Monz. 2020. Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3429–3435, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gaurav Arora. 2020. iNLTK: Natural language toolkit for indic languages. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 66–71, Online. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. 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 *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13.

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.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. 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, Sofia, Bulgaria. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Tirana Noor Fatyanosa and Masayoshi Aritsugi. 2020. Effects of the Number of Hyperparameters on the Performance of GA-CNN. In *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pages 144–153. IEEE.

Tirana Noor Fatyanosa and Masayoshi Aritsugi. 2021. An Automatic Convolutional Neural Network Optimization Using a Diversity-Guided Genetic Algorithm. *IEEE Access*, 9:91410 – 91426.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 866–875. The Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. 2014. An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page I–754–I–762. JMLR.org.

Aaron Klein and Frank Hutter. 2019. Tabular benchmarks for joint architecture and hyperparameter optimization.

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.

Amit Kumar and Anil Kumar Singh. 2019. NLPRL at WAT2019: transformer-based tamil - english indic task neural machine translation system. In *Proceedings of the 6th Workshop on Asian Translation, WAT@EMNLP-IJCNLP 2019, Hong Kong, China, November 4, 2019*, pages 171–174. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Hideya Mino, Isao Goto, Hideto Kazawa, Yusuke Oda, Graham Neubig, and Sadao Kurohashi. 2017. Overview of the 4th workshop on Asian translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 1–54, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319. Association for Computational Linguistics.

Sashi Novitasari, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2020. Cross-lingual machine speech chain for javanese, sundanese, balinese, and bataks speech recognition and synthesis. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages and Collaboration and Computing for Under-Resourced Languages, SLTU/CCURL@LREC 2020, Marseille, France, May 2020*, pages 131–138. European Language Resources association.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Meghana Ravikumar. 2020. Efficient BERT: Finding Your Optimal Model with Multimetric Bayesian Optimization.

Raphael Rubino, Benjamin Marie, Raj Dabre, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2020. Extremely low-resource neural machine translation for asian languages. *Mach. Transl.*, 34(4):347–382.

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, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Raymond Hendy Susanto, Septina Dian Larasati, and Francis M. Tyers. 2012. Rule-based Machine Translation between Indonesian and Malaysian. In *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing, WSSANLP@COLING 2012, Mumbai, India, December 8, 2012*, pages 191–200.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Xuan Zhang and Kevin Duh. 2020. Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems. *Transactions of the Association for Computational Linguistics*, 8:393–408.

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.

A Appendix

Lang pairs	Model	Text
English	Source	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
En-Id	Reference	“Saat ini ada mencit umur 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
	M2M-100_175M	Kami sekarang memiliki kuku 4 bulan yang tidak diabetik yang digunakan untuk menjadi diabetik, katanya.
	TelU-KU-175M	Kami sekarang memiliki mice berusia 4 bulan yang bukan diabetik yang digunakan untuk menjadi diabetes, ia menambahkan.
	TelU-KU-175M_HPO	Kami sekarang memiliki tikus berusia 4 bulan yang non-diabetik yang digunakan untuk diabetik, dia menambahkan.
En-Jv	Reference	Saiki kita nduweni tikus umur-4-sasi sing ora-nduweni-diabetes sing sadurunge nduweni diabetes, ujare.
	M2M-100_175M	Kita ora ana 4 bulan madu yang non-diabetik sing digunakake kanggo diabetes, katanya.
	TelU-KU-175M	Dhèwèké ya iku mice 4 bulan kang non-diabetik kang digunakaké diabetes,” dhèwèké tambah.
	TelU-KU-175M_HPO	Nanging, dhèwèké bisa nganti 4 bulan kang ora diwènèhi yèn diwènèhi diapit.
En-Ms	Reference	“Kami kini mempunyai seekor anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambah.
	M2M-100_175M	Kami kini mempunyai kuku 4 bulan yang tidak diabetik yang digunakan untuk menjadi diabetik, katanya.
	TelU-KU-175M	Kami sekarang mempunyai mice berusia 4 bulan yang bukan-diabetik yang digunakan untuk menjadi diabetes, ia tambah.
	TelU-KU-175M_HPO	Kami sekarang mempunyai tikus berusia 4 bulan yang bukan-diabetik yang digunakan sebagai diabetik, ia menambah.
En-Ta	Reference	"எங்களிடம் இப்போது 4-மாத-வயதுடைய எலி ஒன்று உள்ளது, முன்னர் அதற்கு நீரிழிவு இருந்தது தற்போது இல்லை" என்று அவர் மேலும் கூறினார்.
	M2M-100_175M	நாங்கள் இப்போது 4 மாதங்களுக்குள் சுழுதைகள் உள்ளன, இது இரகசியமாக பயன்படுத்தப்படாத இரகசியமாக உள்ளது, அவர் கூறினார்.
	TelU-KU-175M	We now have 4-month-old mice that are non-diabetic that are used to be diabetes, he added.
	TelU-KU-175M_HPO	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
En-Tl	Reference	Mayroon na tayong 4 na buwang gulang na daga na hindi diabetic na dating diabetic, dagdag niya.
	M2M-100_175M	Kapag medyo nakarating ka na sa age na alam mo na ang dami mo nangpinagdaanan... a good soldier must knowwhen to surrender.
	TelU-KU-175M	Tapos ay may 4-month-old mice na hindi-diabetic na ginagamit upang maging diabetes,” siya ay hinaharap.
	TelU-KU-175M_HPO	We now have four-month-old mice that are non-diabetic that used to be diabetic, he added.
Indonesia	Source	“Saat ini ada mencit umur 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
Id-En	Reference	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
	M2M-100_175M	“Now there is a age of 4 months of non-diabetes that has long been diabetes,” she said.
	TelU-KU-175M	This day there is a 4 months old nondiabetes who had previously been diabetes, his debut.
	TelU-KU-175M_HPO	In this period there is a four-month-old nondiabetes that was formerly diabetes, his tambahnya.
Id-Jv	Reference	Saiki kita nduweni tikus umur-4-sasi sing ora-nduweni-diabetes sing sadurunge nduweni diabetes, ujare.
	M2M-100_175M	“Itu ora tumitake umur 4 bulan nondiabetes sing iku diabetes,” tambahnya.
	TelU-KU-175M	]]]]]]]]]]]]]]]]
	TelU-KU-175M_HPO	“Selang iki duwé umur 4 bulan nondiabetes kang formeré diabetes,” tambahnya.
Id-Ms	Reference	“Kami kini mempunyai seekor anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambah.
	M2M-100_175M	“Pada masa ini, ia mempunyai umur 4 bulan nondiabetes yang sejak itu diabetes,” tambahnya.
	TelU-KU-175M	Saat ini sudah berusia 4 bulan nondiabetes yang dulunya diabetes, tambahnya.
	TelU-KU-175M_HPO	“Saat ini terdapat anak berusia 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
Id-Ta	Reference	"எங்களிடம் இப்போது 4-மாத-வயதுடைய எலி ஒன்று உள்ளது, முன்னர் அதற்கு நீரிழிவு இருந்தது தற்போது இல்லை" என்று அவர் மேலும் கூறினார்.
	M2M-100_175M	இந்த நிலையில், 4 மாதங்களுக்கு முன்னர் சிகிச்சை அளிக்கப்பட்டு சிகிச்சை அளிக்கப்பட்டு சிகிச்சை பெற்றுள்ளது.
	TelU-KU-175M	This time there is a four-month-old nondiabetes that had previously been diabetes, his debut.
	TelU-KU-175M_HPO	This time there is a four-month-old nondiabetes that was formerly diabetes, his tambahnya.
Id-Tl	Reference	Mayroon na tayong 4 na buwang gulang na daga na hindi diabetic na dating diabetic, dagdag niya.
	M2M-100_175M	Kapag medyo nakarating ka na sa age na alam mo na ang dami mo nangpinagdaanan... a good soldier must knowwhen to surrender.
	TelU-KU-175M	Ang ito ay may edad na 4 buwan ng nondiabetes na dating diabetes,” tumutukoy niya.
	TelU-KU-175M_HPO	Sa panahong ito ay mayroong apat na buwan ng nondiabetes na dating diabetes, kanyang tambahnya.
Javanese	Source	Saiki kita nduweni tikus umur-4-sasi sing ora-nduweni-diabetes sing sadurunge nduweni diabetes, ujare.
Jv-En	Reference	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
	M2M-100_175M	Now we have four-year-old age, which are not diagnosed with diabetes, diabetes.
	TelU-KU-175M	We have four-year-old children who have diabetes, ujare.
	TelU-KU-175M_HPO	Today we have a four-year-old mice that is not-to-be-diabetes who now has diabetes, then.
Jv-Id	Reference	“Saat ini ada mencit umur 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
	M2M-100_175M	Tetapi kita memiliki usia 4 tahun yang tidak memiliki diabetes yang sering memiliki diabetes, seperti itu.
	TelU-KU-175M	Saiki kita memiliki tikus berusia 4 tahun yang sekarang-nduweni-diabetes yang memiliki diabetes, ujare.
	TelU-KU-175M_HPO	Tetapi kita memiliki tikus berusia 4 tahun yang tidak-wangian-diabetes yang sekarang memiliki diabetes, ujare.
Jv-Ms	Reference	“Kami kini mempunyai seekor anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambah.
	M2M-100_175M	Sekarang kita mempunyai usia 4 tahun yang tidak mempunyai diabetes yang mempunyai diabetes, boleh.
	TelU-KU-175M	Saiki kita mempunyai tikus berusia 4-sasi yang sekarang-nduweni-diabetes yang mempunyai diabetes, ujare.
	TelU-KU-175M_HPO	Terdapat kita mempunyai tikus berusia-4-sasi yang kini mempunyai diabetes, ujare. ”
Jv-Ta	Reference	"எங்களிடம் இப்போது 4-மாத-வயதுடைய எலி ஒன்று உள்ளது, முன்னர் அதற்கு நீரிழிவு இருந்தது தற்போது இல்லை" என்று அவர் மேலும் கூறினார்.
	M2M-100_175M	நாங்கள் இப்போது 4 வயதில் 4 வயதைக் கொண்டே இருக்கிறோம், இது நோயாளிகளின் நோயாளிகள், நோயாளிகள்.
	TelU-KU-175M	We have four-year-old tikus who are now-anduweni-diabetes with diabetes, ujare.
	TelU-KU-175M_HPO	நாங்கள் 4 வயதிற்குட்பட்ட குழந்தைகளைக் கொண்டிருக்கின்றோம்" (ஆங்கிலம்: "ஆங்கிலம்: <unk><unk><unk><unk><unk><unk><unk><unk><unk><unk><unk><unk>)".
Jv-Tl	Reference	Mayroon na tayong 4 na buwang gulang na daga na hindi diabetic na dating diabetic, dagdag niya.
	M2M-100_175M	Sa ngayon, kami ay nagkaroon ng isang edad ng 4 na edad na hindi nag-aalok sa diabetes na may diabetes, na nag-aalok.
	TelU-KU-175M	Sa ngayon ay may 4-sasyong babae na hindi-duweni-diabetes na nagkaroon ng diabetes, ujare.
	TelU-KU-175M_HPO	Sa ngayon ay mayroong 4 taong gulang na mice na hindi-milyang-diabetes na nakarating na may diabetes, pangalang.”

Table 8: Translation results.

Lang pairs	Model	Text
Malay	Source	“Kami kini mempunyai seekor anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambah.
Ms-En	Reference	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
	M2M-100_175M	“We now have a four-month old baby that has no diabetes,” he added.
	TelU-KU-175M	We now have a four-month boy that has no diabetes, he added.
	TelU-KU-175M_HPO	We now have a four-month-old mice child who has no diabetes, he added.
Ms-Id	Reference	“Saat ini ada mencit umur 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
	M2M-100_175M	“Kami sekarang memiliki seekor tikus yang berusia 4 bulan yang belum menderita diabetes,” katanya.
	TelU-KU-175M	Kami sekarang memiliki seekor tikus berusia 4 bulan yang sudah tidak menghidap diabetes, dia menambahkan.
	TelU-KU-175M_HPO	“Kami kini memiliki seorang anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambahkan.
Ms-Jv	Reference	Saiki kita nduweni tikus umur-4-sasi sing ora-nduweni-diabetes sing sadurunge nduweni diabetes, ujare.
	M2M-100_175M	“Kita ora nduweni anak tikus 4 bulan lan ora ana diabetes,” tambah.
	TelU-KU-175M	We now have a four-month boy tikus who has no menghidap diabetes, she added.
	TelU-KU-175M_HPO	“Kami kini mempunyai seekor anak tikus yang umur 4 bulan yang sudah tidak menghidap diabetes,” beliau menambahkan.
Ms-Ta	Reference	"எங்களிடம் இப்போது 4-மாத-வயதுடைய எலி ஒன்று உள்ளது, முன்னர் அதற்கு நீரிழிவு இருந்தது தற்போது இல்லை" என்று அவர் மேலும் கூறினார்.
	M2M-100_175M	“ஒரு வயதிலிருந்து 4 மாதங்களுக்கு முன்னர் ஒரு குழந்தையைப் பெற்றோம்” என்று அவர் கூறியுள்ளார்.
	TelU-KU-175M	We now have a four-month-old girl who has no diabetes, she added.
	TelU-KU-175M_HPO	We now have a four-month-old tikus child who already does not live on diabetes, he added.
Ms-Tl	Reference	Mayroon na tayong 4 na buwang gulang na daga na hindi diabetic na dating diabetic, dagdag niya.
	M2M-100_175M	Kapag medyo nakarating ka na sa age na alam mo na ang dami mo nangpinagdaanan... a good soldier must know when to surrender.
	TelU-KU-175M	Tapos ay may isang 4 buwan na anak na hindi nakapagpatuloy ng diabetes,” siya ay nagsasabi.
	TelU-KU-175M_HPO	“Kami ngayon ay may isang anak na tikus na 4 na buwan na hindi namuhay ng diabetes,” he added.
Tamil	Source	"எங்களிடம் இப்போது 4-மாத-வயதுடைய எலி ஒன்று உள்ளது, முன்னர் அதற்கு நீரிழிவு இருந்தது தற்போது இல்லை" என்று அவர் மேலும் கூறினார்.
Ta-En	Reference	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
	M2M-100_175M	He said, We have now one of the four-year-old rolls, and it was not until it was long,” he further said.
	TelU-KU-175M	He said to me, I have a four-thirds light now, but it was not until it was lost.”
	TelU-KU-175M_HPO	He said, ”I am now one of the four-member elephants, and that it was not until it had been passed.”
Ta-Id	Reference	“Saat ini ada mencit umur 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
	M2M-100_175M	“Kami sudah memiliki satu dari empat orang, dan tidak ada lagi yang terjadi sebelumnya,” katanya.
	TelU-KU-175M	Ia berkata kepada kami sekarang, ”Saya sudah ada empat jam, dan sebelumnya tidak ada waktu yang lama.”
	TelU-KU-175M_HPO	Ia berkata, ”Tentara ini sekarang memiliki empat kursi, dan sebelumnya sudah tidak ada lagi,” katanya.
Ta-Jv	Reference	Saiki kita nduweni tikus umur-4-sasi sing ora-nduweni-diabetes sing sadurunge nduweni diabetes, ujare.
	M2M-100_175M	Itu uga ana ing kita minangka salah siji saka 4 taun, nanging ora ana ing nganti-nganti iku ing nganti, katanya.
	TelU-KU-175M	Dhèwèké diprodhuksi déning dhèwèké nganti 4 kaliyan, lan ora diprodhuksi”.
	TelU-KU-175M_HPO	Piyambakipun nggantosaken ”Sang” ingkang dipunsebat ”Sang” inggih punika ”Sang” ingkang dipunsebat ”Sang” lan ”Sang” punika ”Sang ”Sang” ingkang dipunsebat ”Sang”.
Ta-Ms	Reference	“Kami kini mempunyai seekor anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambah.
	M2M-100_175M	Beliau berkata, “Kami mempunyai satu daripada empat orang, dan ia tidak pernah berlaku sebelum ini,” katanya.
	TelU-KU-175M	Saya sekarang mempunyai satu lilin empat, ia juga berkata kepada saya tidak akan lama lagi, katanya.
	TelU-KU-175M_HPO	Lagi ini, ia berkata Terdapat unsur api empat, tetapi ia tidak lagi menyahpepijat.”
Ta-Tl	Reference	Mayroon na tayong 4 na buwang gulang na daga na hindi diabetic na dating diabetic, dagdag niya.
	M2M-100_175M	Kapag medyo nakarating ka na sa age na alam mo na ang dami mo nangpinagdaanan... a good soldier must know when to surrender.
	TelU-KU-175M	Siya ay isang seryeng tao, at hindi siya nag-iisip sa kanya.
	TelU-KU-175M_HPO	Iminungkahi na ”ang isang four-mga relyo sa kami ngayon ay isang relasyon, na hindi ito pinapatakbo nang hindi ito naganap.”
Tagalog	Source	Mayroon na tayong 4 na buwang gulang na daga na hindi diabetic na dating diabetic, dagdag niya.
Tl-En	Reference	We now have 4-month-old mice that are non-diabetic that used to be diabetic, he added.
	M2M-100_175M	We have 4 small areas that do not have diabetic dating diabetic, he said.
	TelU-KU-175M	May we have 4 new gullies that are not diabetic dating diabetesic, he says.
	TelU-KU-175M_HPO	There are four-year-old, non- diabetic, former diabetic, he added.
Tl-Id	Reference	“Saat ini ada mencit umur 4 bulan nondiabetes yang dulunya diabetes,” tambahnya.
	M2M-100_175M	Kami memiliki 4 negara yang tidak diabetik dan tidak diabetik, kata dia.
	TelU-KU-175M	Kami memiliki 4 warna putih yang tidak diabetes yang dating diabetes, dia tahu.
	TelU-KU-175M_HPO	Tetapi ada empat orang tua tua yang tidak diabetik yang sebelumnya diabetic, katanya.
Tl-Jv	Reference	Saiki kita nduweni tikus umur-4-sasi sing ora-nduweni-diabetes sing sadurunge nduweni diabetes, ujare.
	M2M-100_175M	Kita ana 4 negara sing padha padha padha padha diabetes sing ora diabetik, ujar dia.
	TelU-KU-175M	Dhèwèké dhèwèké dhèwèké nggèké 4 gulang kang tidak diabetes kang dating diabetes,” dhèwèké.
	TelU-KU-175M_HPO	Tetapi ana 4 gulungan kang umuré lan ora diabetik kang former diabetic, dhèwèké maragani.
Tl-Ms	Reference	“Kami kini mempunyai seekor anak tikus yang berusia 4 bulan yang sudah tidak menghidap diabetes,” beliau menambah.
	M2M-100_175M	Kami mempunyai 4 buah negara yang tidak diabetik dan tidak diabetik, katanya.
	TelU-KU-175M	Kami mempunyai 4 warna putih yang tidak diabetes yang dating diabetes, diadagangkan.
	TelU-KU-175M_HPO	Terdapat ana 4 orang tua yang gaya yang bukan diabetik yang bekas diabetic, kata-kata dia.
Tl-Ta	Reference	"எங்களிடம் இப்போது 4-மாத-வயதுடைய எலி ஒன்று உள்ளது, முன்னர் அதற்கு நீரிழிவு இருந்தது தற்போது இல்லை" என்று அவர் மேலும் கூறினார்.
	M2M-100_175M	நாங்கள் 4 மாதங்கள் கழித்து நோய் நோய் நோய் நோயாளிகள் இல்லை என்று அவர் கூறியுள்ளார்.
	TelU-KU-175M	We have 4 new colored days that are not diabetesic dating diabetesic, hedaged.
	TelU-KU-175M_HPO	நாம் 4வது வயதிலேயே, முன்னேற்றமான, முன்னேற்றமான, முன்னேற்றமான, முன்னேற்றத்தின் முன்னேற்றத்தில்" என்று அவர் கூறுகிறார்.

Table 9: Translation results.

MMTAfrica: Multilingual Machine Translation for African Languages

Chris C. Emezue[*]
Technical University of Munich
Mila Quebec AI Institute[†]
Masakhane

Bonaventure F. P. Dossou[*]
Jacobs University Bremen
Mila Quebec AI Institute[†]
Masakhane

Abstract

In this paper, we focus on the task of multilingual machine translation for African languages and describe our contribution in the 2021 WMT Shared Task: Large-Scale Multilingual Machine Translation. We introduce MMTAfrica, the first many-to-many multilingual translation system for six African languages: Fon (fon), Igbo (ibo), Kinyarwanda (kin), Swahili/Kiswahili (swa), Xhosa (xho), and Yoruba (yor) and two non-African languages: English (eng) and French (fra). For multilingual translation concerning African languages, we introduce a novel backtranslation and reconstruction objective, BT&REC, inspired by the random online back translation and T5 modelling framework respectively, to effectively leverage monolingual data. Additionally, we report improvements from MMTAfrica over the FLORES 101 benchmarks (spBLEU gains ranging from $+0.58$ in Swahili to French to $+19.46$ in French to Xhosa).

In this paper, we make use of the following notations:

- $\circledast$ refers to any language in the set $\{eng, fra, ibo, fon, swa, kin, xho, yor\}$.
- $\diamond$ refers to any language in the set $\{eng, fra, ibo, fon\}$.
- AL(s) refers to African language(s).
- X$\longrightarrow$ Y refers to neural machine translation from language X to language Y.

1 Introduction

Despite the progress of multilingual machine translation (MMT) and the many efforts towards improving its performance for low-resource languages, African languages suffer from under-representation. For example, of the 2000 known African languages (Eberhard et al., 2020) only 17 of them are available in the FLORES 101 Large-scale Multilingual Translation Task as at the time of this research. Furthermore, most research that look into transfer learning of multlilingual models from high-resource to low-resource languages rarely work with ALs in the low-resource scenario. While the consensus is that the outcome of the research made using the low-resource non-African languages should be scalable to African languages, this cross-lingual generalization is not guaranteed(Orife et al., 2020) and the extent to which it actually works remains largely understudied. Transfer learning from African languages to African languages sharing the same language sub-class has been shown to give better translation quality than from high-resource Anglo-centric languages (Nyoni and Bassett, 2021) calling for the need to investigate AL$\longleftrightarrow$AL multilingual translation.

This low representation of African languages goes beyond machine translation (Martinus and Abbott, 2019; Joshi et al., 2020; $\forall$ et al., 2020). The analysis conducted by $\forall$ et al. (2020) revealed that low-resourcedness of African languages can be traced to the poor incorporation of African languages in the NLP research community (Joshi et al., 2020). All these call for the inclusion of more African languages in multilingual NLP research and experiments.

With the high linguistic diversity in Africa, multilingual machine translation systems are very important for inter-cultural communication, which is in turn necessary for peace and progress. For example, one widely growing initiative to curb the large gap in scientific research in Africa is to translate educational content and scientific papers to various African languages in order to reach far more African native speakers (Abbott and Mart-

[*]Authors contributed equally to this work. Correspondence to *chris.emezue@gmail.com* or *femipancrace.dossou@gmail.com*

[†]Independent Research done while interning at Mila Quebec AI Institute

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 398–411
November 10–11, 2021.

inus, 2018; Nordling, 2018; Wild, 2021).

We take a step towards addressing the under-representation of African languages in MMT and improving experiments by participating in the 2021 WMT Shared Task: Large-Scale Multilingual Machine Translation with a major target of ALs⟷ALs. In this paper, we focused on 6 African languages and 2 non-African languages (English and French). Table 1 gives an overview of our focus African languages in terms of their language family, number of speakers and the regions in Africa where they are spoken (Adelani et al., 2021b). We chose these languages in an effort to create some language diversity: the 6 African languages span the most widely and least spoken languages in Africa. Additionally, they have some similar, as well as contrasting, characteristics which offer interesting insights for future work in ALs:

- Igbo, Yorùbá and Fon use diacritics in their language structure while Kinyarwanda, Swahili and Xhosa do not. Various forms of code-mixing are prevalent in Igbo (Dossou and Emezue, 2021b).
- Fon was particularly chosen because there is only a minuscule amount of online (parallel or monolingual) corpora compared to the other 5 languages. We wanted to investigate and provide valuable insights on improving translation quality of very low-resourced African languages.
- Kinyarwanda and Fon are the only African languages in our work not covered in the FLORES Large-Scale Multilingual Machine Translation Task and also not included in the pretraining of the original model framework used for MMTAfrica. Based on this, we were able to understand the performance of multilingual translation finetuning involving languages not used in the original pretraining objective. We also offered a method to improve the translation quality of such languages.

Our main contributions are summarized below:

1. MMTAfrica – a many-to-many AL⟷AL multilingual model for 6 African languages.
2. Our novel reconstruction objective (described in section 4.2) and the `BT&REC` finetuning setting, together with our proposals in section 5.1 offer a comprehensive strategy for effectively exploiting monolingual data of African languages in AL⟷AL multilingual machine translation,
3. Evaluation of MMTAfrica on the `FLORES Test Set` reports significant gains in spBLEU over the M2M MMT (Fan et al., 2020) benchmark model provided by Goyal et al. (2021),
4. We further created a unique highly representative test set – `MMTAfrica Test Set` – and reported benchmark results and insights using MMTAfrica.

Language	Lang ID (ISO 639-3)	Family	Speakers	Region
Igbo	ibo	Niger-Congo-Volta-Niger	27M	West
Fon (Fongbe)	fon	Niger-Congo-Volta-Congo-Gbe	1.7M	West
Kinyarwanda	kin	Niger-Congo-Bantu	12M	East
Swahili	swa	Niger-Congo-Bantu	98M	Southern, Central & East
Xhosa	xho	Niger-Congo-Nguni Bantu	19.2M	Southern
Yorùbá	yor	Niger-Congo-Volta-Niger	42M	West

Table 1: Language, family, number of speakers (Eberhard et al., 2020), and regions in Africa. Adapted from (Adelani et al., 2021b)

2 Related Work

2.1 Multilingual Machine Translation (MMT)

The great success of the encoder-decoder (Sutskever et al., 2014; Cho et al., 2014) NMT on bilingual datasets (Bahdanau et al., 2015; Vaswani et al., 2017; Barrault et al., 2019, 2020) inspired the extension of the original bilingual framework to handle more languages pairs simultaneously – leading to multilingual neural machine translation.

Works on multilingual NMT have progressed from sharing the encoder for one-to-many translation (Dong et al., 2015), many-to-one translation (Lee et al., 2017), sharing the attention mechanism across multiple language pairs (Firat et al., 2016a; Dong et al., 2015) to optimizing a single NMT model (with a universal encoder and decoder) for the translation of multiple language pairs (Ha et al., 2016; Johnson et al., 2017). The universal encoder-decoder approach constructs a shared vocabulary for all languages in the training set, and uses just one encoder and decoder for multilingual translation between language pairs. Johnson et al. (2017) proposed to use a single model and prepend

special symbols to the source text to indicate the target language. We adopt their model approach in this paper.

The current state of multilingual NMT, where a single NMT model is optimized for the translation of multiple language pairs (Firat et al., 2016a; Johnson et al., 2017; Lu et al., 2018; Aharoni et al., 2019; Arivazhagan et al., 2019b), has become very appealing for a number of reasons. It is scalable and easy to deploy or maintan (the ability of a single model to effectively handle all translation directions from N languages, if properly trained and designed, surpasses the scalability of $O(N^2)$ individually trained models using the traditional bilingual framework). Multilingual NMT can encourage knowledge transfer among related language pairs (Lakew et al., 2018; Tan et al., 2019) as well as positive transfer from higher-resource languages (Zoph et al., 2016; Neubig and Hu, 2018; Arivazhagan et al., 2019a; Aharoni et al., 2019; Johnson et al., 2017) due to its shared representation, improve low-resource translation (Ha et al., 2016; Johnson et al., 2017; Arivazhagan et al., 2019b; Xue et al., 2021) and enable zero-shot translation (i.e. direct translation between a language pair never seen during training) (Firat et al., 2016b; Johnson et al., 2017).

Despite the many advantages of multilingual NMT it suffers from certain disadvantages. Firstly, the output vocabulary size is typically fixed regardless of the number of languages in the corpus and increasing the vocabulary size is costly in terms of computational resources because the training and inference time scales linearly with the size of the decoder's output layer. For example, the training dataset for all the languages in our work gave a total vocabulary size of $1,683,884$ tokens ($1,519,918$ with every sentence lowercased) but we were constrained to a decoder vocabulary size of $250,000$.

Another pitfall of massively multilingual NMT is its poor zero-shot performance (Firat et al., 2016b; Arivazhagan et al., 2019a; Johnson et al., 2017; Aharoni et al., 2019), particularly compared to pivot-based models (two bilingual models that translate from source to target language through an intermediate language). Neural machine translation is heavily reliant on parallel data and so without access to parallel training data for zero-shot language pairs, multilingual models face the spurious correlation issue (Gu et al., 2019) and *off-target translation* (Johnson et al., 2017) where the model ignores the given target information and translates into a wrong language.

Some approaches to improve the performance (including zero-shot translation) of multilingual models have relied on leveraging the plentiful source and target side monolingual data that are available. For example, generating artificial parallel data with various forms of backtranslation (Sennrich et al., 2015) has been shown to greatly improve the overall (and zero-shot) performance of multilingual models (Firat et al., 2016b; Gu et al., 2019; Lakew et al., 2018; Zhang et al., 2020) as well as bilingual models (Edunov et al., 2018). Zhang et al. (2020) proposed random online backtranslation to enhance multilingual translation of unseen training language pairs.

Additionally, leveraging monolingual data by jointly learning to reconstruct the input while translating has been shown to improve neural machine translation quality (Févry and Phang, 2018; Lample et al., 2017; Cheng et al., 2016; Zhang and Zong, 2016). Siddhant et al. (2020) leveraged monolingual data in a semi-supervised fashion and reported three major results:

1. Using monolingual data significantly boosts the translation quality of low resource languages in multilingual models.

2. Self-supervision improves zero-shot translation quality in multilingual models.

3. Leveraging monolingual data with self-supervision provides a viable path towards adding new languages to multilingual models.

3 Data Methodology

Table 2 presents the size of the gathered and cleaned parallel sentences for each language direction. We devised preprocessing guidelines for each of our focus languages taking their linguistic properties into consideration. We used a maximum sequence length of 50 (due to computational resources) and a minimum of 2. In the following sections we will describe the data sources for the the parallel and monolingual corpora.

Parallel Corpora: As NMT models are very reliant on parallel data, we sought to gather more parallel sentences for each language direction in an effort to increase the size and domain of each language direction. To this end, our first source was JW300 (Agić and Vulić, 2019), a parallel corpus of

	Target Language							
	ibo	**fon**	**kin**	**xho**	**yor**	**swa**	**eng**	**fra**
ibo	-	3,179	52,685	58,802	134,219	67,785	85,358	57,458
fon	3,148	-	3,060	3,364	5,440	3,434	5,575	2,400
kin	53,955	3,122	-	70,307	85,824	83,898	77,271	62,236
xho	60,557	3,439	70,506	-	64,179	125,604	138,111	113,453
yor	133,353	5,485	83,866	62,471	-	117,875	122,554	97,000
swa	69,633	3,507	84,025	125,307	121,233	-	186,622	128,428
eng	87,716	5,692	77,148	137,240	125,927	186,122	-	-
fra	58,521	2,444	61,986	112,549	98,986	127,718	-	-

Table 2: Number of parallel samples for each language direction. We highlight the largest and smallest parallel samples. We see for example that much more research on machine translation and data collation has been carried out on swa⟷eng than fon⟷fra, attesting to the under-representation of some African languages.

over 300 languages with around 100 thousand biblical domain parallel sentences per language pair on average. Using OpusTools (Aulamo et al., 2020) we were able to get only very trustworthy translations by setting $t = 1.5$ (t is a threshold which indicates the confidence of the translations). We collected more parallel sentences from Tatoeba[1], kde4[2] (Tiedemann, 2012), and some English-based bilingual samples from MultiParaCrawl[3].

Finally, following pointers from the native speakers of these focus languages in the Masakhane community (∀ et al., 2020) to existing research on machine translation for African languages which open-sourced their parallel data, we assembled more parallel sentences mostly in the $\{en, fr\}$⟷AL direction.

From all this we created `MMTAfrica Test Set` (explained in more details in section 3.1), got $5,424,578$ total training samples for all languages directions (a breakdown of data size for each language direction is provided in Table 2) and $4,000$ for dev.

Monolingual Corpora: Despite our efforts to gather several parallel data from various domains, we were faced with some problems: 1) there was a huge imbalance in parallel samples across the language directions. In Table 2 we see that the ⊛ ⟷fon direction has the least amount of parallel sentences while ⊛ ⟷swa or ⊛ ⟷yor is made up of relatively larger parallel sentences. 2) the parallel sentences particularly for AL⟷AL span a very small domain (mostly biblical, internet)

We therefore set out to gather monolingual data from diverse sources. As our focus is on African languages, we collated monolingual data in only these languages. The monolingual sources and volume are summarized in Table 3.

Language(ID)	**Monolingual source**	**Size**
Xhosa (xho)	The CC100-Xhosa Dataset created by Conneau et al. (2019), and OpenSLR (van Niekerk et al., 2017)	158,660
Yoruba (yor)	Yoruba Embeddings Corpus (Alabi et al., 2020) and MENYO20k (Adelani et al., 2021a)	45,218
Fon/Fongbe (fon)	FFR Dataset (Dossou and Emezue, 2020), and Fon French Daily Dialogues Parallel Data (Dossou and Emezue, 2021a)	42,057
Swahili/Kiswahili (swa)	(Shikali and Refuoe, 2019)	23,170
Kinyarwanda (kin)	KINNEWS-and-KIRNEWS (Niyongabo et al., 2020)	7,586
Igbo (ibo)	(Ezeani et al., 2020)	7,817

Table 3: Monolingual data sources and sizes (number of samples).

3.1 Data Set Types in our Work

Here we elaborate on the different categories of data set that we (generated and) used in our work for training and evaluation.

- `FLORES Test Set`: This refers to the dev test set of 1012 parallel sentences in all 101 language directions provided by Goyal

[1] https://opus.nlpl.eu/Tatoeba.php
[2] https://huggingface.co/datasets/kde4
[3] https://www.paracrawl.eu/

et al. (2021)[4]. We performed evaluation on this test set for all language directions except ⊛ ⟷fon and ⊛ ⟷kin.

- `MMTAfrica Test Set`: This is a *substantial* test set we created by taking out a small but equal number of sentences from each parallel source domain. As a result, we have a set from a wide range of domains, while encompassing samples from many existing test sets from previous research. Although this set is small to be fully considered as a test set, we open-source it because it contains sentences from many domains (making it useful for evaluation) and we hope that it can be built upon, by perhaps merging it with other benchmark test sets (Abate et al., 2018; Abbott and Martinus, 2019; Reid et al., 2021).

- `Baseline Train/Test Set`: We first conducted baseline experiments with Fon, Igbo, English and French as explained in section 4.4.1. For this we created a special data set by carefully selecting a small subset of the FFR Dataset (which already contained parallel sentences in French and Fon), first automatically translating the sentences to English and Igbo, using the Google Translate API[5], and finally re-translating with the help of Igbo (7) and English (7) native speakers (we recognized that it was easier for native speakers to edit/tweak an existing translation rather than writing the whole translation from scratch). In so doing, we created a data set of $13,878$ translations in all 4 language directions.

  We split the data set into $12,554$ for training `Baseline Train Set`, 662 for dev and 662 for test `Baseline Test Set`.

4 Model and Experiments

4.1 Model

For all our experiments, we used the mT5 model (Xue et al., 2021), a multilingual variant of the encoder-decoder, transformer-based (Vaswani et al., 2017) "Text-to-Text Transfer Transformer" (T5) model (Raffel et al., 2019). In T5 pre-training, the NLP tasks (including machine translation) were cast into a "text-to-text" format – that is, a task where the model is fed some text prefix for context or conditioning and is then asked to produce some output text. This framework makes it straightforward to design a number of NLP tasks like machine translation, summarization, text classification, etc. Also, it provides a consistent training objective both for pre-training and finetuning. The mT5 model was pre-trained with a maximum likelihood objective using "teacher forcing" (Williams and Zipser, 1989). The mT5 model was also pretrained with a modification of the masked language modelling objective (Devlin et al., 2018).

We finetuned the `mt5-base` model on our many-to-many machine translation task. While Xue et al. (2021) suggest that higher versions of the mT5 model (*Large*, *XL* or *XXL*) give better performance on downstream multilingual translation tasks, we were constrained by computational resources to `mt5-base`, which has $580M$ parameters.

4.2 Setup

For each language direction $X \rightarrow Y$ we have its set of n parallel sentences $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ where x_i is the ith source sentence of language X and y_i is its translation in the target language Y.

Following the approach of Johnson et al. (2017) and Xue et al. (2021), we model translation in a text-to-text format. More specifically, we create the input for the model by prepending the target language tag to the source sentence. Therefore for each source sentence x_i the input to the model is $<Y_{tag}>\ x_i$ and the target is y_i. Taking a real example, let's say we wish to translate the Igbo sentence *Daalụ maka ikwu eziokwu nke Chineke* to English. The input to the model becomes *<eng> Daalụ maka ikwu eziokwu nke Chineke*.

4.3 Training

We have a set of language tags L for the languages we are working with in our multilingual many-to-many translation. In our baseline setup (section 4.4.1) $L = \{eng, fra, ibo, fon\}$ and in our final experiment (section 4.4.2) $L = \{eng, fra, ibo, fon, swa, kin, xho, yor\}$. We carried out many-to-many translation using all the possible directions from L except $eng \longleftrightarrow fra$. We skipped $eng \longleftrightarrow fra$ for this fundamental reason:

- our main focus is on African⟷African or $\{eng, fra\}$ ⟷African. Due to the high-

[4] https://dl.fbaipublicfiles.com/flores101/dataset/flores101_dataset.tar.gz

[5] https://cloud.google.com/translate

resource nature of English and French, adding the training set for $eng \longleftrightarrow fra$ would overshadow the learning of the other language directions and greatly impede our analyses. Our intuition draws from the observation of Xue et al. (2021) as the reason for *off-target* translation in the mT5 model: as English-based finetuning proceeds, the model's assigned likelihood of non-English tokens presumably decreases. Therefore since the `mt5-base` training set contained predominantly English (and after other European languages) tokens and our research is about AL$\longleftrightarrow$AL translation, removing the $eng \longleftrightarrow fra$ direction was our way of ensuring the model designated more likelihood to AL tokens.

4.3.1 Our Contributions

In addition to the parallel data between the African languages, we leveraged monolingual data to improve translation quality in two ways:

1. **our backtranslation (BT):** We designed a modified form of the random online backtranslation (Zhang et al., 2020) where instead of randomly selecting a subset of languages to backtranslate, we selected for each language num_bt sentences at random from the monolingual data set. This means that the model gets to backtranslate different (monolingual) sentences every backtranslation time and in so doing, we believe, improve the model's domain adaptation because it gets to learn from various samples from the whole monolingual data set. We initially tested different values of num_bt to find a compromise between backtranslation computation time and translation quality.

   Following research works which have shown the effectiveness of random beam-search over greedy decoding while generating backtranslations (Lample et al., 2017; Edunov et al., 2018; Hoang et al., 2018; Zhang et al., 2020), we generated num_sample prediction sentences from the model and randomly selected (with equal probability) one for our backtranslated sentence. Naturally the value of num_sample further affects the computation time (because the model has to produce num_sample different output sentences for each input sentence) and so we finally settled with $num\_sample = 2$.

2. **our reconstruction:** Given a monolingual sentence x^m from language m, we applied random swapping (2 times) and deletion (with a probability of 0.2) to get a noisy version $\hat{x}$. Taking inspiration from Raffel et al. (2019) we integrated the reconstruction objective into our model finetuning by prepending the language tag <m> to $\hat{x}$ and setting its target output to x^m.

4.4 Experiments

In all our experiments we initialized the pretrained `mT5-base` model using Hugging Face's AutoModelForSeq2SeqLM[6] and tracked the training process with Weights&Biases (Biewald, 2020). We used the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate (lr) of $3e^{-6}$ and transformer's $get\_linear\_schedule\_with\_warmup$[7] scheduler (where the learning rate decreases linearly from the initial lr set in the optimizer to 0, after a warmup period and then increases linearly from 0 to the initial lr set in the optimizer.)

4.4.1 Baseline

The goal of our baseline was to understand the effect of jointly finetuning with backtranslation and reconstruction on the African$\longleftrightarrow$African language translation quality in two scenarios: when the AL was initially pretrained on the multilingual model and contrariwise. Using Fon (which was not initially included in the pretraining) and Igbo (which was initially included in the pretraining) as the African languages for our baseline training, we finetuned our model on a many-to-many translation in all directions of $\{eng, fra, ibo, fon\}/eng \longleftrightarrow fra$ amounting to 10 directions. We used the `Baseline Train Set` for training and the `Baseline Test Set` for evaluation. We trained the model for only 3 epochs in three settings:

1. `BASE` : in this setup we finetune the model on only the many-to-many translation task: no backtranslation nor reconstruction.

2. `BT` : refers to finetuning with our backtranslation objective described in section 4.2. For our

[6] https://huggingface.co/transformers/model_doc/auto.html#transformers.AutoModelForSeq2SeqLM

[7] https://huggingface.co/transformers/main_classes/optimizer_schedules.html#transformers.get_linear_schedule_with_warmup

baseline, where we backtranslate using monolingual data in $\{ibo, fon\}$, we set $num\_bt = 500$. For our final experiments, we first tried with 500 but finally reduced to 100 due to the great deal of computation required.

For our baseline experiment, we ran one epoch normally and the remaining two with backtranslation. For our final experiments, we first finetuned the model on 3 epochs before continuing with backtranslation.

3. `BT&REC`: refers to joint backtranslation and reconstruction (explained in section 4.2) while finetuning. Two important questions were addressed – 1) the ratio, backtranslation : reconstruction, of monolingual sentences to use and 2) whether to use the same or different sentences for backtranslation and reconstruction. Bearing computation time in mind, we resolved to go with $500 : 50$ for our baseline and $100 : 50$ for our final experiments. We leave ablation studies on the effect of the ratio on translation quality to future work. For the second question we decided to randomly sample (with replacement) different sentences each for our backtranslation and reconstruction.

For our baseline, we used a learning rate of $5e^{-4}$, a batch size of 32 sentences, with gradient accumulation up to a batch of 256 sentences and an early stopping patience of 100 evaluation steps. To further analyse the performance of our baseline setups we ran `comparemt` [8] (Neubig et al., 2019) on the model's predictions.

4.4.2 MMTAfrica

MMTAfrica refers to our final experimental setup where we finetuned our model on all language directions involving all eight languages $L = \{eng, fra, ibo, fon, swa, kin, xho, yor\}$ except eng⟷fra. Taking inspiration from our baseline results we ran our experiment with our proposed `BT&REC` setting and made some adjustments along the way.

The long computation time for backtranslating (with just 100 sentences per language the model was required to generate around $3,000$ translations every backtranslation time) was a drawback. To mitigate the issue we parallelized the process using the multiprocessing package in Python[9]. We further slowly reduced the number of sentences for backtranslation (to 50, and finally 10).

Gradient descent in large multilingual models has been shown to be more stable when updates are performed over large batch sizes are used (Xue et al., 2021). To cope with our computational resources, we used gradient accumulation to increase updates from an initial batch size of 64 sentences, up to a batch gradient computation size of 4096 sentences. We further utilized PyTorch's DataParallel package[10] to parallelize the training across the GPUs. We used a learning rate (lr) of $3e^{-6}$

5 Results and Insights

All evaluations were made using spBLEU (sentencepiece (Kudo and Richardson, 2018) + sacreBLEU (Post, 2018)) as described in (Goyal et al., 2021). We further evaluated on the chrF (Popović, 2015) and TER metrics.

5.1 Baseline Results and Insights

Figure 1 compares the spBLEU scores for the three setups used in our baseline experiments. As a reminder, we make use of the symbol ⋄ to refer to any language in the set $\{eng, fra, ibo, fon\}$.

`BT` gives strong improvement over `BASE` (except in eng⟶ibo where it's relatively the same, and fra⟶ibo where it performs worse).

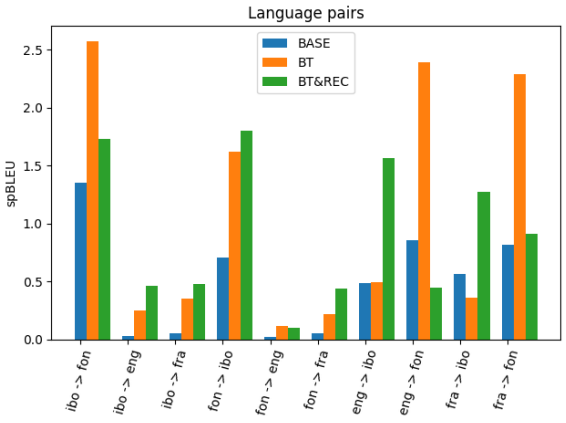


Figure 1: spBLEU scores of the 3 setups explained in section 4.4.1

When the target language is ***fon***, we observe a considerable boost in the spBLEU of the `BT` setting, which also significantly outperformed `BASE`

[8] https://github.com/neulab/compare-mt

[9] https://docs.python.org/3/library/multiprocessing.html

[10] https://pytorch.org/docs/stable/generated/torch.nn.DataParallel.html

and BT&REC. BT&REC contributed very little when compared with BT and sometimes even performed poorly (in eng⟶fon). We attribute this poor performance from the reconstruction objective to the fact that the mt5-base model was not originally pretrained on Fon. Therefore, with only 3 epochs of finetuning (and 1 epoch before introducing the reconstruction and backtranslation objectives) the model was not able to meaningfully utilize both objectives.

Conversely, when the target language is ***ibo*** BT&REC gives best results – even in scenarios where BT underperforms BASE (as is the case of fra⟶ibo and eng⟶ibo). We believe that the decoder of the model, being originally pretrained on corpora containing Igbo, was able to better use our reconstruction to improve translation quaity in ⋄ ⟶***ibo*** direction.

Drawing insights from fon⟷ibo **we offer the following propositions concerning AL⟷AL multilingual translation**:

- our backtranslation (section 4.2) from monolingual data improves the cross-lingual mapping of the model for low-resource African languages. While it is computationally expensive, our parallelization and decay of number of backtranslated sentences are some potential solutions towards effectively adopting backtranslation using monolingual data.

- Denoising objectives typically have been known to improve machine translation quality (Zhang and Zong, 2016; Cheng et al., 2016; Gu et al., 2019; Zhang et al., 2020; Xue et al., 2021) because they imbue the model with more generalizable knowledge (about that language) which is used by the decoder to predict better token likelihoods for that language during translation. This is a reasonable explanation for the improved quality with the BT&REC over BT in the ⋄ ⟶***ibo***. As we learned from ⋄ ⟶***fon***, using reconstruction could perform unsatisfactorily if not handled well. Some methods we propose are:

  1. For African languages that were included in the original model pretraining (as was the case of Igbo, Swahili, Xhosa, and Yorùbá in the mT5 model), using the BT&REC setting for finetuning produces best results. While we did not perform ablation studies on the data size ratio for backtranslation and reconstruction, we believe that our ratio of $2:1$ (in our final experiments) gives the best compromise on both computation time and translation quality.
  2. For African languages that were not originally included in the original model pretraining (as was the case of Kinyarwanda and Fon in the mT5 model), reconstruction together with backtranslation (especially at an early stage) only introduces more noise which could harm the cross-lingual learning. For these languages we propose:
     (a) first finetuning the model on only our reconstruction (described in section 4.2) for fairly long training steps before using BT&REC. This way, the initial reconstruction will help the model learn that language representation space and increase its the likelihood of tokens.

5.2 MMTAfrica Results and Insights

In Table 4, we compared MMTAfrica with the M2M MMT(Fan et al., 2020) benchmark results of Goyal et al. (2021) using the same test set they used – FLORES Test Set. On all language pairs except swa→eng (which has a comparable -2.76 spBLEU difference), we report an improvement from MMTAfrica (spBLEU gains ranging from $+0.58$ in swa⟶fra to $+19.46$ in fra⟶xho). The lower score of swa→eng presents an intriguing anomaly, especially given the large availability of parallel corpora in our training set for this pair. We plan to investigate this in further work.

In Table 5 we introduce benchmark results of MMTAfrica on MMTAfrica Test Set. We also put the test size of each language pair.

Interesting analysis about Fon (fon) and Yorùbá (yor): For each language, the lowest spBLEU scores in both tables come from the ⟶yor direction, except fon⟷yor (from Table 5) which interestingly has the highest spBLEU score compared to the other fon⟶ ⊛ directions. We do not know the reason for the very low performance in the ⊛ ⟶yor direction, but we offer below a plausible explanation about fon⟷yor.

The oral linguistic history of Fon ties it to the ancient Yorùbá kingdom (Barnes, 1997). Furthermore, in present day Benin, where Fon is largely

spoken as a native language, Yoruba is one of the indigenuous languages commonly spoken. [11] Therefore Fon and Yorùbá share some linguistic characteristics and we believe this is one logic behind the fon⟷yor surpassing other fon⟶ ⊛ directions.

This explanation could inspire transfer learning from Yorùbá, which has received comparably more research and has more resources for machine translation, to Fon. We leave this for future work.

Source	Target	spBLEU (FLORES)↑	spBLEU (Ours)↑	spCHRF↑	spTER↓
ibo	swa	4.38	**21.84**	37.38	71.48
ibo	xho	2.44	**13.97**	31.95	81.37
ibo	yor	1.54	**10.72**	26.55	75.72
ibo	eng	7.37	**13.62**	38.90	76.23
ibo	fra	6.02	**16.46**	35.10	75.48
swa	ibo	1.97	**19.80**	33.95	68.22
swa	xho	2.71	**21.71**	39.86	73.16
swa	yor	1.29	**11.68**	27.44	75.23
swa	eng	**30.43**	27.67	56.12	55.91
swa	fra	26.69	**27.27**	46.20	63.47
xho	ibo	3.80	**17.02**	31.30	70.66
xho	swa	6.14	**29.47**	44.68	63.21
xho	yor	1.92	**10.42**	26.77	76.25
xho	eng	10.86	**20.77**	48.69	64.09
xho	fra	8.28	**21.48**	40.65	69.31
yor	ibo	1.85	**11.45**	25.26	74.99
yor	swa	1.93	**14.99**	30.49	79.90
yor	xho	1.94	**9.31**	26.34	86.08
yor	eng	4.18	**8.15**	30.65	86.94
yor	fra	3.57	**10.59**	27.60	81.32
eng	ibo	3.53	**21.49**	37.24	65.68
eng	swa	26.95	**40.11**	53.13	52.80
eng	xho	4.47	**27.15**	44.93	67.77
eng	yor	2.17	**12.09**	28.34	74.74
fra	ibo	1.69	**19.48**	34.47	68.50
fra	swa	17.17	**34.21**	48.95	58.11
fra	xho	2.27	**21.73**	40.06	73.72
fra	yor	1.16	**11.42**	27.67	75.33

Table 4: Evaluation Scores of the Flores M2M MMT model and MMTAfrica on `FLORES Test Set`.

6 Conclusion and Future Work

In this paper, we introduced MMTAfrica, a multilingual machine translation model on 6 African Languages, which outperformed the M2M MMT model Fan et al. (2020). Our results and analyses, including a new reconstruction objective, give insights on MMT for African languages for future research. Moreover, we plan to launch the model on Masakhane MT and FFRTranslate in order to get human evaluation feedback from the actual speakers of the languages in the Masakhane community (Orife et al., 2020) and beyond.

In order to fully test the advantage of MMTAfrica, we plan to finish comparing it on direct and pivot translations with the Masakhane benchmark models (∀ et al., 2020). We also plan to perform human evaluation. All test sets, results, code and checkpoints will be released at https://github.com/edaiofficial/mmtafrica

7 Acknowledgments

The computational resources for running all experiments were provided by the FLORES compute grants[12] as well as additonal computational resources provided by Paco Guzman (Facebook AI) and Mila Quebec AI Institute. We express our profound gratitude to all who contributed in one way or the other towards the development of MMTAfrica including (in no order):

Mathias Müller (University of Zurich) for giving immense technical assistance in finetuning the model and advising us on best hyperparameter tuning practises.

Graham Neubig (Carnegie Mellon University) for explaining and setting up `comparemt` for us to better understand and evaluate the performance of our baseline models.

Angela Fan (FacebookAI) for guiding us during the shared task and taking the time to answer all our questions.

Julia Kreutzer (GoogleAI) for advising us on useful model comparison and evaluation techniques.

Colin Leong (University of Dayton) for painstakingly proof-reading the paper and helping us make it more reader-friendly.

Daria Yasafova (Technical University of Munich) and Yeno Gbenou (Drexel University) for additionally proof-reading the paper.

Finally the entire Masakhane community[13] for, among many other things, 1) guiding us to existing parallel and monolingual data set for the focus African languages, 2) explaining their (the focus African languages) important linguistic charteristics which helped us work better with them (like in preprocessing) and 3) performing (in the future) human evaluation on MMTAfrica.

Indeed it took a village to raise MMTAfrica[14].

[11] https://en.wikipedia.org/wiki/Benin (Last Accessed : 30.08.2021).

[12] http://www.statmt.org/wmt21/flores-compute-grants.html

[13] https://www.masakhane.io/

[14] 'It takes a village to raise a child' is an African proverb that means that an entire community of people with different expertise must provide for and interact positively with the child for the child to actually develop and reach the best possible potential.

Source	Target	Test size	spBLEU↑	spCHRF↑	spTER↓
ibo	swa	60	34.89	47.38	68.28
ibo	xho	30	36.69	50.66	59.65
ibo	yor	30	11.77	29.54	129.84
ibo	kin	30	33.92	46.53	67.73
ibo	fon	30	35.96	43.14	63.21
ibo	eng	90	37.28	60.42	62.05
ibo	fra	60	30.86	44.09	69.53
swa	ibo	60	33.71	43.02	60.01
swa	xho	30	37.28	52.53	55.86
swa	yor	30	14.09	27.50	113.63
swa	kin	30	23.86	42.59	94.67
swa	fon	30	23.29	33.52	65.11
swa	eng	60	35.55	60.47	47.32
swa	fra	60	30.11	48.33	63.38
xho	ibo	30	33.25	45.36	62.83
xho	swa	30	39.26	53.75	53.72
xho	yor	30	22.00	38.06	70.45
xho	kin	30	30.66	46.19	74.70
xho	fon	30	25.80	34.87	65.96
xho	eng	90	30.25	55.12	62.11
xho	fra	30	29.45	45.72	61.03
yor	ibo	30	25.11	34.19	74.80
yor	swa	30	17.62	34.71	85.18
yor	xho	30	29.31	43.13	66.82
yor	kin	30	25.16	38.02	72.67
yor	fon	30	31.81	37.45	63.39
yor	eng	90	17.81	41.73	93.00
yor	fra	30	15.44	30.97	90.57
kin	ibo	30	31.25	42.36	66.73
kin	swa	30	33.65	46.34	72.70
kin	xho	30	20.40	39.71	89.97
kin	yor	30	18.34	33.53	70.43
kin	fon	30	22.43	32.49	67.26
kin	eng	60	15.82	43.10	96.55
kin	fra	30	16.23	33.51	91.82
fon	ibo	30	32.36	46.44	61.82
fon	swa	30	29.84	42.96	72.28
fon	xho	30	28.82	43.74	66.98
fon	yor	30	30.45	42.63	60.72
fon	kin	30	23.88	39.59	78.06
fon	eng	30	16.63	41.63	69.03
fon	fra	60	24.79	43.39	82.15
eng	ibo	90	44.24	54.89	63.92
eng	swa	60	49.94	61.45	47.83
eng	xho	120	31.97	49.74	72.89
eng	yor	90	23.93	36.19	84.05
eng	kin	90	40.98	56.00	76.37
eng	fon	30	27.19	36.86	62.54
fra	ibo	60	36.47	46.93	59.91
fra	swa	60	36.53	51.42	55.94
fra	xho	30	34.35	49.39	60.30
fra	yor	30	7.26	25.54	124.53
fra	kin	30	31.07	42.26	81.06
fra	fon	60	31.07	38.72	75.74

Table 5: Benchmark Evaluation Scores on `MMTAfrica Test Set`

References

Solomon Teferra Abate, Michael Melese, Martha Yifiru Tachbelie, Million Meshesha, Solomon Atinafu, Wondwossen Mulugeta, Yaregal Assabie, Hafte Abera, Binyam Ephrem, Tewodros Abebe, Wondimagegnhue Tsegaye, Amanuel Lemma, Tsegaye Andargie, and Seifedin Shifaw. 2018. Parallel corpora for bi-lingual English-Ethiopian languages statistical machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3102–3111, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jade Abbott and Laura Martinus. 2019. Benchmarking neural machine translation for Southern African languages. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 98–101, Florence, Italy. Association for Computational Linguistics.

Jade Z. Abbott and Laura Martinus. 2018. Towards neural machine translation for african languages.

David I. Adelani, Dana Ruiter, Jesujoba O. Alabi, Damilola Adebonojo, Adesina Ayeni, Mofe Adeyemi, Ayodele Awokoya, and Cristina España-Bonet. 2021a. The effect of domain and diacritics in yorùbá-english neural machine translation.

David Ifeoluwa Adelani, Jade Z. Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen Hassan Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba O. Alabi, Seid Muhie Yimam, Tajuddeen Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin P. Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane Mboup, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima Diop, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021b. Masakhaner: Named entity recognition for african languages. *CoRR*, abs/2103.11811.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation.

In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesujoba Alabi, Kwabena Amponsah-Kaakyire, David Adelani, and Cristina España-Bonet. 2020. Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2754–2762, Marseille, France. European Language Resources Association.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019b. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, and Jörg Tiedemann. 2020. OpusTools and parallel corpus diagnostics. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3782–3789. European Language Resources Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

S.T. Barnes. 1997. *Africa's Ogun: Old World and New*. African systems of thought. Indiana University Press.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

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, Berlin, Germany. Association for Computational Linguistics.

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. ACL.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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. Association for Computational Linguistics.

Bonaventure F. P. Dossou and Chris C. Emezue. 2020. FFR V1.0: fon-french neural machine translation. *CoRR*, abs/2003.12111.

Bonaventure F. P. Dossou and Chris C. Emezue. 2021a. Crowdsourced phrase-based tokenization for low-resourced neural machine translation: The case of fon language. *CoRR*, abs/2103.08052.

Bonaventure F. P. Dossou and Chris C. Emezue. 2021b. Okwugbé: End-to-end speech recognition for fon and igbo. *CoRR*, abs/2103.07762.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig (eds.). 2020. Ethnologue: Languages of the world. twenty-third edition.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Ignatius Ezeani, Paul Rayson, Ikechukwu E. Onyenwe, Chinedu Uchechukwu, and Mark Hepple. 2020. Igbo-english machine translation: An evaluation benchmark. *CoRR*, abs/2004.00648.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Thibault Févry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. *CoRR*, abs/1809.02669.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. 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.

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 *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

∀, Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Tajudeen Kolawole, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddee Hassan Muhammad, Salomon Kabongo, Salomey Osei, et al. 2020. Participatory research for low-resourced machine translation: A case study in african languages. *Findings of EMNLP*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

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, Melbourne, Australia. 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, 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.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

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.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Brussels, Belgium. Association for Computational Linguistics.

Laura Martinus and Jade Z. Abbott. 2019. A focus on neural machine translation for african languages. *CoRR*, abs/1906.05685.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. *CoRR*, abs/1903.07926.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Rubungo Andre Niyongabo, Qu Hong, Julia Kreutzer, and Li Huang. 2020. KINNEWS and KIRNEWS: Benchmarking cross-lingual text classification for Kinyarwanda and Kirundi. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5507–5521, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Linda Nordling. 2018. How decolonization could reshape south african science. *Nature*, 554(7691):159–162.

Evander Nyoni and Bruce A. Bassett. 2021. Low-resource neural machine translation for southern african languages. *CoRR*, abs/2104.00366.

Iroro Orife, Julia Kreutzer, Blessing Sibanda, Daniel Whitenack, Kathleen Siminyu, Laura Martinus, Jamiil Toure Ali, Jade Z. Abbott, Vukosi Marivate, Salomon Kabongo, Musie Meressa, Espoir Murhabazi, Orevaoghene Ahia, Elan Van Biljon, Arshath Ramkilowan, Adewale Akinfaderin, Alp Öktem, Wole Akin, Ghollah Kioko, Kevin Degila, Herman Kamper, Bonaventure Dossou, Chris Emezue, Kelechi Ogueji, and Abdallah Bashir. 2020. Masakhane - machine translation for africa. *CoRR*, abs/2003.11529.

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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Machel Reid, Junjie Hu, Graham Neubig, and Yutaka Matsuo. 2021. Afromt: Pretraining strategies and reproducible benchmarks for translation of 8 african languages.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Shivachi Casper Shikali and Mokhosi Refuoe. 2019. Language modeling data for swahili.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Xu Chen, Sneha Reddy Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. *CoRR*, abs/2005.04816.

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, page 3104–3112, Cambridge, MA, USA. MIT Press.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. 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).

Daniel van Niekerk, Charl van Heerden, Marelie Davel, Neil Kleynhans, Oddur Kjartansson, Martin Jansche, and Linne Ha. 2017. Rapid development of TTS corpora for four South African languages. In *Proc. Interspeech 2017*, pages 2178–2182, Stockholm, Sweden.

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.

Sarah Wild. 2021. African languages to get more bespoke scientific terms. *Nature*, 596(7873):469–470.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. *CoRR*, abs/2004.11867.

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.

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 LMU Munich System for the WMT 2021 Large-Scale Multilingual Machine Translation Shared Task

Wen Lai and **Jindřich Libovický** and **Alexander Fraser**
Center for Information and Language Processing, LMU Munich, Germany
{lavine, libovicky, fraser}@cis.lmu.de

Abstract

This paper describes the submission of LMU Munich to the WMT 2021 multilingual machine translation task for small track #1, which studies translation between 6 languages (Croatian, Hungarian, Estonian, Serbian, Macedonian, English) in 30 directions. We investigate the extent to which bilingual translation systems can influence multilingual translation systems. More specifically, we trained 30 bilingual translation systems, covering all language pairs, and used data augmentation techniques such as back-translation and knowledge distillation to improve the multilingual translation systems. Our best translation system scores 5 to 6 BLEU higher than a strong baseline system provided by the organizers (Goyal et al., 2021). As seen in the Dynalab leaderboard, our submission is the only fully constrained submission that uses only the corpus provided by the organizers and does not use any pre-trained models.

1 Introduction

Neural Machine Translation (NMT) (Vaswani et al., 2017) has been shown to be effective with rich and in-domain bilingual parallel corpora. Although the NMT model obtained promising performances for high resource language pairs, it is hardly feasible to train translation models for all directions of the language pairs since the training progress is time- and resource-consuming. Recent work has shown the effectiveness of multilingual neural machine translation (MNMT), which aims to handle the translation from multiple source languages into multiple target languages with a single unified model (Johnson et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019; Zhang et al., 2020; Fan et al., 2021; Goyal et al., 2021).

The MNMT model dramatically reduces training and serving costs. It is faster to train a MNMT model than to train bilingual models for all language pairs in both directions, and MNMT significantly simplifies deployment in production systems (Johnson et al., 2017; Arivazhagan et al., 2019). Further, parameter sharing across different languages encourages knowledge transfer, which improves low-resource translation directions and potentially enables zero-shot translation (i.e., direct translation of a language pair not seen during training) (Ha et al., 2017; Gu et al., 2019; Ji et al., 2020; Zhang et al., 2020).

We participate in the WMT 2021 multilingual machine translation task for small track #1. The task aims to train a multilingual model to translate 5 Central/East European languages (Croatian, Hungarian, Estonian, Serbian, Macedonian) and English in 30 directions. The multilingual systems presented in this paper are based on the standard paradigm of MNMT proposed by Johnson et al. (2017), which prefixes the source sentence with a special token to indicate the desired target language and does not change the target sentence at all. Language tags are typically used in MNMT to identify the language to translate to. A language code, in the form of a two- or three-character identification such as `en` for English, is the main constituent of a language tag and is provided by the ISO 639 standard[1] (International Organization for Standardization, nd). Following ISO 639 standard, `en` indicates English, `mk` indicates Macedonian, `sr` indicates Serbian, `et` indicates Estonian, `hr` indicates Croatian and `hu` indicates Hungarian in this paper.

Compared with the other three submissions to the task, our submissions have the following advantages:

- Our submissions are fully constrained, which means we using the data only provided by the organizer, and do not use models pre-trained on extra data.
- Our model only has 313M parameters, which

[1] https://en.wikipedia.org/wiki/ISO_639

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 412–417
November 10–11, 2021.

	Whole	Select
No filter	387M	71M
+ punctuation filter	384M	71M
+ deduplicated filter	304M	44M
+ `langid` filter	302M	43M
+ length filter	274M	42M

Table 1: Number of sentences in bitext datasets (total in 15 directions) for different filtering schemes. **Whole** denotes the use of all data provided by the organizers, **Select** denotes the use of data selection.

is smaller than the other submissions.

2 Data

The training data provided by the organizers come from the public available Opus repository (Tiedemann, 2012), which contains data of mixed quality from a variety of domains (WMT-News, TED, QED, OpenSubtitles, etc.). In addition to the bilingual parallel corpora, in-domain Wikipedia monolingual data for each language is provided. The validation and test sets are obtained from the Flores 101 evaluation benchmark (Goyal et al., 2021), which consists of 3001 sentences extracted from English Wikipedia covering a variety of different topics and domains. See Table 1 for details on data used for training our systems.

2.1 Data Preprocessing

To prepare the data for training, we used the following steps to process all of the corpora:

1. The datasets were truecased and the punctuation was normalized with standard scripts from the Moses toolkit[2](Koehn et al., 2007).

2. Sentences containing 50% punctuation are removed.

3. Duplicate sentences are removed.

4. We used a language detection tool[3] (`langid`) to filter out sentences with mixed language.

5. SentencePiece[4] (Kudo and Richardson, 2018) was used to produce subword units. We trained a model with 0.9995 character coverage to have sufficient coverage of character-based languages.

6. The length filtering removes sentences that are too long (more than 250 subwords after segmentation with Sentencepiece), sentences with a mismatched length ratio (more than 3.0) between source and target language are removed.

2.2 Data Selection

Data selection (Moore and Lewis, 2010; Axelrod et al., 2011; Gascó et al., 2012), aims to select the most relevant sentences from the out-of-domain corpora, which improved the in-domain translation performance. The training data provided by the organizers is large scale and contains multiple domains. Therefore, the data selection becomes a key factor affecting the performance of MNMT. Preliminary experiments (see in Table 1 model #3 and model #4) showed that the performance of using all corpora provided by the organizer was poor. Following the original paper (Goyal et al., 2021), we selected three data sources (CCAligned, MultiCCAligned, WikiMatrix) for further experimentation.

3 Method Description

We first trained bilingual translation models with 30 directions for all language pairs. Next, we trained a single multilingual model that can translate all language pairs. Finally, we use back-translation and knowledge distillation technologies to further improve the performance of the multilingual translation system. The details of these components are outlined next.

3.1 Bilingual NMT Model

We use Transformer (Vaswani et al., 2017) architecture for all bilingual models. To achieve the best BLEU score on the validation dataset, random search was used to select the hyperparameters since the datasets are in different sizes. We segment the data into subword units using SentencePiece jointly learned for all languages. The details of selected hyper-parameters are listed in Section 4.1.

3.2 Multilingual NMT Model

The multilingual model architecture is identical to the bilingual NMT model. To train multilingual models, we used a simple modification to the

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer

[3] https://fasttext.cc/docs/en/language-identification.html

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

source sentence proposed by Johnson et al. (2017) which introduce an artificial token at the beginning of the source sentence indicating the target language (Johnson et al., 2017). For instance, for the English-Macedonian (en→mk) translation direction, we insert a token like <2mk> at the beginning of all English sentences and do not change the Macedonian sentences.

3.3 Back Translation

Back-translation (BT) (Sennrich et al., 2016) is a simple and effective data augmentation technique, which makes use of monolingual corpora and has proven to be effective. Back-translation first trains a target-to-source system that is used to translate monolingual target data into source sentences, resulting in a pseudo-parallel corpus. Then we mix the pseudo-parallel corpus with the authentic parallel data and train the the desired source-to-target translation system. Zhang et al. (2020) has shown how BT can be useful for multilingual MT.

After generating the pseudo parallel corpus, we tag our BT data by adding an artificial token <BT> at the beginning of the source sentence (Caswell et al., 2019), which indicates that the data is generated by back-translation.

3.4 Knowledge Distillation

Knowledge Distillation (KD) is a commonly used technique to improve model performance. The standard KD training (Kim and Rush, 2016) derives a student model from a teacher model by training the student model to mimic the outputs of the teacher. We follow a recent approach to KD proposed by Wang et al. (2021), which uses selection at the batch level and at the global level to choose suitable samples for distillation.

4 Experiments

4.1 Training Details

We use the Transformer architecture (Vaswani et al., 2017) as implemented in fairseq[5] (Ott et al., 2019). For training NMT and MNMT systems, we use the `Transformer-Big` architecture (hidden state 1024, feed-forward layer 4096, 16 attention heads, 6 encoder layers, 6 decoder layers). For optimization, we follow the default settings from the original paper (Vaswani et al., 2017) and used the Adam optimizer with a learning rate of 0.0003. To prevent overfitting, we applied a dropout of 0.3 on all layers. At the time of inference, a beam search of size 5 is used to balance the decoding time and accuracy of the search. The number of warm-up steps was set to 4000 and the vocabulary size is 133k. In addition, we set a length penalty factor of 1.7 to maintain a balance between long and short sentences. The batch size is set to 128 during decoding. We trained our models for approximately 3 weeks on one machine with 8 NVIDIA GTX 2080 Ti 11GB GPUs.

Because of the problems of the international tokenization in the standard BLEU score, the organizers used sentence-piece BLEU (spBLEU)[6] (Goyal et al., 2021) as the official evaluation metric which operates on strings segmented using a SentencePiece model. Recently, the BLEU score was criticized as an unreliable automatic metric (Mathur et al., 2020; Kocmi et al., 2021). Therefore, we also evaluate our models using chrF (Popović, 2015) and BERTScore (Zhang et al., 2019).

4.2 Systems

All of our systems described in Section 3.2 are listed as follows:

Flores. As a baseline system, we use the pretrained models public available by Flores teams. We use `flores101_mm100_615M` tested on the devtest datasets as our baseline.

Bilingual. We trained the bilingual models using standard `Transformer-Big` architecture for 6 languages in 30 directions. The hyperparameters used are discussed in Section 4.1.

Multilingual. We trained the multilingual translation model using standard `Transformer-Big` architecture and a specific language token to indicate the desired translation target language.

Tagged BT. We augment the training data by exploring the monolingual corpus using back-translation proposed by Caswell et al. (2019), with tagged back-translated source sentences with an extra token <BT>.

Selective KD. We focused on selective knowledge distillation proposed by Wang et al. (2021), which uses batch-level and global-level selections to pick suitable samples for distillation.

4.3 Results

The results of our systems on the devtest dataset are presented in Table 2. For models 1–4, we observed

[5] https://github.com/pytorch/fairseq

[6] https://github.com/ngoyal2707/sacrebleu/tree/adding_spm_tokenized_bleu

#	Systems	spBLEU	chrF	BERTScore	BEST BLEU
0	Flores	28.0	0.528	0.867	sr-mk (36.0)
1	Bilingual$_{whole}$	21.1	0.477	0.831	en-mk (31.3)
2	Bilingual$_{select}$	28.4	0.533	0.863	sr-en (40.6)
3	Multilingual$_{whole}$	16.7	0.431	0.827	sr-en (26.1)
4	Multilingual$_{select}$	30.9	0.555	0.874	sr-en (40.0)
5	Multilingual$_{select}$ + TaggedBT(Multilingual$_{select}$)	30.7	0.548	0.873	sr-en (40.5)
6	Multilingual$_{select}$ + TaggedBT(Bilingual$_{select}$)	**32.3**	**0.562**	**0.879**	sr-en (41.5)
7*	Multilingual$_{select}$ + TaggedBT(Bilingual$_{select}$) + KD$_{batch}$	33.2	0.572	0.883	sr-en (42.0)
8*	Multilingual$_{select}$ + TaggedBT(Bilingual$_{select}$) + KD$_{global}$	**33.9**	**0.576**	**0.887**	sr-en (42.4)

Table 2: The automatic evaluation metrics on devtest data. **spBLEU**, **chrF**, **BERTScore** denotes the average scores of spBLEU, chrF and BERTScore respectively, **BEST BLEU** denotes the language pair with the best BLEU score. Systems with subscript *whole* denote the use of all data provided by the organizers, and systems with subscript *select* denote the use of data selection. Model #6 is our primary system submitted to the Dynalab leaderboard. Systems 7* and 8* were trained after the shared task and were not used for the final submission.

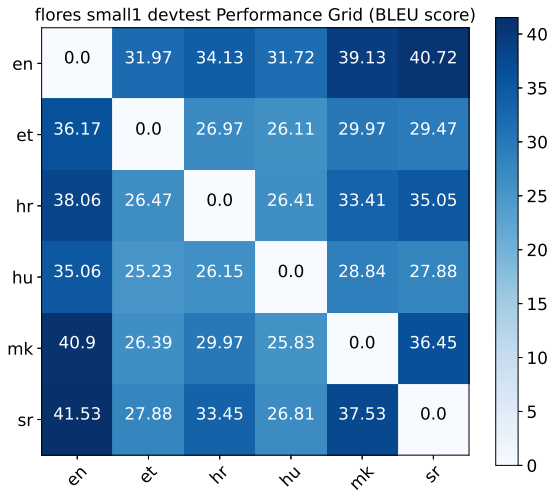


Figure 1: spBLEU scores on devtest data in 30 directions

that the amount of training data is not proportional to the performance of the model for the bilingual or multilingual translation model. The training data provided by the organizers contains multiple domains and does not match the dev/devtext/test data domain. Therefore, we apply the data selection methods to select data-relevant data from the training dataset to do the following experiments. Our multilingual model (#4) performs competitively with the Flores strong baseline (Model #0).

After these initial experiments, we explored how the bilingual models can be used to improve the multilingual model. More specifically, we use the Bilingual$_{select}$ model (#2) and Multilingual$_{select}$ model (#4) to back-translate the relevant monolingual corpora, and then we use the back-translations to train a new multilingual model. Although the overall performance of the Multilingual model (#4) is better than the Bilingual model (#2), back-translation using the Bilingual model (model #6) is better than back-translation using the Multilingual model (model #5). The possible reason is that the multilingual BT is in fact a form of self-training, but bilingual BT uses separate models, which means the knowledge obtained from bilingual BT models is more independent of the knowledge already learned by the baseline multilingual BT model.

Knowledge Distillation further improves performance slightly (Model # 7* and Model # 8*). Based on Model # 6, selective KD (Wang et al., 2021) is added to further improve the performance of the multilingual system.

Our best systems were outperformed by two other shared task submissions, which however used models pre-trained on additional data sources.

The performance grid of our best system (Model # 8*) is presented in Figure 1. We see from the results that the `sr-en` language pair produced the best results in terms of spBLEU score while the `hu-hr` language pair scored the lowest.

5 Conclusions

In this paper, we presented the LMU Munich system for the WMT 2021 Large-scale Multilingual Translation shared task for small track #1. The task evaluates translation between five central/eastern European languages and English, in total 30 translation directions. The system we submitted was fully constrained, using only the data provided by the organizers and not using any pre-trained model. The experiments show that back-translation and knowledge distillation techniques are effective for training multilingual machine translation systems.

6 Acknowledgments

We would like to thank the Flores team for the organization and for the grant of computer credits that we used in our experiments. This work was supported by funding to Wen Lai from LMU-CSC (China Scholarship Council) Scholarship Program (CSC, 202006390016). This work has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement #640550). This work was also supported by the DFG (grant FR 2829/4-1). We thank the other members of the machine translation group at CIS, LMU Munich, for their ideas and feedback.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

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, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 152–161, Avignon, France. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2017. Effective strategies in zero-shot neural machine translation. *arXiv preprint arXiv:1711.07893*.

Baijun Ji, Zhirui Zhang, Xiangyu Duan, Min Zhang, Boxing Chen, and Weihua Luo. 2020. Cross-lingual pre-training based transfer for zero-shot neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 115–122.

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.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *arXiv preprint arXiv:2107.10821*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. 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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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, Lisbon, Portugal. 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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, 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.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Back-translation for Large-Scale Multilingual Machine Translation

Baohao Liao **Shahram Khadivi** **Sanjika Hewavitharana**
eBay Inc.
{baliao|skhadivi|shewavitharana}@ebay.com

Abstract

This paper illustrates our approach to the shared task on large-scale multilingual machine translation in the sixth conference on machine translation (WMT-21). In this work, we aim to build a single multilingual translation system with a hypothesis that a universal cross-language representation leads to better multilingual translation performance. We extend the exploration of different back-translation methods from bilingual translation to multilingual translation. Better performance is obtained by the constrained sampling method, which is different from the finding of the bilingual translation. Besides, we also explore the effect of vocabularies and the amount of synthetic data. Surprisingly, the smaller size of vocabularies perform better, and the extensive monolingual English data offers a modest improvement. We submitted to both the small tasks and achieve the second place. The code and trained models are available at https://github.com/BaohaoLiao/multiback.

1 Introduction

Bilingual neural machine translation (NMT) systems have achieved decent performance with the help of Transformer (Vaswani et al., 2017). One of the most exciting recent trends in NMT is training a single system on multiple languages at once (Johnson et al., 2017b; Aharoni et al., 2019a; Zhang et al., 2020; Fan et al., 2020). This is a powerful paradigm for two reasons: simplifying system development and deployment, and improving the translation quality on low-resource language pairs by transferring similar knowledge from high-resource languages.

This paper describes our experiments on the task of large-scale multilingual machine translation in WMT-21. We primarily focus on the small tasks, especially on Small Task 2 which has a small amount of training data. Small Task 1 contains five Central/East European languages and English, having 30 translation directions. Similarly, Small Task 2 contains five South East Asian languages and English, also having 30 translation directions.

In this work, we mainly concentrate on different back-translation methods (Sennrich et al., 2016a; Edunov et al., 2018; Graça et al., 2019) for multilingual machine translation, including beam search and other sampling methods. Along with it, we also explore the effect of different sizes of vocabularies and the effect of various amounts of synthetic data. On this large-scale multilingual machine translation task, we achieved the second place for both small tasks, obtaining 34.96 and 33.34 average sp-BLEU scores (Goyal et al., 2021) on the hidden test set for the Small Task 1 and 2, respectively.

2 Related Work

Multilingual Neural Machine Translation has received increasing attention recently. Since Dong et al. (2015) extended the traditional bilingual NMT to one-to-many translation, there has been a massive increase in work on MT systems that involve more than two languages (Dabre et al., 2017; Choi et al., 2018; Chu and Dabre, 2019). The recent research on multilingual NMT can be split into two directions: developing language specific components (Kim et al., 2019; Escolano et al., 2020) and training a single model with extensive training data, including parallel and monolingual data (Fan et al., 2020). Here, we continue to explore the second research direction, trying to built a single multilingual NMT model for simple industrial deployment.

Back-translation (Sennrich et al., 2016a) has been proven as a powerful technique to leverage monolingual data for improving low-resource language pairs. Edunov et al. (2018) and Graça et al. (2019) explore different sampling methods for bilingual back-translation, including beam search, constrained and unconstrained sampling. Constrained sampling randomly predicts the next word

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 418–424

within some candidates that have higher prediction probability. And unconstrained sampling randomly predicts the next words from the whole vocabulary without caring for the output distribution. In this paper, we extend their exploration to the realm of multilingualism, where similar languages affect the results.

3 Experimental Setup

3.1 Data

The organizer offers parallel and monolingual data for Small Task 1 and 2. Table 1 shows the size of the data in terms of the number of sentences for each language. There are five extra sets for evaluation, i.e. dev, devtest, hidden dev, hidden devtest and test sets. The dev set with 997 parallel sentences among all language pairs and the devtest set with 1,012 parallel sentences are public. Whereas, the hidden dev and hidden devtest sets are invisible to the participants and used for the first submission period. The hidden test set is also invisible and used for the final ranking.

Pre-processing is done by a regular Moses toolkit (Koehn et al., 2007) pipeline that involves tokenization, byte pair encoding and removing long sentences. We borrow the 256K vocabularies from the organizer's pretrained model and the 128K vocabularies from M2M_100 (Fan et al., 2021), one shared vocabularies among all languages. Our submissions only use the 256K vocabularies, and the 128K vocabularies is used for ablation experiments.

We also perform back-translation on the monolingual data, and only accept the synthetic sentence pair whose length is less than 250 words, and whose length ratio between the source and target sentence length is less than 1.8. In order to balance the volume across different languages, we apply temperature sampling $\tilde{D}_i = (D_i / \sum_j D_j)^{1/T}$ with $T = 5$ over the dataset, where D_i is the number of sentences in the i_{th} language.

3.2 Model

All our models are built using the fairseq implementation (Ott et al., 2019) of the Transformer architecture (Vaswani et al., 2017). Multilingual models are built using the same technique as Johnson et al. (2017a) and Aharoni et al. (2019b), namely adding a language label to the target sentence.

We apply three types of architectures, i.e. *Trans_small*, *Trans_base* and *Trans_big*. The detailed settings of these architectures are shown in Table 2. The parameters of all architectures are in the half-precision floating-point format.

Small Task 1		**Small Task 2**	
Language	**#sent.**	**Language**	**#sent.**
en-et	35.7M	en-id	54.1M
en-hr	63.7M	en-jv	3.0M
en-hu	83.9M	en-ms	13.4M
en-mk	2.7M	en-ta	2.1M
en-sr	48.3M	en-tl	13.6M
et-hr	13.6M	id-jv	780.1K
et-hu	21.5M	id-ms	4.9M
et-mk	3.1M	id-ta	500.8K
et-sr	11.3M	id-tl	2.7M
hr-hu	31.2M	jv-ms	434.7K
hr-mk	4.4M	jv-ta	66.0K
hr-sr	28.4M	jv-tl	817.1K
hu-mk	4.1M	ms-ta	372.6K
hu-sr	31.2M	ms-tl	1.4M
mk-sr	4.2M	ta-tl	563.3K
en	126.4M	en	126.4M
et	3.0M	id	5.5M
hr	3.1M	jv	405.8K
hu	9.2M	ms	1.9M
mk	1.9M	ta	2.1M
sr	4.7M	tl	414.1K

Table 1: Number of sentences of the parallel and monolingual data used for two small tasks. The monolingual English data for the two small tasks are the same.

All our submissions on the shared task leaderboard are *Trans_base*, due to the memory and time limit of the evaluation system. *Trans_small* is mainly used for the ablation experiments. And the pretrained *Trans_big* from M2M_100 (Fan et al., 2021) is finetuned on the parallel corpus to generate high-quality synthetic sentences.

3.3 Optimization and Evaluation

The following hyper-parameter configuration is used: Adam optimizer with $\beta_1 = 0.90$, $\beta_2 = 0.98$, a weight-decay of 0.0001, the label smoothed cross-entropy criterion with a label smoothing of 0.1, an initial learning rate of 0.0003 with the inverse square root lr-scheduler and warmup updates of 2,500 steps. The batch size (the number of tokens) is 4096×32 for *Trans_small*, and 2048×64 for *Trans_base* and *Trans_big*.

For ablation experiments, we continue to train the pretrained *Trans_small* offered by the organizer

Model	Trans_small	Trans_base	Trans_big
#vocabularies	256K	256K	128K
Word representation size	512	1,024	1,024
Feed-forward layer dimension	2,048	4,096	8,192
#prenormed encoder/ decoder layer	6	12	24
#attention head	16	16	16
Dropout rate	0.1	0.1	0.1
Layer dropout rate	0.05	0.05	0.05
#parameters	175M	615M	1.2B

Table 2: Settings of different pretrained models. Pretrained *Trans_small* and *Trans_base* are provided by the organizer. And pretrained *Trans_big* is from Fan et al. (2021).

on the given parallel dataset for one epoch. When combining both parallel and synthetic data, we further train the model finetuned on the parallel data for another one epoch. For the final submissions, we train a pretrained *Trans_base* for two epochs instead of one epoch. Pretrained *Trans_big* from M2M_100 is only further trained on parallel data for two epochs to generate high-quality synthetic data. Even though we only train these models for a few epochs, they seems converged quite well according to the spBLEU curve during validation.

The model is validated every 3,000 steps on the dev set and saved. We use the beam search with a beam size of five, and stop translation when $l_{tgt} = 1.5 * l_{src} + 20$, where l_{src} and l_{tgt} are the source and target sentence length, respectively. The evaluation metric is BLEU based on sentence piece tokenization (spBLEU) (Goyal et al., 2021). We submit the average checkpoint of the last 15 checkpoints to the evaluation system. While for the ablation experiment, we use the best performed model on the dev set.

4 Results

4.1 The Role of Vocabularies

There are two pretrained vocabularies, the one with the size of 256K from the organizer and the one with the size of 128K from M2M_100 (Fan et al., 2021). To evaluate which vocabulary is the better one, we train two *Trans_smalls* with these two vocabularies from scratch on the parallel data of Small Task 2 for five epochs. To make the parameter sizes of these two models comparable, we set the following hyper-parameter for the model with the 128K vocabularies: 5 pre-normed encoder and decoder layers with a word representation size of 768 and a feed-forward layer dimension of 3072, resulting to 181M parameters. The other settings stay the same with *Trans_small* (with 256K vocabularies).

Model	Ave. spBLEU
128K *Trans_small* (scratch)	23.14
256K *Trans_small* (scratch)	21.65
256K *Trans_small* (pretrained)	23.72

Table 3: Average spBLEU on the devtest set of Small Task 2 for the models with different vocabularies.

Model	Ave. spBLEU
1st finetuned on parallel data	28.27
2nd finetuned on synthetic data	32.16
3rd finetuned on synthetic data	33.01

Table 4: Average spBLEU on the devtest set of Small Task 2 for *Trans_base* on different finetuning steps. These three models are iteratively trained. *Trans_base* is first finetuned on the parallel data, and then finetuned on the combination of the parallel data and the synthetic data generated by *Trans_big*, and finally finetuned on the combination of the parallel data and the synthetic data generated by the 2nd step *Trans_base*.

Table 3 shows the performance with different vocabularies. It is obvious that the 128K vocabulary outperforms the 256K vocabulary, 23.14 vs 21.65 spBLEU. However, if we finetune the pretrained *Trans_small* with the 256K vocabulary, 0.58 score improvement is achieved compared to the 128K *Trans_small*. In a word, 128K vocabulary is a better choice for training from scratch, while pretrained model offers us more gain.

Model	Ave. spBLEU
1st finetuned on parallel data	32.46
2nd finetuned on synthetic data	34.73

Table 5: Average spBLEU on the devtest set of Small Task 1 for *Trans_base* on different steps. These two models are iteratively trained. *Trans_base* is first finetuned on the parallel data, and then finetuned on the combination of the parallel data and the synthetic data generated by the previous step *Trans_base*.

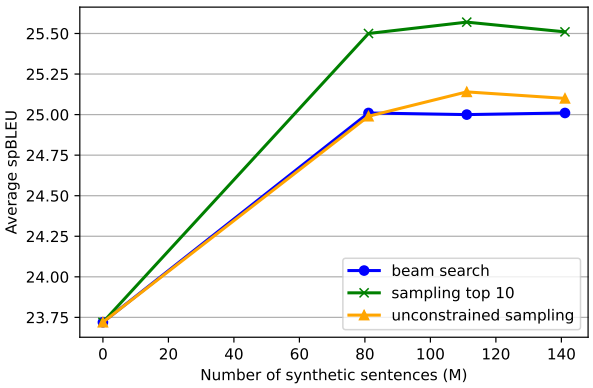


Figure 1: Average spBLEU on the devtest set of Small Task 2 for different back-translation methods with various amount of synthetic data. 80M synthetic data covers only 6M monolingual English data and all other monolingual data. We increase the amount of monolingual English data with a interval of 6M for the last two experiments.

4.2 Different Back-translation Methods

Similar to Edunov et al. (2018), we explore three types of back-translation methods, i.e. beam search with the beam size of five (Sennrich et al., 2016a), unconstrained sampling (Edunov et al., 2018) and sampling constrained to the most 10 likely words (Graves, 2013; Ott et al., 2018; Fan et al., 2018). Unconstrained sampling predicts the next word from the whole vocabulary without caring for the model distribution. Whereas constrained sampling predicts the next words within some candidates that have the highest prediction probabilities. Both constrained and unconstrained sampling can be considered as adding uncertainty to the greedy search.

Figure 1 shows the back-translation results on the devtest set of Small Task 2. We combine three different amount of synthetic data and parallel data to further train our *Trans_small*s after finetuned on parallel data. 80M synthetic sentences cover only 6M monolingual English data and all other monolingual data. In addition to the 80M synthetic sentences, we further increase the amount of monolingual English data to verify the model performance with respect to the amount of synthetic English data on the target side. The reason for this implementation is there are too many monolingual English sentences compared to other languages. We try to check whether it is necessary to use all monolingual English sentences.

As seen in Figure 1, little improvement is obtained with increasing the number of monolingual English sentences after 6M. Besides, in contrast to the results in Edunov et al. (2018) where the unconstrained sampling offers the best performance among these three methods, the constrained sampling method gives us the best score.

Beam search is the worst among these three methods. We hypothesize this is because beam search focuses only on the high probability words, while both constrained sampling and unconstrained sampling methods offer rich translations on the source side. With the diverse synthetic data generated from the sampling methods, model can be trained with more generalization.

In contrast to the bilingual translation (English-German) in Edunov et al. (2018) where unconstrained sampling outperforms constrained sampling, multilingual translation of Small Task 2 contains similar languages. We argue that unconstrained sampling might result in generating synthetic sentences with a mix of similar languages, which damages the quality of synthetic data, while constrained sampling gives us some restriction, to some extent avoiding the mix of different languages.

The reason for the slight effect of the synthetic English (on the source side) data after 6M might be that English is dissimilar to the other five South East Asian languages. Less similar knowledge could be transferred from this synthetic English (on the source side) data to other languages.

4.3 Final Submissions

Section 4.1 suggests us to employ a pretrained model with the 128K vocabulary. M2M_100 (Fan et al., 2021) offers multiple pretrained models with the 128K vocabularies [1]. Their sizes are 418M, 1.2B and 12B, respectively. Considering our limited GPU budget, we finetune the 1.2B model, i.e.

[1] https://github.com/pytorch/fairseq/tree/master/examples/m2m_100

Small Task	devtest	hidden dev	hidden devtest	hidden test
#1	34.73	35.12	35.39	34.96
#2	33.01	33.74	33.51	33.34

Table 6: Average spBLEU on different test sets for both small tasks. The hidden sets are invisible to the participants. The final ranking is based on the model performance on the hidden test set.

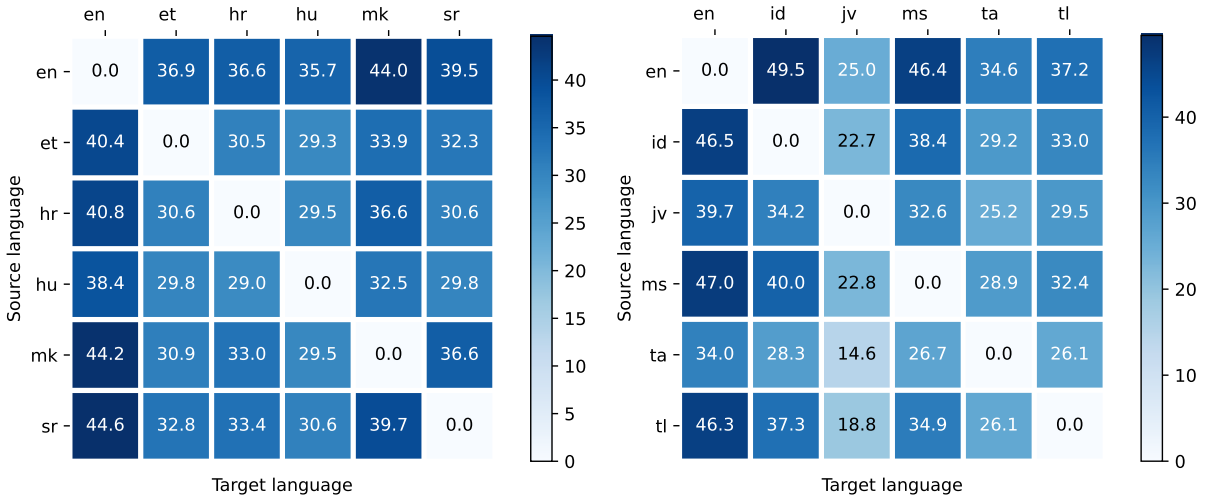


Figure 2: The spBLEU scores of different language pairs for both small tasks on the devtest set from our final submissions.

Trans_big, on parallel data of Small Task 2, obtaining 28.78 spBLEU on the devtest set. Whereas, training a *Trans_base* on the same data only provides 28.23 spBLEU. Even though *Trans_big* outperforms *Trans_base*, we only train it for generating high-quality synthetic data, since it is too large for the evaluation system.

Section 4.2 advises us to use the constrained sampling method on partial monolingual English data. With the constrained sampling method, we generate synthetic sentences with *Trans_big* that is first finetuned on the parallel data. Instead of using all monolingual English data, we synthesize en-id, en-jv, en-ms, en-ta and en-tl with all, 15M, 60M, 10M and 60M monolingual English sentences, respectively, a ratio of about $5:1$ between the number of parallel sentences and synthetic sentences if there are enough monolingual data.

Table 4 shows the results for iterative finetuning. Except for finetuing *Trans_base* on the combination of the parallel data and the synthetic data generated by *Trans_big*, we use the finetuned *Trans_base* to generate the synthetic data secondly and finetune it again. Finally, it offers us 33.01 spBLEU on the devtest set for Small Task 2.

Due to time and resource limit, we only conduct one trial on Small Task 1. We first finetune the pretrained *Trans_base* on parallel data. Then we use this *Trans_base* to generate synthetic data with only 20M monolingual English sentences and all other monolingual sentences. Table 5 shows the corresponding results. Different with Small Task 2, large amount of monolingual English data might be helpful for Small Task 1, since Central/East European languages are more similar to English than Asian languages. Finally, We leave this exploration to the future work.

Table 6 summarizes the results of our submissions on different evaluation sets for both small tasks. And Figure 2 lists the spBLEU scores for all language pairs of both small tasks on the devtest set. Finally, our submissions achieve the second place for both small tasks.

5 Conclusion

We demonstrate that a pretrained model with the smaller size of vocabularies is a better choice. Because of the memory and time limit of the evaluation system, we can only apply a 1.2B model with the smaller vocabularies to generate high-quality synthetic data. Besides, we have a different obser-

vation than previous research for bilingual back-translation: the constrained sampling method performs the best among all three back-translation methods, including the beam search and the unconstrained sampling. Finally, we also show that extensive monolingual English data offers a modest improvement. Combining these three findings, we iteratively train our models on partial high-quality synthetic data, achieving the second place for both small tasks.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019a. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019b. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, page 3874–3884.

Gyu Hyeon Choi, Jong Hun Shin, and Young Kil Kim. 2018. Improving a Multi-Source Neural Machine Translation Model with Corpus Extension for Low-Resource Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chenhui Chu and Raj Dabre. 2019. Multilingual multi-domain adaptation approaches for neural machine translation. *CoRR*, abs/1906.07978.

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.

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. 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*, pages 489–500.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2020. Training multilingual machine translation by alternately freezing language-specific encoders-decoders. *CoRR*, abs/2006.01594.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017a. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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. 2017b. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine

translation models without shared vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257, Florence, Italy. 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 association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Conference of the Association for Computational Linguistics (ACL)*.

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.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Maastricht University's Large-Scale Multilingual Machine Translation System for WMT 2021

Danni Liu, Jan Niehues
Department of Data Science and Knowledge Engineering, Maastricht University
{danni.liu,jan.niehues}@maastrichtuniversity.nl

Abstract

We present our development of the multilingual machine translation system for the large-scale multilingual machine translation task at WMT 2021. Starting form the provided baseline system, we investigated several techniques to improve the translation quality on the target subset of languages.

We were able to significantly improve the translation quality by adapting the system towards the target subset of languages and by generating synthetic data using the initial model. Techniques successfully applied in zero-shot multilingual machine translation (e.g. similarity regularizer) only had a minor effect on the final translation performance.

1 Introduction

This paper describes Maastricht University's participation in the large-scale multilingual machine translation task of WMT 2021. We participate in Small Track #2. In this track, the task is to build a translation system between English and 5 Southeast Asian languages. The evaluation is performed on all 30 possible translation directions between these languages. We are provided with parallel data extracted from Wikipedia and other sources for all language pairs, as well as a large-scale multilingual machine translation model pretrained on 124 languages (Goyal et al., 2021) including the languages in this task.

Starting from the provided baseline models, we investigate several directions in order to improve the performance on the 30 target language directions. As the first step, we focus on methods to adapt the model to these language directions. Specifically, we investigate different strategies to fine-tune the model on the proposed parallel training data.

Since the provided parallel data is extremely limited for several translation directions, we investigate the use of synthetic parallel data. We focus on using pivot languages in order to use well performing language direction to generate training data for worse performing language directions.

Finally, we investigate the usefulness of techniques to promote the similarity of representation between different languages. While these techniques were shown essential for models to perform zero-shot machine translation (Arivazhagan et al., 2019a; Pham et al., 2019; Liu et al., 2021), in our experiments the impact of these methods is only limited.

	jv	id	ms	tl	ta	en
su	44	243	137	440	108	904
jv		644	340	662	46	2,556
id			4,060	2,356	415	48,486
ms				1,174	297	12,023
tl					489	12,348
ta						1,864

Table 1: Number of sentences for each languages pair after preprocessing (in thousand sentences).

2 Data

We start by introducing the training data and preprocessing steps.

2.1 Languages

As required for the small tracks, we only use the data provided by the organizers. The covered languages are: Javanese (jv), Indonesian (id), Malay (ms), Tagalog (tl), Tamil (ta), English (en). Although Sundanese (su) is later excluded from the evaluations, we still include it in the training data because of its high relatedness to Javanese (jv), one of the lowest-resource languages in this track.

2.2 Preprocessing

After de-duplicating, we remove sentences with more than 50% punctuation marks or digits, and

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 425–430
November 10–11, 2021.

sentence pairs of length ratios beyond 1:5 in character count. We also apply frequency cleaning following M2M-100 (Fan et al., 2021). An overview of the training data amount after preprocessing is shown in Table 1.

3 Techniques

Our efforts to improve upon the provided baselines can be categorized into three directions: finetuning (subsection 3.1), utilizing synthetic data (subsection 3.2), encouraging similarities between languages (subsection 3.3).

3.1 Language Adaptation

While the provided baseline systems are trained on over 100 languages, our target track focuses on 6 specific languages. Therefore, we investigate different fine-tuning methods to adapt the model to the target languages. In this framework, we initialize from the pretrained baseline model and continue training on different subsets of the given training data. First (§3.1.1), we finetune on each of the language pairs. Then (§3.1.2), we apply finetuning on all the languages jointly. Finally (§3.1.3), in an attempt to preserve model performance on all other directions, we use adapter layers dedicated to the languages of interest.

3.1.1 Language-Specific Adaptation

First, we adapt the model to each translation direction individually, resulting in 30 different translation systems. Each of the models is trained on a single language pair from the provided translation directions.

While this approach achieves good performance, the main disadvantage is that we will have 30 individual systems. Since all of the individual models are fine-tuned from the same baseline model, we hypothesize that the resulting models are relatively similar. Therefore, we investigate the possibility of checkpoint-averaging on all the models adapted to individual language pairs, as successfully done in previous evaluations of multilingual translation (Pham et al., 2017).

3.1.2 Language-Independent Adaptation

Similar to the motivation for the checkpoint-averaging described above, in order to preserve one single model, we adapt the baseline system to all the target language pairs jointly by continue training on all the provided training data.

3.1.3 Adapter Layers

The approaches above update all parameters of the pretrained baseline model during the fine-tuning stage. As a result, the models would lose performance on translation directions other than those in the training data. To avoid this catastrophic forgetting, we take inspiration from the adapters (Bapna and Firat, 2019) and insert feedfoward layers after each encoder/decoder layer. When finetuning, we only train these parameters while keeping the rest of the model frozen. At test time, the model keeps the adapter layers for the languages seen in training. When handling languages unseen in training, the model drops the adapters and falls back to the pretrained baseline.

A main difference to the multilingual adapters (Bapna and Firat, 2019) is that our adapter layers are not language-pair-specific. Instead, they are shared among all the directions. A main reason for sharing the adapters is that, when scaling to more languages, a quadratic set of adapters would be needed.

Due to resource constraints, in this work we only train one set of adapter layers for the translation directions in Small Track #2. Nevertheless, we believe this approach could remain applicable when scaling to more languages like in Large Track. This could, for instance, be achieved by multiple sets of adapter layers dedicated to different language families.

3.2 Synthetic Data

Motivated by the strong improvements from synthetic data in multilingual speech translation evaluation (Anastasopoulos et al., 2021), we investigate the creation of synthetic parallel data for all language direction with limited available parallel data. Based on the corpus statistics (Table 1), we select all languages pairs with less than n parallel sentences as low-resource directions. In the initial experiments, we choose a threshold of $n = 500$K sentences pairs. Following successful initial experiments, we increase n to 2M sentences. For all the language with $k < n$ parallel sentences, we generated $n - k$ synthetic sentences. Consequently, the final system is trained on at least n sentences for each language pair.

While monolingual data was provided, as the initial translation system performed poorly on low-resource directions, we chose not to directly generate synthetic data from the monolingual data. In-

stead, we create synthetic data based on parallel data between the source and a pivot language. Synthetic target-side data is created by translating out of the pivot language. When selecting the pivot language, we choose the source-pivot pair with the highest BLEU scores.[1]

In this above-described scenario, as the source sentences are human-generated and the synthetic target sentences are automatically generated, we hypothesize that the mistakes are concentrated on the target side. This is normally addressed by using back-translation and generating the translation in the inverse directions. Since we aim to use the generated languages for both direction (source to target and target to source) for the sake of efficiency, we generate half of the sentences in the one direction and the other half in the inverse direction.

3.3 Encouraging Similar Representations

As shown in the statistics in Table 1, our training data is highly unbalanced. Extreme low-resource pairs such as ta↔jv could be considered as few-shot directions. We therefore explore several techniques shown useful for zero-shot conditions, and investigate their usefulness in this current scenario.

3.3.1 Similarity Regularization

First shown in (Arivazhagan et al., 2019a; Pham et al., 2019), an auxiliary loss promoting similarities between source and target languages facilitate zero-shot translation. Given source sentence X and target sentence Y, besides the translation loss, we minimize the following auxiliary loss:

$$\mathcal{L}_{similarity} = \lambda \cdot \texttt{dist}(X, Y), \qquad (1)$$

where $\texttt{dist}(\cdot)$ is the Euclidean distance between two meanpooled sentence embeddings, and λ is the weight for this auxiliary loss.

3.3.2 Residual Removal

The residual removal approach was shown helpful to zero-shot translation by reducing the positional information from source sentences (Liu et al., 2021). Specifically, the residual connections of a middle encoder layer is removed to relax the strong positional correspondence between input tokens and encoder outputs.

[1] In the later experiments, the best-performing source-pivot direction is always source-English.

4 Experimental Setup

4.1 Training Details

The provided M2M-100 models (Fan et al., 2021; Goyal et al., 2021) cover over 100 languages and have a vocabulary size of 256K. To accelerate training and reduce GPU memory usage, we trim away the word embedding of those tokens that do not occur in our training data. After vocabulary trimming, our vocabulary size is 165K. An exception where we do not trim the vocabulary is when training with adapters, since the goal is to preserve performance on all languages.

To counteract the data imbalance among the language pairs, following previous works (Arivazhagan et al., 2019b; Tang et al., 2020), we use a sampling temperature of 5.0 which upsamples the low-resource pairs.

For the model with similarity regularizer, we use weight of 0.1 on the auxiliary loss. For the model with residual removal, the residual layer is skipped after the third encoder layer. For the adapters, we use a bottleneck dimension of 256.

4.2 Decoding and Evaluation

When decoding we use a beam size of 5, and limit the maximum output length to 1.3 * source length + 5. We report translation performance on the FloRes-101 (Goyal et al., 2021) devtest set. The BLEU reported scores are the spBLEU (Goyal et al., 2021) variant based on sentencepiece (Kudo and Richardson, 2018) tokenization. The systems are also submitted to Dynabench (Kiela et al., 2021)[2] for evaluation on a blind test set.

5 Results

In this section, we report the results of the three directions we explored: finetuning on the focus languages (subsection 5.1), utilizing synthetic data (subsection 5.2), and encouraging similarities between languages (subsection 5.3).

5.1 Language Adaptation

The results of adapting the model towards different language pairs are shown in Table 2.

5.1.1 Language-Dependent and -Independent Adaptation

We start with the small baseline model for faster experiment iterations, with results summarized in

[2] https://dynabench.org/flores

System	BLEU
baseline small	11.5
+ lang. dep. fine-tune	20.6
+ averaging	5.7
+ lang. indep. fine-tune	19.6
baseline big	15.4
+ lang. indep. fine-tune	27.4
+ shared adapter layers (rest frozen)	23.0

Table 2: Results of different fine-tuning approaches from the baseline models. Language-dependent fine-tuning achieves the strongest performance, but creates individual models for each translation direction. Averaging the models from the individual directions performs poorly. Fine-tuning on all directions falls slightly behind language-specific fine-tuning but preserves one single model.

the upper section of Table 2. When adapting to each language pair individually (lang. dep. fine-tune), we see large gains with average BLEU score increasing from 11.5 to 20.6. In contrast to previous work (Pham et al., 2017), we are not able to preserve this gain by averaging all the individual models into one single models. Instead, averaging the models results in a low average BLEU score of 5.7. This suggests the adapted individual models are relatively dissimilar and cannot be simply averaged.

Nevertheless, by fine-tuning on all 30 language directions together (lang. indep. fine-tune), we achieve a comparable gain in performance, results in a BLEU score of 19.6. Since this is achieved by a single model instead of 30 individual models, we continue with jointly training on all directions in the upcoming experiments on the big baseline model. Similar to findings on the small model, by fine-tuning on all the languages, we were able to improve the average BLEU score of the big baseline model from 15.4 to 27.4.

5.1.2 Adapters

As shown in the lower section of Table 2, by inserting adapters into the large baseline model and only training these modules, we achieve 23.0 BLEU on average. While the gain is less compared to full parameter tuning, the model preserves performance on the remaining tens of thousands directions.

As motivated previously (§3.1.3), the adapter layers are shared across the language directions rather than language-pair-specific. This could explain the performance gap to full parameter tuning.

System	jv-ta	ta-jv
baseline big	3.8	3.1
+ parallel data	8.5	7.9
+ syn. jv-ta data	10.0	8.2
+ syn. ta-jv data	15.0	10.7
+ both syn. data	15.9	9.7
+ both syn. data big	16.0	11.7

Table 3: Impact of synthetic data on jv↔ta, the lowest-resource language pair in this task.

5.2 Synthetic data

In the first set of experiments, we evaluate the influence of synthetic data only on the translations between Javanese (jv) and Tamil (ta), since this was the language pair with the least data (44K sentences). The synthetic data was always produced by the system fine-tuned on all the target language directions. The results are summarized Table 3. First, although the available parallel data is limited, we see a clear improvement of the baseline model when trained on the provided training data.

Adding the synthetic data (225K sentences for jv-ta and ta-jv each) does improve the performance compared to only using the parallel data. For both directions, the data generated from ta-jv was performing better than the other data. Since the combination of both directions performed the best for the jv-ta direction and reasonable good for the other direction and it is not clear how we should select the direction without perfoming test for each langauge pair, we continued the experiments by always using synthetic data generated by both directions.

By increasing the amount of synthetic data, so that the model is not trained on around 500k sentences by 2M sentence, we see additional gains to the best performance of 16.0 and 11.7 BLEU points for both directions. This is an improvement nearly by a factor of 3 compared to the baseline system.

Given the best system so far with 27.4 average BLEU, we continue fine-tuning with the additional synthetic data. This leads to an improvement to 27.9 BLEU on average. This improvement is significantly lower than expected, considering the general positive role of utilizing synthetic data. This has two potential reasons. First, as all other language directions have more data, the gains from the additional data could be reduced. Furthermore, the initial model is fine-tuned on the parallel data

Directions	# Sent.	Δ BLEU
jv↔ta	46K	+0.6
ms↔ta	297K	+0.1
jv↔ms	340K	+0.1
Overall	89M	+0.0

Table 4: The average change in BLEU after fine-tuning with residual removal. There is no gain in overall average BLEU, and limited gain in the top 3 lowest-resource directions.

of all the language pairs and therefore performing better.

5.3 Encouraging Similar Representations

Next we report the results of the approaches that promote language similarity as motivated in subsection 3.3.

5.3.1 Similarity Regularizer

Based on the best system trained with synthetic data (with 27.9 BLEU on average), we continue fine-tuning with the similarity regularizer described in §3.3.1. While we observe consistent increase in the similarity scores on the dev set, fine-tuning with the similarity regularizer alone does not improve the system further, achieving 27.7 BLEU on average. Nevertheless, we see gains when combining the similarity regularizer and the adapters described in §3.1.3. As adding the adapter layers expands the capacity of the existing model, we hypothesize the similarity regularizer could help combat overfitting. With this combination, we achieve an average of 28.1 BLEU.

5.3.2 Residual Removal

Based on the baseline big + fine-tune model (with 27.4 BLEU on average), we fine-tune once again using the residual-removal architecture described in §3.3.2. In Table 4, we summarize the average change in BLEU after this additional fine-tuning step. While there was no improvement in the overall average BLEU score, we observe some gain in the lowest-resource direction of jv↔ta which has 46K parallel data. However, the gain falls largely for the second and third lowest-resource directions.

5.4 Final System

The final system submitted to the evaluation is presented in Table 5. In a first step, we fine-tuned on the provided parallel data. Using this model, we created additional synthetic data. Fine-tuning the previous model on the parallel data and the synthetic data gave an additional improvement of 0.5 BLEU.

System	BLEU
baseline big	15.4
+ fine-tune	27.4
+ synthetic data	27.9
+ sim. regularizer + adapter	28.1

Table 5: Average BLEU scores on FLoRes-101 devtest set on 30 directions of the final system.

Finally, on top of the previous improvements, our best system uses the additional similarity regularization and adapters during training and further improves the average BLEU by 0.2 points to 28.1. The submitted system achieves 28.6 BLEU on average on the blind test set[3].

6 Conclusion

This paper summarizes our participation in the WMT 2021 large-scale multilingual translation task. We focus on Small Track #2 for English and 5 Southeast Asian languages. Building upon the provided baseline models, we achieved the largest gain from fine-tuning on the parallel data of all directions in this task. By further utilizing synthetic data and a combination of similarity regularization and adapters, we were able to further improve the system.

References

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *arXiv preprint arXiv:1903.07091*.

[3] https://dynabench.org/models/445

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019b. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021. Improving zero-shot translation by disentangling positional information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1259–1273, Online. Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

Ngoc-Quan Pham, Matthias Sperber, Elizabeth Salesky, Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2017. Kit's multilingual neural machine translation systems for iwslt 2017. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Data Processing Matters: SRPH-Konvergen AI's Machine Translation System for WMT'21

Lintang Sutawika*
Konvergen AI
Jakarta, Indonesia
lintang@konvergen.ai

Jan Christian Blaise Cruz*
Samsung Research Philippines
Manila, Philippines
jcb.cruz@samsung.com

Abstract

In this paper, we describe the submission of the joint Samsung Research Philippines-Konvergen AI team for the WMT'21 Large Scale Multilingual Translation Task - Small Track 2. We submit a standard Seq2Seq Transformer model to the shared task without any training or architecture tricks, relying mainly on the strength of our data preprocessing techniques to boost performance. Our final submission model scored 22.92 average BLEU on the FLORES-101 devtest set, and scored 22.97 average BLEU on the contest's hidden test set, ranking us sixth overall. Despite using only a standard Transformer, our model ranked first in Indonesian → Javanese, showing that data preprocessing matters equally, if not more, than cutting edge model architectures and training techniques.

1 Introduction

This paper describes the machine translation system submitted by the joint team of Samsung Research Philippines and Konvergen AI for the WMT'21 Large Scale Multilingual Translation Task. Our team participated in **Small Track #2**, where the task is to produce a multilingual machine translation system for five Southeast-Asian languages: Javanese, Indonesian, Malay, Tagalog, and Tamil[1], plus English, in all 30 directions.

We will first describe the filtering heuristics that we used to preprocess the data, and then outline the steps we took to train and evaluate our models. Specific hyperparameters, preprocessing decisions, and other training parameters will be listed in their corresponding sections. Finally, we report our results on the FLORES-101 (Goyal et al., 2021) devtest set, as well as on the competition's hidden test set.

*Equal contribution. Order determined via coinflip.

[1]Tamil is considered an official language in Singapore, a Southeast Asian country

2 Parallel Text Preprocessing Heuristics

The contest dataset comprises of various bitext sources, including: bible-uedin (Christodouloupoulos and Steedman, 2015), CCAligned (El-Kishky et al., 2020), ELRC 2922[2], MultiCCAligned (El-Kishky et al., 2020), ParaCrawl [3], TED2020 (Reimers and Gurevych, 2020), WikiMatrix (Schwenk et al., 2019), tico-19, Ubuntu, OpenSubtitles, QED, Tanzil, Tatoeba, GlobalVoices, GNOME, KDE4, and WikiMedia (Tiedemann, 2012).

We preprocess the datasets before training in order to minimize spurious relations that originate from incorrect text pairs. Our preprocessing removes samples based on a few heuristics that we developed based on our observation on the datasets. Each bitext file is applied a different set of preprocessing based on observation. For example we filter by number content for datasets such as CCAAligned while TED2020 is not applied that same filter.

In this section, we will cover the decisions made during preprocessing. We observe a score increase of 1.91 BLEU on our submission model when the preprocessing is applied. We report the total number of lines filtered from the bitext for all language pairs on Table 1.

2.1 Filter by Duplicate

Duplication is present throughout the dataset. Table 2 outlines samples of duplication based on three distinct types:

- **Duplicates within the same language** Within a subset file of a designated language, multiple lines have the same string while the its counterpart may feature different translations.

[2]https://elrc-share.eu/

[3]https://www.paracrawl.eu/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 431–438
November 10–11, 2021.

ISO	Language Pair	Before Preprocessing	After Preprocessing	Reduction
en-id	English - Indonesian	54,075,891	27,186,074	49.73%
en-ms	English - Malaysian	13,437,727	7,674,956	42.89%
en-tl	English - Tagalog	13,612,403	5,302,768	61.04%
en-jv	English - Javanese	3,044,920	388,766	87.23%
en-ta	English - Tamil	2,115,925	1,420,827	32.85%
id-ms	Indonesian - Malaysian	4,857,321	3,371,777	30.58%
id-tl	Indonesian - Tagalog	2,743,305	1,823,140	33.54%
id-jv	Indonesian - Javanese	780,119	432,734	44.53%
id-ta	Indonesian - Tamil	500,898	393,336	21.47%
ms-tl	Malaysian - Tagalog	1,358,486	985,493	27.46%
ms-jv	Malaysian - Javanese	434,710	250,070	42.47%
ms-ta	Malaysian - Tamil	372,623	351,416	5.69%
tl-jv	Tagalog - Javanese	817,146	544,233	33.40%
tl-ta	Tagalog - Tamil	563,337	482,618	14.33%
jv-ta	Javanese - Tamil	65,997	48,806	26.05%

Table 1: Number of parallel text lines per language pair before and after applying preprocessing

- **Partial duplication** The whole string of text in one language is present in its counterpart translation.
- **Duplication among parallel text** Both source and target text line feature exactly the same string. While this may be correct for named entities, most of these duplication are short and can be non-informative.

2.2 Filtering by Language and Letters

In algorithmically-aligned datasets such as CCAligned, some training examples are not in the list of contest languages. We find full text lines that are in Azerbaijani, Turkish, Arabic, and Japanese. To identify these languages, we use langdetect[4]. This filter works for sentences that are fully foreign. It is also the case that foreign letters that may refer to named-entity can be found in the dataset. We consider this to be allowable so long as the the foreign character string is present in both source and target text line. To filter this, we use AlphabetDetector[5] and check if detected foreign letters are present in both text line.

2.3 Filter by Specific Keywords and Symbols

There are a number of cases where the translations are generally correct but also feature extra keywords that have no relation to the parallel text. These keywords are generally in English and are consistently present in a number of bitext datasets such as KDE4, GNOME, and Ubuntu.

Bitexts such as OpenSubtitles feature secondary information that relates to a particular scene (for example "*(loud music playing)*"). These secondary information may be in parentheses to denote an action being done or to signify a song being played. These secondary information are not always available for each language. We opt to remove all lines that have these specific symbols.

2.4 Filtering Number Content

We apply a filter to remove incorrect text lines in the bitext by checking if both source and target text lines feature the same numeric values such as date and quantities. Table 4 shows that filtering by number can remove text lines that do not relate to one another as numeric values tend to translate the same. Due to the limited time allotted for the shared task, we opt to remove entirely parallel sentences that do not have matching numbers. We filter this by using regular expressions.

2.5 Filtering by Length

Text lines with very long lengths are generally not informative, we find most of these text lines consists of a list of names that would normally be found in a bibliography. We set an arbitrary max length of 500 characters for both source and target sentences.

[4]https://pypi.org/project/langdetect/
[5]https://pypi.org/project/alphabet-detector/

Duplicates within the same file	
GNOME.en-tl.en	**GNOME.en-tl.tl**
Error reading from file: %s	Error sa pagbasa ng talaksang '%s': %s
Error seeking in file: %s	Error sa pagbasa ng talaksang '%s': %s
Error closing file: %s	Error sa pagbasa ng talaksang '%s': %s
Partial duplication	
WikiMatrix.en-jv.en	**WikiMatrix.en-jv.jv**
CJ E&M Corporation.	Drama iki diprodhuksi déning CJ E&M Corporation.
New Orleans, Louisiana.	Lair ing New Orleans, Louisiana.
Edward Thomas Hardy.	Jeneng dawané ya iku Edward Thomas Hardy.
Duplication among parallel text	
OpenSubtitles.en-ta.en	**OpenSubtitles.en-ta.ta**
Those who are invited will find the way.	Those who are invited will find the way.
Gazelle, whose face the full moon forms:	Gazelle, whose face the full moon forms:
Time has warned us never to approach her.	Time has warned us never to approach her.

Table 2: Examples of duplication based on three types

KDE4.en-id.en	**KDE4.en-id.id**
Task Scheduler	Penjadwal TugasComment
Configure and schedule tasks	Atur dan jadwal tugasName

Table 3: Example of translations that also have an extra keyword. Underlined text are keywords that are misplaced in correct translations.

	MultiCCAligned.id-tl.id	**MultiCCAligned.id-tl.tl**
Removed	Di. 13:00 - 17:30 Di 24 nov. 10h – 18h	Mo. 13:00 - 18:00 Sa 23 nov. 10h – 18h
Kept	(Terakhir diperbarui saat: 24/03/2020) Harga / $: 1,2835	(Huling nai-update Sa: 24/03/2020) presyo / $: 1.2835

Table 4: Incorrect translations can be easily identified by checking whether numeric values in both strings match. In the first example, the sentence pair was removed due to differing date and time. In the second example, the sentence pair was kept as we do not check punctuation for numerical values.

3 Experiments

3.1 Model Architecture

For our submission, we wish to measure how much performance can be boosted by heuristics-based data preprocessing alone. Given that we anticipate most, if not all, submissions to the shared task will be transformer-based models, we opt to use the standard "vanilla" Sequence-to-Sequence Transformer (Vaswani et al., 2017) model with little-to-no changes. This lets us more clearly compare the performance boost of our filtering heuristics against the boost provided by a number of architecture augmentations and training tricks that other submissions might have.

In addition to using a standard Transformer model, we only train the model directly on our filtered bitext and do not make use of Backtranslation (Sennrich et al., 2015a) for data augmentation. We also start from-scratch with models initialized using Glorot Uniform (Glorot and Bengio, 2010), opting not to use massively-pretrained translation models such as M2M-100 (Fan et al., 2021) as our starting checkpoint.

Following Vaswani et al. (2017), we produce two models: a base model and a large model. For the sake of simplicity, for the rest of the paper, we will refer to our models trained with our filtered data as $\textbf{Base}_{\textbf{Heuristics}}$ and $\textbf{Large}_{\textbf{Heuristics}}$.

The hyperparameters used for our models are presented in Table 5.

	Base	Large
Vocab Size	37,000	37,000
Encoder Layers	6	6
Decoder Layers	6	6
Attention Heads	8	16
Embedding Dim.	512	1024
Feedforward Dim.	2048	4096
Dropout	0.1	0.3
Attention Dropout	0.1	0.3
Pos. Embeddings	Sinusoid	Sinusoid
Parameters	63M	214M

Table 5: Model hyperparameter choices for the base and large Transformer variants.

3.2 Data Preformatting and Tokenization

Our models employ one single shared vocabulary for all languages and directions. We train our tokenizer using the SentencePiece[6] library, limiting our vocabulary to 37,000 BPE (Sennrich et al., 2015b) tokens, and training with a character coverage of 0.995.

Before training the tokenizer, we first preformat the dataset into the format to be used for training later on. We append the source and target language's ISO-639-1 code enclosed in square brackets at the beginning of each sentence. For example:

`[en] [tl]` Today is a sunny day.

is the preformatted version of "Today is a sunny day." when translating from English to Tagalog.

This preformatting is only done for the source sentences in the training dataset, while the target sentences are untouched.

For the purpose of training the tokenizer, the six language tokens (`[en]`, `[id]`, `[jv]`, `[ms]`, `[ta]`, and `[tl]`) are treated as special tokens to ensure that they will not be segmented later on.

3.3 Training Setup

We then compile our filtered, preformatted bitext and train our base and large models. During training, we limit all source and target sentences to a maximum sequence length of 150 subword tokens. All sentences that are much longer are truncated.

Our models are trained using the Adam (Kingma and Ba, 2014) optimizer. Following Vaswani et al. (2017), we also use the "Noam" learning rate scheduler, linearly increasing the learning rate from 0 for the first 8000 steps, then decaying afterward. We also set Adam's $\beta_2 = 0.998$ and use a label smoothing factor of 0.1.

For batching, we accumulate tokens until we reach a maximum size of approximately 32,000 tokens per batch, an increase over the 25,000 tokens used in Vaswani et al. (2017). We then train the base model and the large model for 100,000 steps and 300,000 steps, respectively. All our models are trained on 8 NVIDIA Tesla P100 GPUs in parallel using the OpenNMT-py (Klein et al., 2017) toolkit.

3.4 Translation

To generate translations using the model, we use Beam Search with beam size 5 and apply an average length penalty of 0.6. During generation, we limit all outputs to a maximum sequence length of 100, preemptively terminating generation if it begins to exceed this maximum length. We do not use sampling during translation, nor increase the temperature parameter as this induces randomness (Lopez et al., 2020).

We test our experimental models on the FLORES-101 devtest set. We report our BLEU scores using the SPM-BLEU variant of SacreBLEU[7] (Post, 2018).

4 Results

After training our models and producing sample translations from the FLORES-101 devtest set, we compare the results of our two models with a number of baselines:

- Transformers with No Heuristics – These models are essentially identical with our Transformer models in terms of architecture, hyperparameters, and training setups, except the bitext they are training on are the raw training corpus given in the competition (i.e. the filtering heuristics were not applied on them). We train these models as an ablation experiment to be able to identify how much of the final performance is attributable to the filtering heuristics.

- M2M-100 615M – This is the baseline given for the WMT'21 Large-scale Multilingual Translation Task Small Track 2 competition. This M2M-100 (Fan et al., 2021) model was

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

[7] BLEU+case.mixed+numrefs.1+smooth.exp+tok.spm +version.1.5.0

trained on CCMatrix and CCaligned with no further finetuning on the contest dataset.

- DeltaLM+ZCode – This is the best performing model for the Small Track 2. The model is a finetuned version of the DeltaLM (Ma et al., 2021) encoder-decoder pretrained model.

All analyses and results within this section are based on the *public devtest set* and not the contest's hidden test set, unless specified. A summary of the BLEU scores for all models and baselines are available on Table 6.

4.1 Transformer + Heuristics vs. Baselines

We report the results of our $\text{Base}_{\text{Heuristics}}$ and $\text{Large}_{\text{Heuristics}}$ models against the M2M-100 615M model baseline as well as the best performing model for the shared task.

$\text{Base}_{\text{Heuristics}}$ scored an average BLEU of 20.78 on all 30 directions. On the other hand, $\text{Large}_{\text{Heuristics}}$ scored 22.92 average BLEU on all 30 directions, which is 2.14 BLEU points higher than the base model. Both models outperformed the M2M-100 615M baseline, with the base model giving a 5.32 BLEU improvement, and the large model giving a 7.46 BLEU improvement.

It is worth noting that, while the $\text{Base}_{\text{Heuristics}}$ outperforms the baseline on average, it fails to outperform it on four specific translation directions: en↔id and en↔ms. Note that it is these two language pairs that have the most number of training sentences in the training corpus.

The language pairs that benefit significantly from training on the contest dataset are language pairs that are of less volume than en↔id and en↔ms. This is likely due to these pairs being less-sampled in M2M100's training dataset, and thus were not as learned by the model compared to pairs with a higher volume of training data.

The same observations can be found when comparing the performance of $\text{Large}_{\text{Heuristics}}$ against the baseline model. $\text{Large}_{\text{Heuristics}}$ only marginally outperformed the baseline in one direction (id→en, +0.07 BLEU), and marginally underperformed against the baseline in one direction (ms→en, -0.47 BLEU). This higher performance for M2M-100 is likely due to the training method used in the model in addition to the size of the training corpora used. While M2M-100 is advantageous in these translation directions, the difference is only marginal, most likely owing to $\text{Large}_{\text{Heuristics}}$'s size which gives it higher capacity.

Both our transformer models and the baseline model are significantly outperformed by the DeltaLM+ZCode model, which is the best performing model in the competition. The best model outperforms our best model ($\text{Large}_{\text{Heuristics}}$) by a significant 11.02 average BLEU, and the baseline model by 18.48 average BLEU.

While DeltaLM+ZCode outperforms our model in terms of average performance, it is worth noting that our model – a standard Transformer without any augmentations and training tricks – managed to outperform DeltaLM+ZCode in one translation direction: id→jv.

$\text{Large}_{\text{Heuristics}}$ scored 23.91 BLEU while DeltaLM+ZCode scored 23.35 BLEU. While the difference is marginal (+0.56 BLEU), our model still outperforms the best model in this direction, which we attribute to the quality of our data preprocessing and filtering heuristics.

4.2 Heuristics vs. No Heuristics

To quantify how much our filtering heuristics contributed to the final performance of our models, we trained two additional models: both identical to our base and large transformer variants, except the training corpus used was not processed using our filtering heuristics. For these ablation experiments, we use the same BPE tokenizer that is used for our main transformer models (trained on the filtered data). This is to ensure full model equivalency. To prevent confusion, we will refer to these ablation models simply as **Base** and **Large** to differentiate them from our contest models $\textbf{Base}_{\textbf{Heuristics}}$ and $\textbf{Large}_{\textbf{Heuristics}}$.

On average, both sizes of models performed worse when trained without the filtering heuristics. Base scored 19.28 average BLEU on the devtest set, 1.5 points lower than $\text{Base}_{\text{Heuristics}}$. On the other hand, Large scored average 21.01 BLEU, which is 1.91 points lower than $\text{Large}_{\text{Heuristics}}$.

It is interesting, however, that Base outperformed $\text{Base}_{\text{Heuristics}}$ in two translation directions: en→ms and ms→id. This may indicate that the filtering heuristics work better for a certain subset of languages. We look towards exploring how filtering methods such as ours affect multilingual translation datasets in terms of balance and informativeness in the future.

On the other hand, Large performed worse than $\text{Large}_{\text{Heuristics}}$ in all 30 directions. This may be due to the increase in total trainable parameters, as

	Base$_{\text{Heuristics}}$	**Large$_{\text{Heuristics}}$**	**Base**	**Large**	**M2M100 Baseline**	**DeltaLM +ZCode**
en→id	35.94	39.29	35.12	36.51	36.34	50.90
id→en	31.20	33.40	29.22	30.93	33.33	47.35
en→jv	21.53	23.57	16.95	20.98	15.06	27.70
jv→en	22.09	24.61	18.85	21.26	21.38	39.44
en→ms	31.36	36.93	36.63	38.60	32.63	46.77
ms→en	31.92	33.16	30.31	32.97	33.63	47.86
en→ta	9.15	10.64	8.78	9.68	4.24	35.48
ta→en	17.00	19.55	15.83	18.47	7.52	35.29
en→tl	26.91	33.23	27.87	27.56	9.95	40.52
tl→en	31.22	33.65	26.51	29.61	26.59	48.55
id→jv	23.18	23.91	21.41	22.30	15.86	23.35
jv→id	25.45	27.10	24.15	25.15	23.21	34.64
id→ms	30.58	33.94	28.38	33.01	29.32	38.30
ms→id	30.94	33.68	31.29	32.54	31.44	40.36
id→ta	7.04	7.88	6.78	7.09	1.44	29.61
ta→id	13.74	16.46	13.35	14.87	4.99	28.56
id→tl	23.32	25.27	22.30	23.23	9.32	33.56
tl→id	25.31	27.76	23.40	25.03	20.76	38.70
jv→ms	23.36	25.08	19.92	23.63	19.57	33.14
ms→jv	21.08	21.29	12.33	20.97	14.22	23.91
jv→ta	4.70	4.97	3.85	4.62	3.52	24.19
ta→jv	9.25	11.13	7.54	9.22	2.51	18.35
jv→tl	17.43	19.61	15.79	17.31	11.96	28.50
tl→jv	16.96	18.82	14.56	17.00	12.31	23.17
ms→ta	7.01	7.87	6.65	7.23	2.38	28.83
ta→ms	15.09	16.64	14.54	16.44	4.70	26.83
ms→tl	23.30	24.97	22.17	23.01	11.04	32.81
tl→ms	25.86	27.10	23.19	25.85	18.16	36.15
ta→tl	15.26	18.43	14.98	16.05	3.15	26.64
tl→ta	6.27	7.65	5.89	6.60	3.10	28.80
Average	20.78	22.92	19.28	21.01	15.46	33.94

Table 6: Summary of BLEU scores on the FLORES-101 devtest set. The first two columns show the performance of our Transformer models trained with the data filtering heuristics. The next two columns show the same Transformer models, but trained on an unpreprocessed version of the training dataset. We also show the scores of the M2M-100 615M baseline model, as well as the best performing model (DeltaLM+ZCode) for the Small Track 2. Large$_{\text{Heuristics}}$ (column 2) is our final submission model for the contest.

larger models need more data with higher quality to be effectively trained.

4.3 The Case of Tamil

We observe that our models, including the other models on the shared task leaderboard, struggled with Tamil. X↔ta translation is on average much worse in terms of BLEU score compared to the other translation directions that do not involve it.

We hypothesize that this is due to two things.

First, Tamil is the most underrepresented language in the shared task dataset, with X↔ta having the least amount of parallel text for every language X in the training set. This causes the model, to a certain extent, to underfit on directions that translate to or from Tamil.

Second, Tamil is the only language in the shared task dataset that does not use the latin alphabet. Combined with the fact that it is the most underrepresented language in the dataset, there is a possibility that the model may have treated Tamil as noise during training. The observation that X→ta performs worse on average compared to its inverse direction ta→X lends more credence to this hypothesis. The model is not trained well to represent sentences in Tamil, and thus, struggles when generating Tamil translations.

Part of our planned future work includes identifying methods to improve translation in multilingual datasets where the alphabets used may be more than one. This is to improve translation to non-latin alphabet languages in future methods.

4.4 Hidden Test Set Performance

We also report the performance of our models on the shared task's hidden test set. We once more compare our results against the baseline M2M-100 model as well as the best performing DeltaLM+ZCode model.

Our final submission for the shared task was our $\text{Large}_{\text{Heuristics}}$ model, which performed with an average BLEU of 22.97 on the shared task's hidden test set. This is a marginal difference from it's devtest set score (+0.05 average BLEU).

$\text{Large}_{\text{Heuristics}}$, unsurprisingly, still outperformed $\text{Base}_{\text{Heuristics}}$ (20.73 average BLEU, +2.24 improvement) and the baseline M2M-100 model (14.02 average BLEU, +8.95 improvement) in the hidden test set. The shared task's best performing model, DeltaLM+ZCode, still outperforms all other models in the hidden test set, scoring 33.89 average BLEU, a 10.92 improvement over our best model.

	Public Test	**Hidden Test**	**Rank**
M2M-100 615M	15.46	14.02	8
DeltaLM+ZCode	33.94	33.89	1
$\text{Base}_{\text{Heuristics}}$	20.78	20.73	-
$\text{Large}_{\text{Heuristics}}$	22.92	22.97	6

Table 7: Average BLEU scores on the contest's hidden test set. The $\text{Base}_{\text{Heuristics}}$ model is unranked as it was not submitted as our final model.

On the hidden test set, $\text{Large}_{\text{Heuristics}}$ still ranked first in the id→jv translation direction, scoring 24.05 BLEU. This outperforms DeltaLM+ZCode's 23.79 BLEU (+0.26) and M2M-100's 15.33 BLEU (+8.72).

A summary of our model's performance on the hidden test set, as well as the baseline and best performing model, can be found on Table 7

5 Conclusion

In this paper, we described the translation systems submitted by the joint Samsung Research Philippines-Konvergen AI team for the WMT'21 Large Scale Multilingual Translation Small Track 2 shared task. We outline the filtering heuristics that we took to preprocess our data. We then train two models with a bitext preprocessed using our filtering heuristics, with our best model reaching an average BLEU score of 22.92 on the devtest set, and outperforming the baseline model by 7.46 BLEU points. In addition, we rank sixth in the contest leaderboard overall, scoring 22.97 BLEU on the hidden test set.

We also reached first place for the id→jv translation direction, beating all other more complex models, despite only using a standard transformer without any special augmentations and training tricks. This provides empirical evidence that data quality and preprocessing decisions weigh just as much, if not even more, than cutting edge model architectures and training techniques do.

References

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document

pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

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. JMLR Workshop and Conference Proceedings.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

Diederik P 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.

Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. 2020. Simplifying paragraph-level question generation via transformer language models. *arXiv preprint arXiv:2005.01107*.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *arXiv preprint arXiv:2106.13736*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

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*.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

TenTrans Large-Scale Multilingual Machine Translation System for WMT21

Wanying Xie[1,2] **Bojie Hu**[1] **Han Yang**[1] **Dong Yu**[2] **Qi Ju**[1*]
[1] TencentMT Oteam, China
[2] Beijing Language and Culture University, China
xiewanying07@gmail.com, yudong@blcu.edu.cn
{bojiehu, sharryyang, damonju}@tencent.com

Abstract

This paper describes TenTrans large-scale multilingual machine translation system for WMT 2021. We participate in the Small Track 2 in five South East Asian languages, thirty directions: Javanese, Indonesian, Malay, Tagalog, Tamil, English. We mainly utilized forward/back-translation, in-domain data selection, knowledge distillation, and gradual fine-tuning from the pre-trained model FLORES-101. We find that forward/back-translation significantly improves the translation results, data selection and gradual fine-tuning are particularly effective during adapting domain, while knowledge distillation brings slight performance improvement. Also, model averaging is used to further improve the translation performance based on these systems. Our final system achieves an average BLEU score of 28.89 across thirty directions on the test set.

1 Introduction

We participate in the WMT 2021 large-scale multilingual machine translation task small track 2 in 6 languages: English, Indonesian, Javanese, Malay, Tamil, Tagalog (briefly, En, Id, Jv, Ms, Ta, Tl). Any two of these languages translated into each other produces a total of 30 directions, including English↔Indonesian, English↔Javanese, English↔Malay, English↔Tamil, English↔ Tagalog, Indonesian↔Javanese, Indonesian↔Malay, Indonesian↔Tamil, Indonesian↔Tagalog, Javanese ↔Malay, Javanese↔Tamil, Javanese↔ Tagalog, Malay↔Tamil, Malay↔Tagalog and Tamil↔Tagalog. To meet the requirements for data restrictions, our systems are all built with constrained data sets. For all systems, we adopt a universal encoder-decoder architecture that shares parameters across all languages (Johnson et al., 2017).

Our systems are based on several techniques and approaches. We experiment with base and deeper Transformer (Vaswani et al., 2017) architectures to get reliable baselines, fine-tune the pre-training model FLORES-101 (Goyal et al., 2021) to further improve the baseline system. Moreover, we generate pseudo bilingual sentences from the large-scale monolingual data, apply sequence level knowledge distillation (Kim and Rush, 2016) on partial language pairs, and try a more effectively fine-tuning strategy to domain adaptation (Gu et al., 2021). Particularly in the language pairs with inferior translations, we specifically improve their performance. All of these technologies have improved our systems, particularly data selection and gradual fine-tuning. We carefully rethought this strategy and found the main gain may come from in-domain knowledge adaptation.

This paper was structured as follows: Section 2 describes the data set. Then, we present a detailed overview of our systems in Section 3. The experiment settings and main results are shown in Section 4. Finally, we conclude our work in Section 5.

2 Data Prepration

We use FLORES-101 SentencePiece (SPM) [1] tokenizer model with 256K tokens to tokenize bitext and monolingual sentences [2]. Since it is important to clean data strictly (Wang et al., 2018), we follow m2m-100 data preprocessing procedures [3] to filter bitext data. The rules are as follows:

- Remove sentences with more than 50% punctuation.

*Corresponding author: Qi Ju.

Our code, data, and model can be obtained at https://github.com/TenTrans/TenTrans

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

[2] https://dl.fbaipublicfiles.com/flores101/pretrained_models/flores101_mm100_615M.tar.gz

[3] https://github.com/pytorch/fairseq/tree/master/examples/m2m_100

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 439–445
November 10–11, 2021.

	En↔Id	En↔Jv	En↔Ms	En↔Ta	En↔Tl	Id↔Jv	Id↔Ms	Id↔Ta
No filter	36.15M	1.52M	7.43M	1.19M	6.97M	0.75M	4.23M	0.46M
Filtered	33.67M	1.41M	7.03N	1.06M	6.15M	0.67M	3.97M	0.41M
	Id↔Tl	Jv↔Ms	Jv↔Ta	Jv↔Tl	Ms↔Ta	Ms↔Tl	Ta↔Tl	
No filter	2.56M	0.41M	0.06M	0.74M	0.33M	1.27M	0.53M	
Filtered	2.18M	0.36M	0.05N	0.61M	0.30M	1.09M	0.44M	

Table 1: Number of sentences in bitext data sets.

	En	Id	Jv	Ms	Ta	Tl
No filter	126.44M	5.46M	0.41M	1.87M	2.06M	0.41M
Filtered	113.36M	5.26M	0.38N	1.85M	2.03M	0.39M

Table 2: Number of sentences in monolingual data sets.

- Deduplicate training data.
- Remove all instances of evaluation data from the training data.
- Filter sentences that are longer than 250 tokens or length ratio upper than 3.

For monolingual data, we still employ those rules except the length ratio filter. See Table 1 for the statistics of bitext data sets and Table 2 for monolingual data sets.

3 System Overview

3.1 Base Systems

Our systems are based on the Transformer architecture (Vaswani et al., 2017) as implemented in TenTrans [4], a unified end-to-end multilingual and multi-task training platform. We first train a model following the Transformer *base* setup to jointly training all language pairs as our base system. Then, inspired by Wang et al. (2019), we experiment with raising network capacity by increasing encoder/decoder layers and feed-forward networks. We found that using a deeper encode layer (24) and a larger feed-forward network size (4096) can provide reasonable performance improvements while maintaining manageable network size and not increasing inference time.

Because of the recent popularity of using large-scale pre-training models to fine-tune specific languages and tasks (Fan et al., 2020; Liu et al., 2020), we use the pre-trained model FLORES-101 released by the organizer to fine-tune on the bitext data. This system has further improved our translation performance in all thirty translation directions. Note that to fine-tune FLORES-101 we train our models using FAIRSEQ (Ott et al., 2019).

3.2 Forward-Translation and Back-Translation

Back-translation is an effective and common way to boost translation quality by using monolingual data to produce pseudo training parallel data. As opposed to back-translation, forward translation use source-side monolingual data to translate into the target language, and can be quite effective in some cases (Bogoychev and Sennrich, 2019). Wu et al. (2019) has shown that when monolingual data from source and target languages are used together to produce pseudo data, the translation quality is best, and the experimental performance will be improved with the increase of data.

In this work, considering the excellent performance of forward-translation and back-translation, we use both methods together. For translation directions with more than 5 million bitext data, such as En↔Id, En↔Ms, En↔Tl, we separately train an individual model for each direction and use it for the pseudo-corpus generation. For other translation directions with less than 5 million bitext data, we use the baseline system of all language pairs jointly training for translating pseudo sentences. Due to a large amount of English monolingual data, English monolingual sentence was randomly divided into 13.36M, 25M, 25M, 25M, and 25M for En→Id, En→Jv, En→Ms, En→Ta, and En→Tl translation respectively. All monolingual data of Id, Jv, Ms, Ta, and Tl are used in translation to all other directions.

3.3 In-domain Data Selection

The training data is provided by the publicly available Opus repository, which contains data of various quality from a variety of domains, while the hidden test set is the same domain as the provided dev and devtest datasets. After fine-tuning on a

[4] https://github.com/TenTrans/TenTrans

	En→Id	En→Jv	En→Ms	En→Ta	En→Tl	Id→En	Id→Jv	Id→Ms	Id→Ta	Id→Tl
0.7	3.83M	7.62K	1.15M	0.21M	0.36M	12.32M	0.05M	2.11M	0.23M	0.43M
0.8	3.44M	6.93K	1.05M	0.19M	0.34M	12.12M	0.05M	2.08M	0.22M	0.42M
0.9	2.82M	5.99K	0.89M	0.16M	0.30M	11.82M	0.04M	2.02M	0.22M	0.41M
0.99	1.24M	3.14K	0.44M	0.09M	0.17M	10.73M	0.03M	1.84M	0.20M	0.37M
	Jv→En	Jv→Id	Jv→Ms	Jv→Ta	Jv→Tl	Ms→En	Ms→Id	Ms→Jv	Ms→Ta	Ms→Tl
0.7	59.99K	41.90K	15.82K	9.98K	14.77K	3.65M	2.20M	24.77K	0.18M	0.33M
0.8	57.12K	40.74K	14.83K	9.78K	14.29K	3.59M	2.17M	23.32K	0.18M	0.33M
0.9	53.05K	39.07K	13.76K	9.47K	13.61K	3.49M	2.11M	21.24K	0.17M	0.32M
0.99	41.59K	34.07K	10.47K	8.37K	11.39K	3.14M	1.91M	15.36K	0.16M	0.29M
	Ta→En	Ta→Id	Ta→Jv	Ta→Ms	Ta→Tl	Tl→En	Tl→Id	Tl→Jv	Tl→Ms	Tl→Ta
0.7	0.72M	0.28M	17.60K	0.21M	0.20M	1.12M	0.43M	17.95K	0.31M	0.15M
0.8	0.71M	0.27M	16.83K	0.20M	0.19M	1.11M	0.42M	17.32K	0.30M	0.15M
0.9	0.69M	0.27M	15.73K	0.20M	0.19M	1.09M	0.41M	16.38K	0.30M	0.15M
0.99	0.63M	0.24M	12.49K	0.18K	0.16M	1.03M	0.38M	0.01M	0.28M	0.13M

Table 3: Data filtered at different thresholds for all language pairs.

mixture of authentic bitext and pseudo-data, we select domain-specific data from the bitext and continue to fine-tune to further improve translation quality.

Due to the scarcity of in-domain data, we utilize pre-trained language model multilingual BERT (Devlin et al., 2019) to train a domain classifier for extracting in-domain sentences from authentic bilingual sentences. To train the domain classifier, we consider all available dev data as positive data, and randomly sample bilingual data as negative samples. At the same domain test set, the domain classifier recognition accuracy is achieved at 93.97%. We select sentences predicted to be positive with a probability greater than threshold 0.7 to form an in-domain corpus.

3.4 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) is a way to train a smaller network of students to perform better by learning from a larger teacher model. On this basis, sequence-level knowledge distillation trains the student model on the new data generated by the teacher model to further improve the performance of the student (Kim and Rush, 2016).

A multilingual translation model that trains too many languages at the same time may degrade performance (Xie et al., 2021), especially involving 30 translation directions in this work. It makes it harder for the model to accommodate all language pairs. Based on this, we fine-tune the FLORES-101 model on five language pairs with En→Ta, Id→Ta, Jv→Ta, Ms→Ta, Tl→Ta to produce an Any-to-Ta specific translation model (Tan et al., 2019). These five language pairs are chosen because they do not perform very well and have more room for improvement. We used this model as the teacher model to translate the training data of the five language pairs. The new data was then combined with data of other language pairs to train the student model.

3.5 Gradual Fine-tuning

Fine-tuning can improve the machine translation model by adapting the initial model trained on abundant but less domain-specific examples to the data in the target domain. This domain adaptation is usually accomplished with a phase of fine-tuning. While Xu et al. (2021) prove that gradual fine-tuning over a multi-stage process can yield substantial further gains. Intuitively, the model is iteratively trained to convergence on data whose distribution progressively approaches that of the in-domain data, similar to the curriculum learning strategy (Bengio et al., 2009; Kocmi and Bojar, 2017).

In this work, we use gradual fine-tuning combined with in-domain data selection. After training the domain classifier, authentic bilingual sentences with positive predictions and probabilities greater than the thresholds of 0.7, 0.8, 0.9, and 0.99 are selected to form in-domain corpora with different similarity degrees. Data statistics with different thresholds are shown in the Table 3. The higher the threshold, the more the selected data fits into the domain of the dev set and test set. We started with a gradual fine-tuning on the domain-specific data selected at the 0.7 thresholds, followed by the 0.8 thresholds, and so on.

System	Average BLEU
Transformer	22.25
+ F&B	25.05
+ deep (24)	25.43
FLORES-101	15.38
+ Fine-tuning	24.23
+ F&B	26.50
+ Data Selection	27.24
+ Gradual FT	28.03
+ KD	28.15
+ Recover 12	28.32
Averaging	**28.94**

Table 4: Average BLEU (%) scores of all systems. The '+' means the approach added to the system over the previous line.

To further improve performance, we selected 12 language pairs that are significantly better than the baseline system. We consider them BLEU-sensitive and performance-friendly language pairs, which include En→Ta, Id→Ta, Jv→En, Jv→Ta, Jv→Tl, Ms→Ta, Ta→En, Ta→Id, Ta→Jv, Ta→Ms, Ta→Tl and Tl→Ta. After the gradual fine-tuning, we recover all the authentic bilingual sentences of these 12 language pairs, while the training sentences of other language pairs are still the training data when the threshold is 0.99. We continue to fine-tune the multilingual translation model. We find that the results still improve on these 12 language pairs and the performance of other language pairs is almost unchanged.

3.6 Model Averaging

Model averaging is typically used between 5 or 10 adjacent checkpoints on the same system. It is almost impossible to average different systems because neurons or parameters at the same location in different systems may be responsible for completely different knowledge or responsibilities. Our systems kept the random seeds consistent, and the training data did not differ too much, so we tried a variety of model averaging methods to see whether the performance was improved. We finally chose average multiple checkpoints in a single system, and then averaged on different systems. In this way, the translation result can be further improved.

4 Experiments

4.1 Experiment Settings

Except for the FLORES-101 fine-tuning experiments training on 48 NVIDIA P40 GPUs, the rest of our experiments are carried out with 16 NVIDIA P40 GPUs. Our model apply Adam (Kingma and Ba, 2015) as optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We set the label smoothing to 0.2 and the dropout rate to 0.3. The initial learning rate is set to 5e-4 varied under a warm-up strategy with 4000 steps. For training, the batch size is 4096 tokens per GPU. For fine-tuning FLORES-101, we apply a temperature sampling strategy with sampling temperature T = 1.5 (Arivazhagan et al., 2019). During inference, we decode with beam search and set beam size to 4 for all language pairs. The translation results we reported is detokenized and then the quality is evaluated using the 4-gram case-sensitive BLEU (Papineni et al., 2002) with the *SacreBLEU* tool (Post, 2018).[5]

4.2 Main Results

Results for all of our systems are shown in Table 4. For convenience, we only report the average BLEU for 30 language pairs. The detailed BLEU scores for each language pair of systems implemented by TenTrans tool are shown in Table 5, and the relevant systems for fine-tuning FLORES-101 are shown in Table 6.

As shown in Table 4, we found that the baseline system with fine-tuning FLORES-101 performed better than the baseline system with no pre-training model (24.23 vs. 22.25). Forward-translation and back-translation (F&B) greatly improved the translation performance in both TenTrans (25.05 vs. 22.25) and FLORES-101 (26.50 vs. 24.23) frameworks. The results of individual models for forward-translation and back-translation are shown in Table 7. Deep Transformer with 24 encoder layers further improves translation results, but still not as high as fine-tuning FLORES-101 systems. Given the excellent performance of the pre-trained model, our subsequent series of approaches are based on fine-tuning FLORES-101.

In-domain data selection is restricted to in-domain data size (threshold 0.7), but we also obtain a solid improvement of 0.74 BLEU on average. Gradual fine-tuning (Gradual FT) is also effective, which enables the model to potentially better fit

[5]BLEU+case.mixed+numrefs.1+smooth.exp+tok.spm +version.1.5.0

System	En→Id	En→Jv	En→Ms	En→Ta	En→Tl	Id→En	Id→Jv	Id→Ms	Id→Ta	Id→Tl
Transformer	43.37	18.12	40.44	15.99	29.21	36.39	17.93	33.18	13.98	24.47
+ F&B	42.93	23.54	41.73	22.18	30.65	36.58	21.65	33.70	17.93	24.98
+ deep (24)	44.13	23.61	42.59	22.76	31.38	37.95	21.50	34.34	18.13	25.69
	Jv→En	Jv→Id	Jv→Ms	Jv→Ta	Jv→Tl	Ms→En	Ms→Id	Ms→Jv	Ms→Ta	Ms→Tl
Transformer	23.38	24.33	21.60	9.27	16.06	35.21	34.29	16.61	13.72	23.46
+ F&B	26.20	26.04	24.55	14.43	19.39	37.08	34.50	21.09	18.15	24.10
+ deep (24)	26.25	25.72	24.08	14.76	19.95	38.02	34.73	21.13	18.65	24.95
	Ta→En	Ta→Id	Ta→Jv	Ta→Ms	Ta→Tl	Tl→En	Tl→Id	Tl→Jv	Tl→Ms	Tl→Ta
Transformer	14.61	13.12	5.83	13.21	14.55	34.28	28.05	13.53	26.35	12.93
+ F&B	21.04	15.76	10.62	16.39	17.55	36.52	27.87	18.48	27.75	18.17
+ deep (24)	21.19	17.02	9.70	17.04	18.20	36.52	28.85	17.89	27.93	18.35

Table 5: Results of the systems implemented by TenTrans on the devtest set.

System	En→Id	En→Jv	En→Ms	En→Ta	En→Tl	Id→En	Id→Jv	Id→Ms	Id→Ta	Id→Tl
FLORES-101	37.28	15.35	33.40	3.38	6.38	33.75	16.55	29.45	1.36	8.07
+ Fine-tuning	44.79	17.60	41.39	20.14	30.71	38.86	18.26	34.56	16.64	26.50
+ F&B	44.06	23.43	43.01	22.67	32.49	40.29	21.53	35.31	18.86	26.62
+ Data Selection	44.73	22.02	43.46	24.32	33.14	41.05	20.81	35.66	18.60	28.28
+ Gradual FT	45.30	23.13	43.74	25.77	33.43	41.53	22.04	35.88	19.96	28.68
+ KD	45.49	24.40	43.97	28.21	33.90	40.99	22.07	35.95	23.52	28.99
+ Recover 12	45.52	**24.46**	44.02	**28.89**	**34.27**	40.94	22.08	35.93	**23.67**	28.82
Averaging	**46.15**	24.16	**44.47**	27.24	34.19	**42.00**	**22.64**	**36.55**	22.87	**29.45**
	Jv→En	Jv→Id	Jv→Ms	Jv→Ta	Jv→Tl	Ms→En	Ms→Id	Ms→Jv	Ms→Ta	Ms→Tl
FLORES-101	20.90	22.77	18.98	3.73	12.07	34.24	32.18	15.18	2.22	9.64
+ Fine-tuning	25.38	25.95	23.08	9.78	17.95	38.46	35.87	17.46	16.61	25.02
+ F&B	27.23	26.69	24.98	15.46	20.93	39.38	36.20	21.73	19.34	27.17
+ Data Selection	30.67	28.92	26.57	15.10	21.68	41.60	36.37	21.21	20.33	27.25
+ Gradual FT	31.63	29.69	27.16	16.82	22.40	41.68	36.72	21.63	21.41	27.40
+ KD	30.79	29.44	26.84	17.28	22.93	41.55	36.73	20.80	23.30	**28.50**
+ Recover 12	30.76	29.55	26.84	17.30	23.05	41.78	36.76	21.57	**23.63**	28.08
Averaging	**31.82**	**30.22**	**27.93**	**18.47**	**23.74**	**42.54**	**37.27**	**22.15**	23.47	28.34
	Ta→En	Ta→Id	Ta→Jv	Ta→Ms	Ta→Tl	Tl→En	Tl→Id	Tl→Jv	Tl→Ms	Tl→Ta
FLORES-101	8.41	5.36	3.11	4.89	3.30	26.10	20.43	11.52	18.00	3.46
+ Fine-tuning	19.91	16.44	7.76	15.96	17.04	36.61	30.85	12.68	28.71	15.80
+ F&B	23.68	17.75	10.83	17.61	18.91	39.80	30.84	**19.48**	29.71	18.97
+ Data Selection	25.11	19.14	10.00	18.91	20.20	41.42	32.71	16.27	30.97	20.61
+ Gradual FT	25.29	20.07	**12.86**	19.68	20.85	41.42	33.55	18.12	31.33	21.75
+ KD	25.04	19.03	11.27	18.91	21.20	40.95	32.83	17.06	31.19	21.33
+ Recover 12	25.04	19.59	11.08	19.07	20.92	41.17	32.96	18.58	31.30	21.93
Averaging	**26.19**	**21.21**	12.75	**20.25**	**21.87**	**42.06**	**34.17**	19.00	**31.93**	**23.04**

Table 6: Results of the systems about FLORES-101 on the devtest set. The '+' means the approach added to the system over the previous line. "Average" stands for averaging the model of the three best checkpoints of the "+ Gradual FT" system and the "+ Recover 12" system respectively.

System	En→Id	Id→En	En→Ms	Ms→En	En→Tl	Tl→En
Individual	46.08	40.72	42.95	38.76	31.84	37.94

Table 7: Results of the individual models for forward-translation and back-translation.

System	AVE	En→Ta	Id→Ta	Jv→Ta	Ms→Ta	Tl→Ta
FLORES-101 + Fine-tuning	15.79	20.14	16.64	9.78	16.61	15.80
Any-to-Ta	22.83	28.75	23.40	17.02	23.27	21.70

Table 8: Results of FLORES-101 fine-tuning on Any-to-TA language pairs only.

Dataset	sp_bleu
flores101-small2-test	28.89
flores101-small2-dev	29.25
flores101-small2-devtest	28.94

Table 9: Flores MT Evaluation (Small task 2) results: https://dynabench.org/models/460

the distribution of the target domain. The knowledge distillation, however, has not brought much improvement (28.15 vs. 28.03). The translation performance of the teacher model is shown in Table 8. We guess that it may be because the translation quality of the teacher model is not excellent enough, which leads to the improvement of the student model is not satisfactory. We then recovered bilingual sentences for 12 BLEU-sensitive language pairs. As shown in Table 6, the performance of these 12 language pairs improved significantly, while the results of the other language pairs barely changed, so our average BLEU improved further. For model averaging, we tried different combinations and finally found that averaging the three best checkpoints in "+ Gradual FT" and "+ Recover 12" will produce the best performance (28.94).

4.3 Submitted Results

As shown in Table 9, we ultimately chose the best-performing model on devtest to submit to Dynabench [6] and achieve 28.89 in the hidden test set.

5 Conclusion

This paper introduced our TenTrans submissions on WMT21 large-scale multilingual machine translation small task 2. Our main exploration is using more diversified architectures and fine-tuning strategy, utilizing forward-translation and back translation and approaches including in-domain data selection, knowledge distillation, and gradual fine-tuning. We experimented with these methods and continuously improve our system performance. On the whole, all of our systems performed competitively and ranked 3rd on the leaderboard.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 41–48.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *CoRR*, abs/1911.03362.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Shuhao Gu, Yang Feng, and Wanying Xie. 2021. Pruning-then-expanding model for domain adaptation of neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3942–3952. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural*

[6] https://dynabench.org/flores

Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 1317–1327. The Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kocmi and Ondrej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, pages 379–386.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: 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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 963–973.

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 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The niutrans machine translation system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 528–534. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1810–1822.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215. Association for Computational Linguistics.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5725–5737. Association for Computational Linguistics.

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton W. Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. *CoRR*, abs/2103.02205.

Multilingual Machine Translation Systems from Microsoft for WMT21 Shared Task

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, Furu Wei
Microsoft Corporation
{t-jianya,shumma,haohua,dozhang,lidong1,shaohanh}@microsoft.com
{alferre,saksingh,hanyh,xiaso,fuwei}@microsoft.com

Abstract

This report describes Microsoft's machine translation systems for the WMT21 shared task on large-scale multilingual machine translation. We participated in all three evaluation tracks including Large Track and two Small Tracks where the former one is unconstrained and the latter two are fully constrained. Our model submissions to the shared task were initialized with DeltaLM[1], a generic pre-trained multilingual encoder-decoder model, and fine-tuned correspondingly with the vast collected parallel data and allowed data sources according to track settings, together with applying progressive learning and iterative back-translation approaches to further improve the performance. Our final submissions ranked first on three tracks in terms of the automatic evaluation metric.

1 Introduction

Recently, multilingual neural machine translation has attracted lots of attention because it enables one model to translate between multiple languages (Dong et al., 2015; Johnson et al., 2017; Arivazhagan et al., 2019; Dabre et al., 2020; Philip et al., 2020; Lin et al., 2021). To improve the performance of the multilingual translation models, there are various approaches on the training methods (Aharoni et al., 2019; Wang et al., 2020a,c), the model structures (Wang et al., 2018; Gong et al., 2021; Zhang et al., 2021a), and the data augmentation (Tan et al., 2019; Pan et al., 2021). M2M (Fan et al., 2020) leverages the large-scale data mined from the web data and explore the strategies to scale the model size and train the model effectively. Meanwhile, the multilingual pre-trained language models have proven beneficial for the multilingual machine translation models. mBART (Liu et al., 2020) pre-trains a multilingual model with the multilingual denoising objective to improve the multilingual machine translation.

[1] https://aka.ms/deltalm

In this work, we explore the effects of different advanced approaches for multilingual machine translation models, especially on the large-scale dataset. We first explore the way to leverage the pre-trained language models that have been trained with large-scale monolingual data. We use the public available DeltaLM-Large checkpoint to initialize the model. DeltaLM (Ma et al., 2021) is a multilingual pre-trained encoder-decoder model, which has been proven useful for multilingual machine translation.

We further explore the training methods and the data augmentation to improve the model. For efficient training, we apply progressive learning (Li et al., 2020; Zhou et al., 2021; Zhang et al., 2021b) to our model that continue-trains a shallow model into a deep model. Specifically, we first train a model with 24 encoder layers, and then continue-train it by adding 12 layers on the top of the encoder. As for the data augmentation, we implement iterative back-translation (Hoang et al., 2018; Dou et al., 2020) that back-translates the data for multiple rounds. Due to the limits of time and GPU memories of the shared task, we do not explore other approaches like mixture-of-experts (MOE) and model ensemble.

We participated in all three tracks including Large Track, Small Track #1, and Small Track #2. Our final submissions are fine-tuned from DeltaLM with the allowed data sources according to the track settings, followed by progressive learning and iterative back-translation. The submissions on three tracks all rank first in terms of the automatic evaluation metric.

2 Data

Large Track The monolingual and bilingual data are collected from multiple sources, including CCAligned (El-Kishky et al., 2020), CCMatrix (Schwenk et al., 2021), OPUS-100 (Zhang et al., 2020), JW300 (Agic and Vulic, 2019), Tatoeba

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 446–455
November 10–11, 2021.

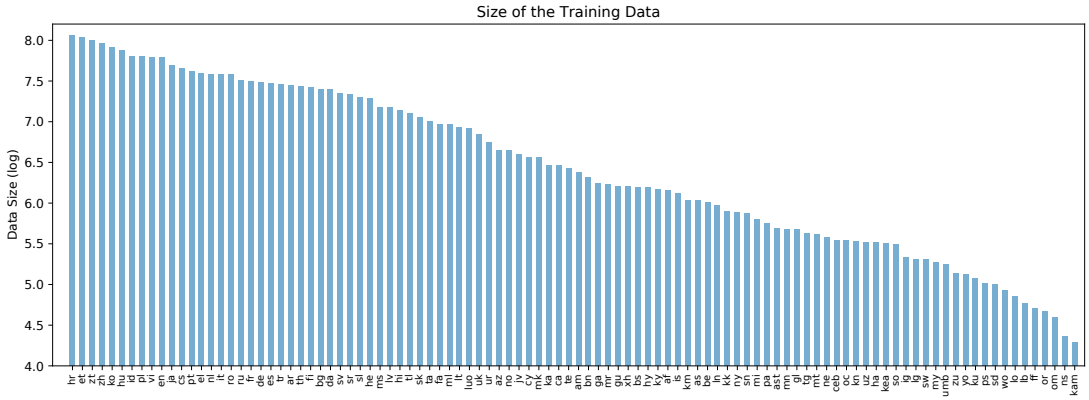


Figure 1: Dataset statistics of the bilingual data of the 102 languages. For better visualization, we apply the logarithmic function (base 10 logarithm) to the size of the training data. Each column denotes the data size of a language that was paired with the remaining 101 languages. For example, the first column denotes the number of bilingual sentence pairs that contain sentences from language hr.

(Tiedemann, 2012), WMT2021 news track[2], multilingual track data[3], and our in-house data. To improve the translation quality of non-English languages, we construct dual-pseudo parallel data (or dual-pseudo data briefly) in which the source and target sides per each sentence pair are translated from the same monolingual English sentence respectively. The Wikipedia English monolingual sentences are translated to other 70 languages by leveraging various machine translation models including in-house MT models, M2M (Fan et al., 2020), the multilingual model of small tracks, and our intermediate multilingual MT model.

Finally, the training data was split into three parts: the bitext data (1.7B parallel sentences from 394 language pairs), the back-translation (1.4B parallel sentences from 45 language pairs), and the dual-pseudo data (8.7B parallel sentences of 70 languages from 4830 language pairs). Figure 1 lists the statistics of the bilingual training data size of 102 languages.

Small Track #1 We use the constrained monolingual and bilingual data of 6 languages (Croatian, Hungarian, Estonian, Serbian, Macedonian, and English) provided by the shared task. According to the statistics, the bitext data contains 273M sentence pairs of all translation directions. Inspired by the previous work, we leverage the multilingual iterative back-translation method with one single multilingual model to generate parallel pseudo data. For En→X and X→En directions, we generate the back-translation data of 390M sentence pairs. As for X→Y directions, we generate the dual-pseudo data of 1.18B sentence pairs, where X and Y stand for any two non-English languages.

Small Track #2 The monolingual and bilingual corpora of 6 languages (Javanese, Indonesian, Malay, Tagalog, Tamil, and English) provided by the shared task are used for the multilingual model training, containing 98M bilingual data, 256M generated back-translation data, and 860M generated dual-pseudo data.

3 Large-scale Data Augmentation

In this section, we introduce details about how to perform the iterative back-translation method (Hoang et al., 2018) to augment data. We use different models for data augmentation according to different tracks. For the small tracks, the multilingual models were trained over the constrained data sets to generate data. For the large track, we leverage the M2M model (Fan et al., 2020), the intermediate multilingual MT models, and in-house MT models to generate different language pairs' data respectively, so as to play their respective advantages to enhance the data generation quality.

In practice, both the monolingual and bilingual corpora are effectively utilized in three ways: 1) For the back-translation data of X→En and En→X directions, we used the best model to generated X data accordingly by back-translating monolingual English Wikipedia data; 2) For the dual-pseudo data of X→Y directions, they are generated by back-translating the same English text to X and Y

[2] http://statmt.org/wmt21/translation-task.html

[3] http://data.statmt.org/wmt21/multilingual-task/

respectively. Alternatively, when the monolingual data of either X or Y is enough, we also directly perform back-translation between X and Y to obtain pseudo parallel data; 3) We try to augment existing bilingual corpora with the third language. Given the bilingual corpus (X_1, Y_1), we generate pseudo parallel corpus of (X_1, Y_2) and (X_2, Y_1) by back-translating X_1 to X_2 and Y_1 to Y_2, where X_2 and Y_2 are non-English languages.

4 Preprocessing

Filtering To enhance the model performance, we remove the noisy sentence pairs with the incorrect language identification or character encoding. More specifically, we remove the sentences longer than 1024 words and truncate the sentence to 512 tokens. We also construct three corpora after tokenization with different length ratio limitations, i.e. $\{1.5, 2.0, 2.5, 3.0\}$, between the source and the target sentence. Our multilingual model is first trained on the entire noisy data set and then continually tuned on cleaner data with descending length ratio, where the number of training directions is also gradually reduced by removing noisy language pairs. Therefore, we can progressively fine-tune the multilingual model in an efficient way (noisy corpora $\rightarrow$ clean corpora $\wedge$ numerous directions $\rightarrow$ selected directions $\wedge$ shallow encoder layers $\rightarrow$ deep encoder layers). Besides, to clean the back-translation corpora, we remove the sentences containing unknown tokens (`[UNK]`). Regarding the language Sr (Serbian), those sentences comprised of Latin characters in training data were also discarded since we found that the validation sets use Cyrillic script for this language instead.

Tokenization After data filtering, we use the SentencePiece (Kudo and Richardson, 2018) to tokenize all raw training, validation, and test data sets, where the SentencePiece model is consistent with the one used for DeltaLM (Ma et al., 2021). We shuffled the whole training dataset before launching the training of multilingual models. The input sentence is prefixed with the language tag to indicate the translation direction.

5 Model and Training

5.1 DeltaLM

We adopt the `DeltaLM_large` architecture as the backbone model for all our experiments, which has 24 Transformer encoder layers and 12 interleaved decoder layers with an embedding size of 1024, a dropout of 0.1, the feed-forward network size of 4096, and 16 attention heads. We directly initialize our model with the public available DeltaLM large checkpoint[4].

5.2 Multilingual Fine-tuning

The training data was split into the bitext corpora $D_b = \{D_b^1, \ldots, D_b^u\}$, the back-translation corpora $D_{bt} = \{D_{bt}^1, \ldots, D_{bt}^v\}$, and the dual-pseudo corpora $D_{dp} = \{D_{dp}^1, \ldots, D_{dp}^w\}$, where u, v, w represent the number of the corpora of different translation directions. The multilingual model with parameters Θ is jointly trained over the corpora to optimize the combined objective as below:

$$\begin{aligned}\mathcal{L}_{MT} = &-\lambda_1 \sum_{i=1}^{u} \mathbb{E}_{x,y \in D_b^i} \left[-\log P(y|x;\Theta)\right] \\ &-\lambda_2 \sum_{i=1}^{v} \mathbb{E}_{x,y \in D_{bt}^i} \left[-\log P(y|x;\Theta)\right] \\ &-\lambda_3 \sum_{i=1}^{w} \mathbb{E}_{x,y \in D_{dp}^i} \left[-\log P(y|x;\Theta)\right]\end{aligned} \tag{1}$$

where x, y denote the sentence pair in the bilingual corpus. $\mathcal{L}_{MT}$ is the combined translation objective of the multilingual model. $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 + \lambda_2 + \lambda_3 = 1.0$) are used to balance the training objectives of the bitext corpora, the back-translation corpora, and the dual-pseudo corpora. In this work, we first set $\lambda_1 = 0.33, \lambda_2 = 0.33, \lambda_3 = 0.33$ and then reset $\lambda_1 = 0.6, \lambda_2 = 0.2, \lambda_3 = 0.2$ to focus more on the bitext corpora avoiding the noise introduced by pseudo data.

We follow the dynamic temperature-based data-sampling strategy (Fan et al., 2020; Wang et al., 2020b) to ease the underrepresentation of low-resource languages. The probability of picking a language is proportional to its number of sentences D_l, i.e., $p_l = \frac{D_l}{\sum_i D_i}$. We set the temperature $T = 5$ to rescale and control the distribution $p_l^{\frac{1}{T}}$. It can balance the samples between the high-resource languages and the low-resource languages.

5.3 Progressive Learning

We implement the progressive training method to train the model from shallow to deep (Li et al., 2020). The training process can be divided into two stages. In the first stage, the pre-trained DeltaLM model with 24 encoder layers and 12 decoder layers is directly adopted to initialize the multilingual

[4] https://aka.ms/deltalm

translation model with the same architecture. The shallow translation model with 24 encoder layers and 12 decoder layers is fine-tuned on all available multilingual corpora. In the second stage, we increase the depth of the encoder from 24 layers to 36 layers, where the bottom 24 layers of the encoder are initialized with the shallow model's encoder and the top 12 layers are randomly initialized. Then we perform continue training. The deeper encoders enlarge the model's capacity, but no much extra decoding cost is introduced.

5.4 Training Details

We train multilingual models with the Adam optimizer (Kingma and Ba, 2014) ($\beta_1 = 0.9$, $\beta_2 = 0.98$). The learning rate is set as 1e-4 with a warm-up step of $4,000$. The models are trained with the label smoothing with a ratio of 0.1. All experiments are conducted on 64 NVIDIA V100 or 32 A100 GPUs. The batch size is 1536 or 2048 tokens per GPU and the model is updated every 32 (for 64 V100 GPUs) or 64 (for 32 A100 GPUs) steps to simulate the large batch size.

5.5 Decoding

To enhance the performance of low-resource language pairs for X→Y directions, we adopt the pivot-based translation method (Kim et al., 2019). We use English as the pivot language and employ a unified model to perform the pivot-based translation. When the performance of X→Y directions on the validation set is better than the pivot-based translation X→En ∧ En→Y, we directly translate the language X into Y. Otherwise, we translate them in the pivot way. This approach is used for the submission to Large Track and Small Track #2. As for Small Track #1, we do not use the pivot-based translation.

6 Evaluation Results

Following the previous work (Goyal et al., 2021), we use the dev and the devtest of the FLORES-101 benchmark as our validation set and test set respectively. During the inference, the beam search strategy is performed with a beam size of 4 for the target sentence generation. We set the length penalty as 1.0 by default. The last N checkpoints ($N = \{1, 5, 10, 15, 20\}$) are averaged for evaluation and we select the best checkpoint based on the performance on the validation set. We report the SentencePiece-based BLEU using spBLEU[5].

6.1 Large Track

Given the unbalanced large-scale multilingual corpora, we use the hybrid strategy for the translation for Large Track. The pivot-based translation is more suitable for the low-resource translation direction between non-English languages since the corpora of X→Y are commonly scarce. Our model with the 36 encoder layers significantly outperforms the shallow counterpart with the 24 encoder layers, which indicates that using a deep encoder and shallow decoder is a good trade-off between the translation quality and the decoding speed. Table 1 shows that our model with the hybrid strategy gets the best performance with less inference cost than the pivot-based translation which costs double inference time compared to the direct translation. We build a massively multilingual neural machine model, which translates between any pair of 102 languages. In Figure 2 and Figure 3, we reported the spBLEU scores of the shallow model with 24 encoder layers and 6 decoder layers and our best multilingual model with 36 encoder layers and 12 decoder layers in all translation directions, where the languages are ordered alphabetically by the language code. Nearly 30% translation directions adopt the pivot-based translation, where the zero-resource and low-resource translation directions lack of supervised training data tend to be chosen for pivot-based translation.

6.2 Small Track #1

In Table 2, we compare the performance of M2M with our method in different architectures on any to English (X→En), English to any (En→X), and the translation between any non-English languages (X→Y). Both En→X and X→En contain 5 directions, while X→Y have 20 directions. Given the enormous bilingual and back-translation data of Small Track #1, we are able to perform the direct translation for all X→Y directions. Furthermore, we explore the deep encoder (36 encoder layers) and shallow decoder (12 decoder layers) considering the limited inference time. From Table 2, we can observe that the largest model (36 encoder layers and 12 decoder layers) has a significant improvement of +9.41 BLEU points over the strong M2M baseline.

[5] https://github.com/ngoyal2707/sacrebleu.git

	#Languages	#Params	#Layers	$\text{Avg}_{X\rightarrow En}$	$\text{Avg}_{En\rightarrow Y}$	$\text{Avg}_{X\rightarrow Y}$	Avg_{all}
M2M (Fan et al., 2020)	102	175M	6/6	15.43	12.02	5.85	6.00
	102	615M	12/12	20.03	16.21	7.66	7.86
DeltaLM + Zcode (Direct)	102	711M	24/6	30.39	23.52	11.21	11.52
	102	862M	24/12	33.09	27.21	13.56	13.89
	102	1013M	36/12	33.35	27.39	14.34	14.65
DeltaLM + Zcode (Pivot)	102	711M	24/6	31.32	24.04	14.74	14.99
	102	862M	24/12	33.09	27.21	17.20	17.45
	102	1013M	36/12	33.35	27.39	17.36	17.62
DeltaLM + Zcode (Hybrid)	102	711M	24/6	31.32	24.04	14.76	15.01
	102	862M	24/12	33.09	27.21	17.27	17.52
	102	1013M	36/12	**33.35**	**27.39**	**17.44**	**17.70**

Table 1: Evaluation results of Large Track for M2M and our method of 102 languages on the devtest of the FLORES-101 benchmark. $\text{Avg}_{X\rightarrow En}$ denotes the average score of directions between other languages to English. $\text{Avg}_{X\rightarrow En}$ denotes the average score of directions between English and other languages. $\text{Avg}_{X\rightarrow Y}$ denotes the average score of directions between non-English languages to other non-English languages. Avg_{all} denotes the average result of all translation directions.

	#Languages	#Params	#Layers	$\text{Avg}_{X\rightarrow En}$	$\text{Avg}_{En\rightarrow Y}$	$\text{Avg}_{X\rightarrow Y}$	Avg_{all}
M2M (Fan et al., 2020)	102	175M	6/6	24.60	20.83	20.80	21.44
	102	615M	12/12	31.58	29.62	26.66	27.98
DeltaLM + Zcode (Direct)	6	862M	24/12	43.78	41.02	34.38	37.06
	6	1013M	36/12	**44.34**	**41.32**	**34.68**	**37.39**

Table 2: Evaluation results of Small Track #1 for M2M and our method of 6 languages (Croatian, Hungarian, Estonian, Serbian, Macedonian, English) on the devtest of the FLORES-101 benchmark. **DeltaLM + Zcode (Direct)** denotes the strategy that we choose the direct translation for all translation directions, where the target language symbol is prefixed to the input sentence to indicate the translation direction. Our mutilingual translation model is only trained on the constrained corpora of 6 languages provided by the shared task.

6.3 Small Track #2

Compared to Small Track #1 (273M bilingual pairs), Small Track #2 contains smaller while more unbalanced training data (93M bilingual pairs). Therefore, we consider the hybrid strategy for the X→Y translation directions. We separately calculate the BLEU scores of the direct and the pivot-based translation on the validation set. For those directions satisfying $\text{BLEU}_{direct}(X, Y) \geq \text{BLEU}_{pivot}(X, Y)$, we employ the direct translation. Otherwise, we use the pivot-based translation direction by first translating the source language to English and then to the target language. According to Table 3, **DeltaLM + Zcode (Hybrid)** outperforms both the direct and pivot-based translation by about +0.5 BLEU points. It confirms that the hybrid strategy is essential since the training data of the X→En and En→Y is easy to obtain while the X→Y is hard to obtain. The deep model with the 36 encoder layers and 12 decoder layers has comparable performance with the shallow model with the 24 encoder layers and 12 decoder layers, which may be caused by the overfitting problem on the low-resource directions.

6.4 Discussion on Progressive Learning

Given the pre-trained model and large-scale parallel data, we adopt progressive learning as an alternative to fine-tune the multilingual model on the multilingual translation task. Our multilingual model is first trained on the large-scale noisy data and then continues to be tuned on the clean data (Noisy Data → Clean Data), where the model is denoted as ③. Since the training data of K languages in the dual-pseudo parallel data is generated by the same English monolingual data, we are able to adopt all possible $K \times (K-1)$ training directions on the clean data. The performance of many translation directions is improved by the additional dual-pseudo data while the performance of other directions has been degraded compared to the initial model ④, due to the poor quality of some languages in the dual-pseudo data. Therefore, only the part of $K \times (K-1)$ training directions is selected to continue training the multilingual model (Numerous Directions → Selected Directions), which we denoted as ②. To further enlarge the model's capacity, we extend the shallow model with 24 encoder layers to the deep model with 36

	#Languages	#Params	#Layers	$\text{Avg}_{X\rightarrow En}$	$\text{Avg}_{En\rightarrow Y}$	$\text{Avg}_{X\rightarrow Y}$	Avg_{all}
M2M (Fan et al., 2020)	102	175M	6/6	18.95	15.16	9.49	12.01
	102	615M	12/12	24.67	19.14	12.11	15.38
DeltaLM + Zcode (Direct)	6	862M	24/12	43.12	39.78	28.69	32.94
	6	1013M	36/12	43.56	39.04	28.60	32.83
DeltaLM + Zcode (Pivot)	6	862M	24/12	43.12	39.78	29.02	33.17
	6	1013M	36/12	43.56	39.04	28.63	32.85
DeltaLM + Zcode (Hybrid)	6	862M	24/12	**43.12**	**39.78**	**29.38**	**33.40**
	6	1013M	36/12	43.56	39.04	28.99	33.09

Table 3: Evaluation results of Small Track #2 for M2M and our method of 6 languages (Javanese, Indonesian, Malay, Tagalog, Tamil, English) on the devtest of the FLORES-101 benchmark. **DeltaLM + Zcode (Hybrid)** denotes the strategy that we choose the pivot-based translation (X→En, En→X) for low-resource X→Y directions and direct translation for high-resource X→Y directions.

ID	Large Track	Avg_{all}
①	DeltaLM + Zcode	14.65
②	① - Shallow Model → Deep Model	13.89
③	② - Numerous Directions → Selected Directions	13.09
④	③ - Noisy Data → Clean Data	12.24

Table 4: Ablation study of the large track on devtest. DeltaLM + Zcode is fine-tuned on the multilingual translation task via progressive learning.

Track	Submission Name	Avg_{all}
Large	DeltaLM + Zcode (Microsoft)	16.63
Small #1	DeltaLM + Zcode (Microsoft-Small)	37.59
Small #2	DeltaLM + Zcode (Microsoft-Small)	33.89

Table 5: Submission results based on the hidden test sets of our method on three tracks, including Large Track, Small Track #1, and Small Track #2.

encoder layers, where the top 12 encoder layers are initialized by random parameters (Shallow Model → Deep Model). Putting them all together, we obtain the final model ① **DeltaLM + Zcode**. Table 4 summarizes the results of the ablation study of these approaches. It shows that each approach has a significant contribution to the final model. This proves the effectiveness of progressive learning that can gradually improve performance in different aspects.

7 Submissions

Considering the trade-off between the decoding time and the performance, we submit the model (24 encoder layers and 12 decoder layers) with the hybrid strategy to both the Large Track and Small Track #2, while the deep model (36 encoder layers and 12 decoder layers) with the direct translation is submitted to Small Track #1. Table 5 summarizes the evaluation results of our model on the hidden test sets. According to the final results on the leaderboard, **DeltaLM + Zcode** ranks first across three tracks.

8 Conclusion

This paper describes Microsoft's submission to the large-scale multilingual machine translation of the WMT21 shared task. Our multilingual translation model achieves substantial improvement over the baseline systems by fine-tuning the pre-trained language model DeltaLM. We further enhance the model performance with the progressive learning and the iterative back-translation methods. As a result, our submitted systems get the top evaluation results on three tracks, including Large Track, Small Track #1, and Small Track #2.

References

Zeljko Agic and Ivan Vulic. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *ACL 2019*, pages 3204–3210.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *NAACL 2019*, pages 3874–3884.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5):99:1–99:38.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL 2015*, pages 1723–1732.

Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. 2020. Dynamic data selection and weighting for iterative back-translation. In *EMNLP 2020*, pages 5894–5904.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. Ccaligned: A massive collection of cross-lingual web-document pairs. In *EMNLP 2020*, pages 5960–5969.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Hongyu Gong, Xian Li, and Dmitriy Genzel. 2021. Adaptive sparse transformer for multilingual translation. *CoRR*, abs/2104.07358.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *ACL 2018*, pages 18–24.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. In *EMNLP 2019*, pages 866–876.

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

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP 2018*, pages 66–71.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. In *EMNLP 2020*, pages 995–1005.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In *ACL 2021*, pages 293–305.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *TACL*, 8:726–742.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *ACL 2021*, pages 244–258.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *EMNLP 2020*, pages 4465–4470.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. Ccmatrix: Mining billions of high-quality parallel sentences on the web. In *ACL 2021*, pages 6490–6500.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *ICLR 2019*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *LREC 2012*, pages 2214–2218.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020a. Balancing training for multilingual neural machine translation. In *ACL 2020*, pages 8526–8537.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *EMNLP 2018*, pages 2955–2960.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020b. Multi-task learning for multilingual neural machine translation. In *EMNLP 2020*, pages 1022–1034.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020c. On negative interference in multilingual models: Findings and A meta-learning treatment. In *EMNLP 2020*, pages 4438–4450.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021a. Share or not? learning to schedule language-specific capacity for multilingual translation. In *ICLR 2021*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL 2020*, pages 1628–1639.

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. 2021b. Learning with feature-dependent label noise: A progressive approach. In *ICLR 2021*.

Baohang Zhou, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. 2021. An end-to-end progressive multi-task learning framework for medical named entity recognition and normalization. In *ACL 2021*, pages 6214–6224.

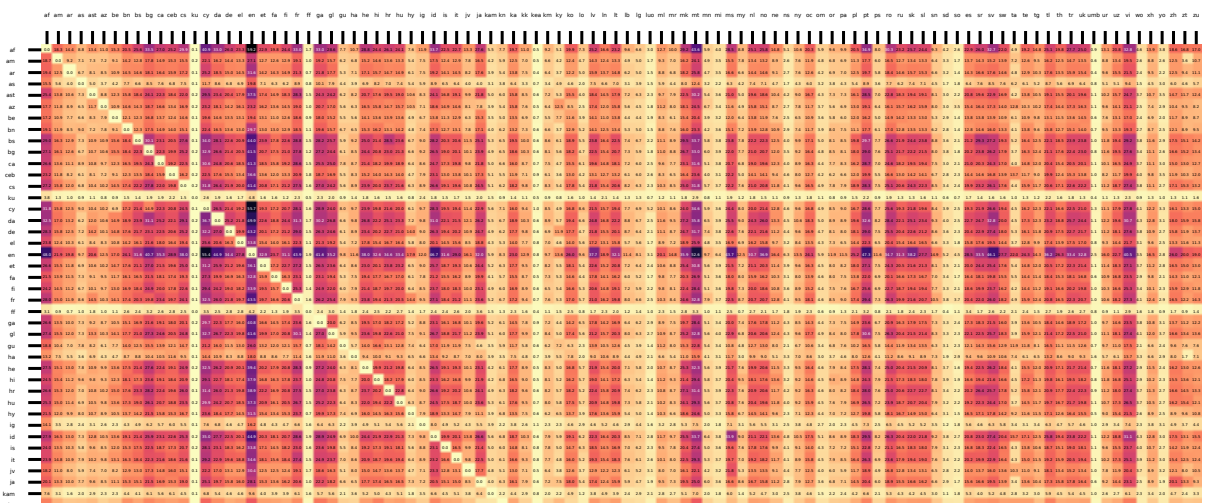


Figure 2: Evaluation results of our multilingual model (24 encoder layers and 6 decoder layers) on all translation directions on the FLORES-101 devtest set. The language x in the i-th row and language y in the j-th column denotes the translation direction from the language x to language y. For example, the cell of the 1-th row (af) and the 2-th column (am) represents the result of the translation direction af→am. The table shows the results of all translation directions of 102 languages.

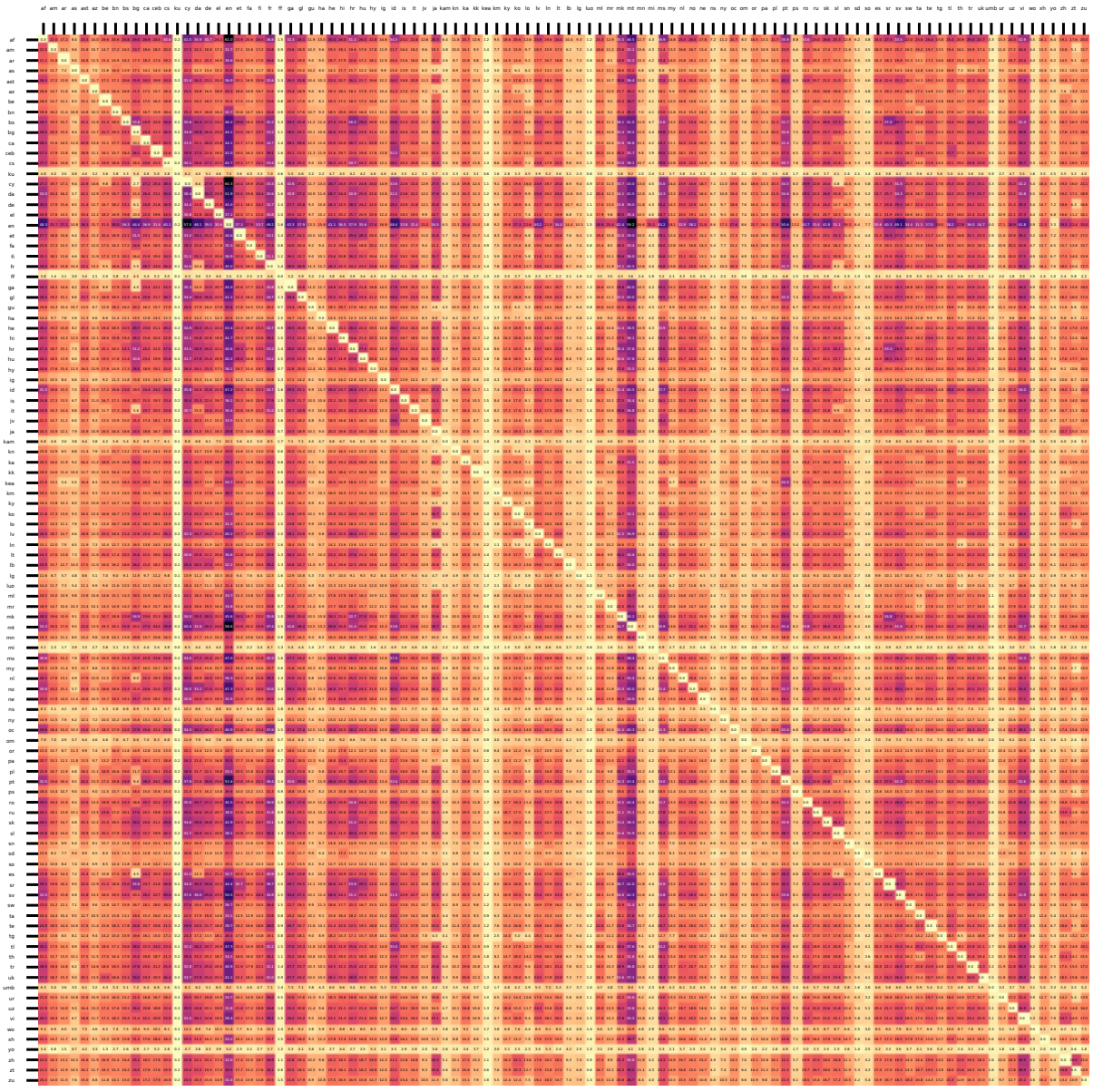


Figure 3: Evaluation results of our multilingual model (36 encoder layers and 12 decoder layers) on all translation directions on the FLORES-101 devtest set. The language x in the i-th row and language y in the j-th column denotes the translation direction from the language x to language y. For example, the cell of the 1-th row (af) and the 2-th column (am) represents the result of the translation direction af→am. The table shows the results of all translation directions of 102 languages.

HW-TSC's Participation in the WMT 2021 Large-Scale Multilingual Translation Task

Zhengzhe Yu, Daimeng Wei, Zongyao Li, Hengchao Shang,
Zhanglin Wu, Xiaoyu Chen, Jiaxin Guo, Minghan Wang,
Lizhi Lei, Min Zhang, Hao Yang, Ying Qin,
Huawei Translation Service Center, Beijing, China
{yuzhengzhe,weidaimeng,lizongyao,shanghengchao,
chenxiaoyu35,wuzhanglin2,guojiaxin1,wangminghan,
leilizhi,zhangmin186,yanghao30,qinying}@huawei.com

Abstract

This paper presents the submission of Huawei Translation Services Center (HW-TSC) to the WMT 2021 Large-Scale Multilingual Translation Task. We participate in Small Track #2, including 6 languages: Javanese (Jv), Indonesian (Id), Malay (Ms), Tagalog (Tl), Tamil (Ta) and English (En) with 30 directions under the constrained condition. We use Transformer architecture and obtain the best performance via multiple variants with larger parameter sizes. We train a single multilingual model to translate all the 30 directions. We perform detailed pre-processing and filtering on the provided large-scale bilingual and monolingual datasets. Several commonly used strategies are used to train our models, such as Back Translation, Forward Translation, Ensemble Knowledge Distillation, Adapter Fine-tuning. Our model obtains competitive results in the end.

1 Introduction

This paper introduces our submission to the WMT 2021 Large-Scale Multilingual Translation Task. We participate in Small Track #2, including 6 languages: Javanese (Jv), Indonesian (Id), Malay (Ms), Tagalog (Tl), Tamil (Ta) and English (En) with 30 directions. We consider that the officially provided dataset has the acceptable size and quality and therefore only participate in the constrained evaluation. Our method is mainly based on previous works but with fine-grained data cleaning techniques and a multi-step multilingual training strategy.

For each language pair, we perform multi-step data cleaning on the provided dataset and only keep a high-quality subset for training. At the same time, several training strategies are tested in a pipeline, including Backward (Edunov et al., 2018) and Forward (Wu et al., 2019a) Translation, Multilingual Translation (Johnson et al., 2017), Iterative Joint Training (Zhang et al., 2018), Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019), Adapter Fine-Tuning (Bapna et al., 2019), and Ensemble (Garmash and Monz, 2016).

Based on the task requirements, we train a single multilingual model that translates all 30 directions. We refer to (Johnson et al., 2017) and employ language tags (Wu et al., 2021). By combining multiple strategies, our model achieves considerable quality improvements in all directions.

Section 2 focuses on our data processing strategies while section 3 describes our training techniques, including model architecture and the iterative training strategy, etc. Section 4 explains our experiment settings and training processes and section 5 presents our experiment results.

2 Data

2.1 Data Source

For all language pairs, we follow the constrained data requirements and take full advantage of the bilingual and monolingual training data available. Table 1 lists the data sizes of each language pair before and after filtering.

2.2 Data Pre-processing

We conduct the following steps to pre-process the data:

- Filter out repeated sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Convert XML escape characters.
- Normalize punctuations using Moses (Koehn et al., 2007).
- Delete html tags, non-UTF-8 characters, unicode characters and invisible characters.
- Filter out sentences with mismatched parentheses and quotation marks; sentences of

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 456–463
November 10–11, 2021.

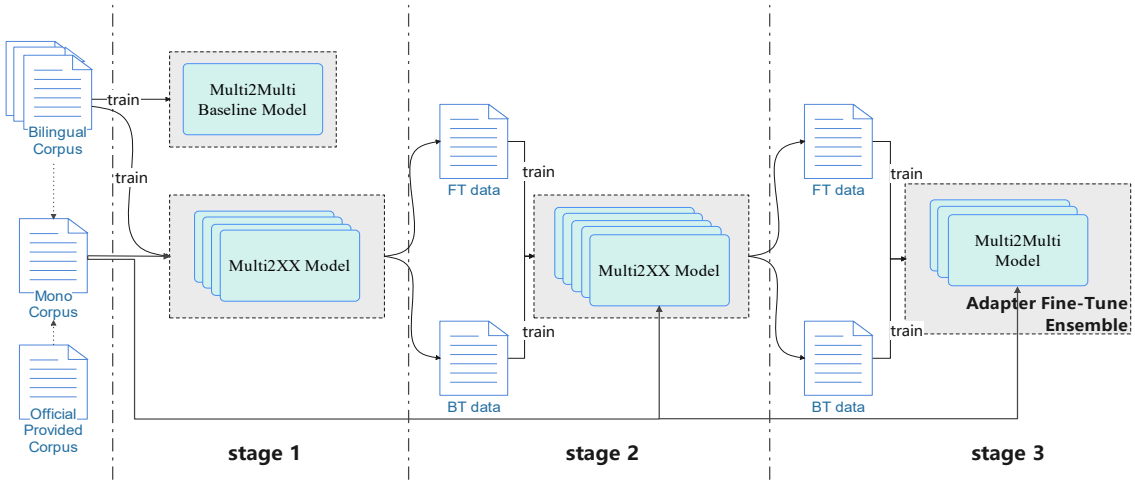


Figure 1: This figure shows the training process for the WMT 2021 Large-Scale Multilingual Translation Task, which consists of three stages. In stage 1, one Multi→Multi model as baseline and five Multi→XX models are trained. In stage 2, the synthetic data by forward and sampling back translation (FTST) is used to train the second round Multi→XX models. In stage 3, second round synthetic FTST data is used to train three Multi→Multi models. Finally, adapter fine-tune and model ensemble are used to enhance the performance.

which punctuation percentage exceeds 0.3; sentences with the character-to-word ratio greater than 12 or less than 1.5; sentences of which the source-to-target token ratio higher than 3 or lowers than 0.3; and sentences with more than 120 tokens. Based on our experience in the industry, this strategy can reduce the low-level errors in model inference and the problem of missing translations.

- Apply langid (Joulin et al., 2016b,a) to filter sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter sentence pairs with poor alignment.
- Use LaBSE (Feng et al., 2020) to rank and filter the monolingual data.

Data sizes before and after cleaning are listed in Table 1.

2.3 Data Selection

According to (Arivazhagan et al., 2019), high-resource language pairs may squeeze the living space of low-resource language pairs. In other words, different data sizes across languages may lead to uneven translation quality in a multilingual model. Since we incorporate all 30 directions in one multilingual model, this issues should be addressed. We use temperature sampling strategy (Zoph et al., 2016) with T=5 to over-sample the low-resource language pairs.

We train all 30 directions under the constrained condition. To improve the performance of back-translation, we combine officially provided monolingual data with the monolingual data extracted from corresponding bilingual corpora. Data sizes are listed in Table 1. The detailed bilingual data size after forward translation and sampling back translation (FTST) and over-sampling are listed in Table 3.

3 System Overview

3.1 Model

Transformer (Vaswani et al., 2017) has been widely used for machine translation in recent years, which has achieved good performance even with the most primitive architecture without much modifications. Therefore, we choose to start from Transformer-Deep (Sun et al., 2019) and consider it as a baseline. The detailed model parameters are as follow: 35-layer encoder, 3-layer decoder, 512 hidden units and a batch size of 4096. We used the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, and the same warmup and decay strategy for learning rate as (Vaswani et al., 2017), with 4,000 warmup steps. During training, we employ label smoothing with a value of 0.1 (Szegedy et al., 2016). For evaluation, we use beam search with

Language pairs	Raw bi data	Filtered bi data	Mono data
En/Id	54M	16.5M	En: 80M
En/Jv	3M	2.2M	
En/Ms	13.4M	12.1M	
En/Ta	2.1M	1.9M	
En/Tl	13.6M	8.7M	
Id/Jv	0.78M	0.51M	Id: 58M
Id/Ms	4.8M	4.3M	
Id/Ta	0.5M	0.4M	
Id/Tl	2.7M	1.6M	
Jv/Ms	0.43M	0.26M	Jv: 3.8M
Jv/Ta	0.06M	0.037M	
Jv/Tl	0.8M	0.32M	
Ms/Ta	0.37M	0.32M	Ms: 19.7M
Ms/Tl	1.3M	0.8M	Tl: 12.2M
Ta/Tl	0.5M	0.3M	Ta: 5M

Table 1: Bilingual data sizes before and after filtering, and monolingual data used in the task. The monolingual data includes officially provided monolingual data and the mono data extracted from the bilingual corpus of corresponding languages.

a beam size of 4 and length penalty $\alpha = 0.6$ (Wu et al., 2016).

3.2 Data Augmentation

Back-translation (Edunov et al., 2018) is an effective way to enhance translation quality by using monolingual sentences to generate synthetic training parallel data. As described in (Wu et al., 2019b), similar to back translation, the monolingual corpus in source language can also be used to generate forward translation text with a trained MT model, and the generated forward and backward translation data can both be merged with the authentic bilingual data. This strategy can increase the data size to a large extent.

We take full advantage of the officially provided monolingual data for data augmentation. In terms of back translation, we adopt top-k sampling for high-resource languages, and adopt beam search for low-resource languages. With regard to forward translation, we translate monolingual data using beam search. Through sampling, we ensure that the sizes of data generated by forward and back translation are relatively equal. In this paper, we refer to the combination of forward and sampling back translation as FTST.

3.3 Multilingual Strategy

Johnson et al. (2017) propose a simple solution to use a single neural machine translation model to translate among multiple languages, and the model requires no change to the model architecture. Instead, the model introduces an artificial token at the beginning of the input sentence to specify the required target language. According to (Wu et al., 2021), we add "2XX" (XX indicates the target language, e.g. 2id) at the beginning of the source sentence. All languages use a shared vocabulary. We train the hybrid SentencePiece model (Kudo and Richardson, 2018) in conjunction with all 6 languages as the shared word segmentation system for all language pairs. We keep the vocabulary within 40k, including tokens of all 6 languages (En/Id/Jv/Ms/Ta/Tl).

Two mainstream methods about multilingual training are available: two models with XX→Multi and Multi→XX separately and a mono Multi→Multi model. According to (Johnson et al., 2017), Multi→XX performs better than Multi→Multi and XX→Multi in general. Multi→Multi model contains too many languages pairs (30 in this case), so conflicts and confusions may occur among language pairs in different directions. However, due to the requirements of the task, we need to provide a Multi→Multi model that includes all 30 directions. In our experiment, we divide 30 language pairs into five Multi→XX multilingual models as step 1. Than we use five Multi→XX multilingual models to conduct back-translation and train a Multi→Multi model as step 2 and step 3, as shown in Figure 1.

3.4 Iterative Joint Training

Zhang et al. (2018) propose a new iterative joint training method, that is, using monolingual data from both source and target sides to train a source-to-target (forward) model and a target-to-source (backward) model at the same time. The two models generate synthetic data for each other. The advantage of such method is that both of the two models gain improvement after each iteration with the synthetic data provided by the other, and then can generate synthetic data with higher quality. Such training procedure is repeated after the two models converge.

3.5 Language independence Adapter Fine-tuning

Previous works demonstrate that fine-tuning a model with in-domain data could effectively improve the model performance. However, due to limitations of a multilingual translation model, once the model is trained, when fine-tuning one of the language pairs, the performance of others will go worse. Thanks to the finding of Adapter (Bapna et al., 2019), we are able to fine-tune each language pair without impacting the performance of others. In the experiment, we set the adapter size to 512 and fine-tune the model on the bilingual data for each language pair in 30 directions with 3,000 tokens per batch for one epoch.

3.6 Ensemble Knowledge Distillation (EKD)

Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019) improves the performance of a student model by distilling knowledge from a group of trained teacher models. Comparing with some soft label distillation methods, the EKD for NMT is relatively straightforward, which can be implemented by training the student models on the combination of the original training set and the translation from the ensembled teacher model on the training set. In our experiments, we ensemble models as the teacher model to translate the FLORES dev set, and use the translation results to further fine-tune models.

3.7 Ensemble

Model ensemble is a widely used technique in previous WMT workshops (Garmash and Monz, 2016), which can improve the performance by combining the predictions of several models at each decoding step. In our work, we ensemble models with different architectures to further improve system performances.

System	FLORES dev	FLORES devtest
baseline M2M	26.9	26.8
FTST1	28.2 (+1.3)	28.1 (+1.3)
FTST2	29.4 (+1.2)	29.6 (+1.5)
Adapter Fine-Tune	30.2 (+0.8)	30.1 (+0.5)
ensemble	30.7 (+0.5)	30.9 (+0.8)
wmt21 final submit	28.6	28.3

Table 2: The experimental results on FLORES dev/devtest, BLEU scores in table are the average of 30 directions.

4 Experiment Settings

4.1 Settings

We use the open-source fairseq (Ott et al., 2019) for training and SentencePieceBLEU to measure system performances. Each model is trained using 8 GPUs. The architectures and main parameters we used are described in section 3.1. Marian (Junczys-Dowmunt et al., 2018) is used for decoding during inference.

4.2 Training Process

We employ iterative training and phase-based data augmentation. Figure 1 shows our training process in details. The specific steps are as follows:

1) Process data using methods described in section 2.2. Train one Multi→Multi model as baseline and five Multi→XX models as forward models and backward models.

2) Generate back translation and forward translation data. Mix the data with parallel training data and train second round five Multi→XX models.

3) Generate back translation and forward translation data using models trained in step 2. Mix data with bilingual training data and train three Multi→Multi models.

4) Average the last eight checkpoints of each model and adapter fine-tune it with bilingual data. Ensemble models to produce the final system.

5 Results and analysis

We use methods described in Section 2.2 for data processing. Model architecture mentioned in Section 3.1 is employed to increase system diversity. On the basis of Multi→Multi baselines model, we use FTST data augmentation to further enhance model performance.

Table 2 lists the results of our experiment on FLORES dev set and devtest set (Goyal et al., 2021). Comparing with the baseline model, the first round FTST Multi→XX models leads to 1.3 BLEU increase on average for the 30 directions. Further, the second round FTST achieves 1.2 BLEU increase on average. We fine-tune the model using bilingual data with adapter and achieve 0.8 BLEU increase on average. Finally, ensemble further leads to 0.5 BLEU increase. When submitting the final results, because of time limits, we only finish round-two FTST. As for model inference, there is a problem with our fairseq architecture, resulting in poor model quality that seriously affects the FTST results. The final model we submitted achieves 28.64 BLEU on FLORES dev and 28.34 BLEU on FLORES devtest. After the submission, we fixed the problem and continued our experiments, eventually achieving 30.7 BLEU on on FLORES dev and 30.9 BLEU on FLORES devtest. The detailed experiment results are listed in Table 4.

In our experiment, due to the inference problem mentioned above, we have not seen much performance improvements. The low quality of model inference leads to poor FT results, which made no contributions to the model. And even worse, it offsets the gain brought by BT results to the model. We also found that the Multi→en model does surpass the Multi→Multi model in quality, which is the same as the results observed by the industry.

6 Conclusion

This paper presents the submissions of HW-TSC to the WMT 2021 Large-Scale Multilingual Translation Task. We perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated. We finally achieve competitive results.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

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.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield,

Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

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.

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.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019a. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019b. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. *arXiv preprint arXiv:2106.07930*.

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*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

A Details of Data size and BLEU

Language pairs	Bilingual data	Bi + FTST data	Over-Sampling T=5
En/Id	46M	56M	56M
En/Jv	2.2M	5M	34M
En/Ms	12M	22M	47M
En/Ta	1.9M	12M	41M
En/Tl	8.7M	18.7M	45M
Id/Jv	0.5M	8.1M	38M
Id/Ms	4.3M	24M	47M
Id/Ta	0.4M	10.6M	40M
Id/Tl	1.6M	21.6M	46.8M
Jv/Ms	0.2M	7.8M	38M
Jv/Ta	0.03M	8.9M	39M
Jv/Tl	0.3M	7.9M	38M
Ms/Ta	0.3M	10.5M	40.4M
Ms/Tl	0.8M	20M	46M
Ta/Tl	0.3M	10M	40M

Table 3: Bilingual data sizes before and after FTST, and Bilingual data sizes after over sampling.

Language Pair	Baseline	Final Submit	FTST1	FTST2	Adapter Fine-Tune	Ensemble
En2Id	46.6	49.2	49.4	49.4	49.5	49.8
Id2En	42.8	43.5	43.7	44.1	44.3	44.8
En2Jv	24.6	26.5	26.3	27.1	27.4	28.3
Jv2En	30.6	31.1	30.9	32.4	32.4	33.3
En2Ms	44.4	45.1	44.8	46.7	46.8	47.8
Ms2En	43.6	42.9	42.7	44.3	44.5	46
En2Ta	24.8	24.8	24.6	26.5	26.6	27.2
Ta2En	24.6	24.6	24.3	26.3	26.4	27.3
En2Tl	33.5	35.3	35.2	36.7	37.9	38.6
Tl2En	41.7	42.1	40.9	43.7	43.9	44.7
Id2Jv	19.5	21.3	20.9	23.5	23.8	24.7
Jv2Id	25.9	28.8	28.6	28.9	28.9	29.6
Id2Ms	35.2	36.9	36.7	38	38.8	39.6
Ms2Id	34	38.2	37.8	38.7	38.8	39.7
Id2Ta	20	21.2	20.9	22	22.7	23.2
Ta2Id	17.1	18.8	18.9	19.1	20.3	21
Id2Tl	26.6	28.7	28.4	29.7	30	30.8
Tl2Id	31	33.7	33.9	35	36.2	36.8
Jv2Ms	24.8	26.8	26.9	28.9	29.8	30.2
Ms2Jv	19.7	21.2	21.4	22.5	23.4	23.8
Jv2Ta	13	14.3	13.9	15	16.2	17.5
Ta2Jv	8.6	9.8	10	11.2	12.8	13.1
Jv2Tl	17.6	20.5	19.9	22.3	22.7	23.4
Tl2Jv	15.5	17	17.2	19.4	19.9	20.2
Ms2Ta	20.2	22.1	20.5	24.1	24.3	24.9
Ta2Ms	19.3	20.4	20.5	22.3	22.5	23.2
Ms2Tl	27.7	28	27.6	30.2	30.6	31.5
Tl2Ms	31	32.5	32	33.7	34.1	34.8
Ta2Tl	18.4	20.8	20.1	21.9	23.1	23.7
Tl2Ta	21.9	23.1	23.2	24.9	26.2	27.1
Average	26.8	28.3	28.1	29.6	30.1	30.9

Table 4: BLEU for each direction on FLORES devtest

On the Stability of System Rankings at WMT

Rebecca Knowles
National Research Council Canada
Rebecca.Knowles@nrc-cnrc.gc.ca

Abstract

The current approach to collecting human judgments of machine translation quality for the news translation task at WMT – segment rating with document context – is the most recent in a sequence of changes to WMT human annotation protocol. As these annotation protocols have changed over time, they have drifted away from some of the initial statistical assumptions underpinning them, with consequences that call the validity of WMT news task system rankings into question. In simulations based on real data, we show that the rankings can be influenced by the presence of outliers (high- or low-quality systems), resulting in different system rankings and clusterings. We also examine questions of annotation task composition and how ease or difficulty of translating different documents may influence system rankings. We provide discussion of ways to analyze these issues when considering future changes to annotation protocols.

1 Introduction

At the WMT (now Conference on Machine Translation) shared task on news translation, research groups build machine translation systems to accurately translate news data, as tested on test sets of recent news documents. The systems are clustered and ranked on their performance as judged by human annotators. The way that human judgments of translation quality have been collected has varied over the course of WMT's history.

In this work, we examine how changes in the collection of human judgments over the last three years have resulted in rankings that are now less robust to the effects of outliers (high- or low-performing systems) and overall annotation task composition. We replicate the human judgment rankings from 2018-2020, perform simulations for reranking, and examine issues of annotation task composition and translation difficulty. We find that sampling sentences for annotators to annotate by document – intended as a step towards evaluating sentences in context – reintroduces a known problem from the earlier era of relative rankings, namely that systems suffer or benefit in their rankings based on the quality of the other data being rated alongside them in the same annotation tasks.

We begin with a discussion of the progression of direct assessment (DA) styles employed in WMT evaluations (§2) and how scoring is performed (§3), before delving into theoretical and practical understandings of the z-scores used to rank systems (§4 and §5), including simulations and analysis of specific examples. We also discuss issues around document distribution and translation difficulty (§6), and close with considerations for downstream impacts (§7) and future study (§8).

2 Historical Context

In 2016, WMT added direct assessment (DA) scoring of system outputs as an investigatory ranking, with relative ranking (RR) remaining the official scoring mechanism (Bojar et al., 2016). In relative ranking, five system outputs for a given segment were ranked in comparison to one another, from which pairwise translation comparisons were generated; these were then used to produce overall system rankings by means of the TrueSkill algorithm (Herbrich et al., 2007; Sakaguchi et al., 2014). Relative ranking can be used to compare systems, but does not provide an absolute score, thus obscuring how close a good system is to a "perfect" translation or, at the other extreme, how poor a system is as compared to others.

The following year, 2017, WMT adopted DA as its main assessment format on the basis of high Pearson correlations between RR and DA in the previous year's investigations (Bojar et al., 2017). In DA (Graham et al., 2013, 2014, 2016), annotators provide an absolute numerical score (0-100) for MT output adequacy (at the sentence level or at the document level) using a sliding scale.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 464–477

The use of DA has changed since it was first introduced to WMT. In 2016, it was trialed for monolingual evaluations of translation fluency and monolingual evaluations of adequacy. Here we provide an overview of changes from the 2016 task to the present, based on Findings papers descriptions.

Bojar et al. (2016) noted that the 2016 version of DA assessments has the potential to avoid a known bias of the RR setup. In RR, each rating task consisted of ranking the outputs of five systems on the same input segment, and "a system may suffer more losses if often compared to the reference, and similarly it may benefit from being compared to a poor competitor" (Bojar et al., 2011). In the 2016 DA setup, translations were annotated in sets of 100, including quality assurance tasks, but each segment was annotated individually, rather than in direct comparison to other system output for the same segment.[1] Quality assurance tasks can include *references* (which should score highly), *"bad" references* (which should score poorly; these are produced by randomly replacing substrings in references to degrade quality), and *repeat assessments* of a segment (which should be scored consistently).

In 2017, DA was adopted as the main annotation style, with exact duplicate segment translations being able to be scored just once (rather than once per system that produced them) and with human assessment scores "standardized according to each individual human assessor's overall mean and standard deviation score" (Bojar et al., 2017).

Bojar et al. (2018) describes two setups for the 2018 DA tasks, a standard structure (with repeat pairs, "bad" references, and references, as quality assurance) and an alternate setup where an additional constraint was imposed, such that within each 100-translation task, for each input the task would include the corresponding output of *all* MT systems. This makes a tradeoff between the aim of DA (to make absolute score judgments rather than relative ones) and getting a single annotator to provide scores for all systems' output of the same source input (which risks reintroducing some form of relative judgement to the task). This is also the first year that the findings paper explicitly spells out the goal of the way tasks (referred to here using the Amazon Mechanical Turk nomenclature "Human Intelligence Task" or HIT) are built in the standard 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 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.[2]

The 2018 shared task also introduced source-based DA, trialling a bilingual version of the task. Rather than scoring MT output against a reference, this version scores it against the source segment, which allows human references to be scored as a "human system" rather than solely as a QA task. They raise a number of potential cautions against drawing strong conclusions, namely that bilingual DA is not yet validated, the alternate task structure may introduce biases, the year's sample size for source-based DA was smaller than 1,500 judgments per system, and that there may be quality issues with some reference segments.

In 2019, WMT introduced additional versions of DA (Barrault et al., 2019). They used monolingual (reference-based) assessment for translation into English and for language pairs that did not include English at all. For translation out of English, they performed bilingual (source-based) DA. The style of DA used in previous years is renamed to SR–DC (Segment Rating without Document Context), as a new style, SR+DC (Segment Rating *with* Document Context) is introduced. In the new SR+DC style, the full translation of a single docu-

[1] It is still possible that there may be biases based on the segments observed in any given set of 100.

[2] Here we reproduce this quote from Barrault et al. (2019), though it appears consistent 2018-2020.

ment by a single MT system is shown to the annotator in order (but still scored segment-by-segment);[3] a task consists of multiple such documents. The generation of annotation tasks is described as follows: all documents translated by all systems are pooled, then sampled (without replacement) until up to 70 segments are selected, at which point quality control documents are added, and finally the order of documents in the task is shuffled. Barrault et al. (2020) uses both SR–DC and SR+DC styles.

3 Scoring

In order to experiment with questions surrounding human evaluation, it is necessary to understand and be able to replicate the official scores produced by WMT. For the human annotations of interest (segment-level evaluation, with or without document context), there are two main types of scores: raw scores and z-scores, with the latter used as the official ranking. These are presented in a table, ordered by z-score, and clusters of systems deemed statistically significantly different (according to a Wilcoxon rank-sum test $p < 0.05$) are separated by horizontal lines.[4]

Following the approach used at WMT, after removing any HITs deemed unacceptable due to quality issues, we calculate raw and z-scores for systems as follows. First, any worker ID whose scores have a standard deviation of 0 is removed. Given a raw score x generated by the worker with worker ID W, its corresponding z-score z is computed as

$$z = \frac{x - \text{mean}(y \in W)}{\text{std}(y \in W)} \quad (1)$$

where $\text{mean}(y \in W)$ is the mean of *all* raw scores generated by worker W, and $\text{std}(y \in W)$ is the standard deviation of *all* raw scores generated by that worker. When we say that the mean and standard deviation are computed from *all* raw scores from a given worker ID, this includes references (which are treated as systems in SR+DC but are treated as quality assurance in SR–DC), "bad references" (which are only ever used for quality assurance), and repeats.[5] However, after computing the mean and standard deviation, only a subset of scores are used to actually compute system averages: those with type "SYSTEM" or "REPEAT" (discarding "BAD_REF" and "REF" types).[6] To compute averages (raw or z-score), first an average is computed for any "SYSTEM" or "REPEAT" scores that share the same system ID, the same document ID, *and* the same sentence ID; that is, if a given sentence of a given document was annotated multiple times for a particular system, we first average those scores (so that more frequently annotated sentences do not receive more weight). Then, for each system, all of its "SYSTEM" or "REPEAT" type scores are averaged, resulting in a system-level score.

We note that the 2019 and 2020 document context (SR+DC) evaluations differ in their quality assurance (see Table 1). In both 2019 and 2020, references are treated as a "Human" system, to be ranked alongside the other systems; which may explain the lack of "REF" labeled segment types in the data. In 2019, the Appraise interface data used to generate the rankings did not include any segments labeled as "REPEAT", "REF", or "BAD_REF", though these are described as being included in the HITs (Barrault et al., 2019); perhaps they were removed before processing the data. In 2020, the Appraise data *did* include segments labeled as "BAD_REF", but none labeled as "REPEAT" or as "REF", while the 2020 Mechanical Turk document-level ones included all three. The 2019 data collected using the Turkle platform contains no human or reference data and we do not use it for any of our analysis in this work.

We reimplemented the scoring system using python and plan to release code for this paper. We were able to exactly replicate the raw scores and z-scores for most of the language pairs of interest from 2018-2020,[7] as well as the significance clusters.[8] See Appendix A for details. We use this reimplementation of the WMT scoring scripts in or-

[3] There is also a Document Rating with Document Context DR+DC, but we do not examine that in this work.

[4] A horizontal line is drawn below a system if and only if it is significantly better ($p < 0.05$) than *every* system with a lower z-score than it.

[5] We compute mean and standard deviation using `ad-latest.csv`, but use `ad-good-raw-redup.csv` to compute the individual z-scores and averages. The files are downloaded from 2018-2020 WMT websites: `http://www.statmt.org/wmt18/results.html`, `http://www.statmt.org/wmt19/results.html`, and `http://www.statmt.org/wmt20/results.html`.

[6] "SYSTEM" type are system outputs, while the remainder are quality assurance: "REPEAT" are repeated system outputs which are also valid for computing averages, "BAD_REF" are degraded references, and "REF" are references.

[7] In order to match the z-scores generated by the R packages used for WMT, we set `ddof` equal to 1 when using the `stats.zscore` function from scipy.

[8] We replicated the significance clusters using scipy's `stats.mannwhitneyu` function.

Dataset	SYSTEM	REPEAT	REF	BAD_REF
newstest2018-humaneval	265387	26489	26003	36924
appraise-doclevel-humaneval-newstest2019	194625	0	0	0
mturk-sntlevel-humaneval-newstest2019	92164	13266	13177	13113
turkle-sntlevel-humaneval-newstest2019	47799	0	0	0
appraise-doclevel-humaneval-newstest2020	186663	0	0	26856
mturk-sntlevel-humaneval-newstest2020	26262	3741	3746	3773
mturk-doclevel-humaneval-newstest2020	93777	12887	12939	12965

Table 1: Counts of sentence types in ad-good-raw-redup.csv files from 2018-2020. We omit the Turkle data from most of our analysis because it contains neither human systems nor reference data.

der to score authentic and modified WMT data, to examine underlying assumptions and hypothesize about how these may impact final system rankings.

For 21 language pairs annotated in SR–DC style and 25 in SR+DC style from 2018-2020, we were able to exactly replicate rankings, nearly replicate rankings (e.g., with rounding difference related changes to one significance line), or produce rankings whose differences could be explained by delays in data collection (2020 en-iu).[9] Appendix A provides more details on replication. We use our recalculated rankings and clusters as the starting point for all remaining analysis in this paper.

4 Understanding z-scores

While we've described how the z-score is calculated in the setting of the WMT human annotations, it's important to take a closer look at z-scores to understand how they behave in different scenarios. In this section, we explore z-scores and their underlying assumptions in hypothetical scenarios.

Given a raw score x, a mean μ, and a standard deviation σ, the **z-score** (or standard score) is the number of standard deviations above or below the mean that x falls. The z-score for a given raw score x can be computed as follows:

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

This is a linear transformation; the shape of the distribution of z-scores is the same as that of the raw scores, but now with a mean of 0 and a standard deviation of 1. It is a unitless score.

Intuitively, the z-score provides a potential way of comparing scores from different annotators, but it requires a careful examination of underlying assumptions. If we think of the z-score as a unitless score, perhaps we can think of each annotator as having their own measurement units: we might have a lenient annotator and a harsh annotator, such that a raw score of 50 by the lenient annotator is quite *bad* while a raw score of 50 for the harsh annotator is actually quite *good*. In order to directly compare the two annotators' scores, we would like to map them to a shared scale, a unitless z-score. Under what assumptions is it appropriate to calculate z-scores to compare annotators' scores?

We start with perhaps the most obvious (but frequently unstated) assumption: there exists some latent "quality" of a given translation, which can be judged by a human annotator, such that annotators roughly agree about what constitutes a "good" or a "bad" translation. In practice, human annotators may disagree – for any number of reasons (Basile et al., 2021) – about which of two translations of "similar quality" is better, but we assume that the disagreement is not *extreme*; i.e., we hope that under a correlation coefficient like Pearson's r or Spearman's ρ, the correlation between annotators' scores would be much closer to 1 than to -1. For the sake of simplicity in the following examples, we will assume there exists a "true" and "objective" score for every translation.

Suppose that we have some translations with a true mean score of μ and a true standard deviation of σ. A lenient annotator scores all of the translations such that the distribution of their scores has a mean of $\mu + n$ and a standard deviation of σ, while a harsh annotator scores all of the translations such that the distribution of *their* scores has a mean of $\mu - m$ and a standard deviation of σ.[10] When we compute their z-scores, it is easier to directly compare sentence scores, since they are now on the same scale. This seems like a reasonable use of z-scores, but in this scenario annotators are scoring exactly the same data, which doesn't scale to WMT-style annotations; annotators simply don't

[9] Language codes: Chinese (zh), Czech (cs), German (de), English (en), Estonian (et), Finnish (fi), Gujarati (gu), Inuktitut (iu), Japanese (ja), Kazakh (kk), Khmer (km), Lithuanian (lt), Pashto (ps), Polish (pl), Russian (ru), Tamil (ta), Turkish (tr).

[10] We use the same standard deviation for simplicity, with arbitrary positive values of n and m.

have time to score all of the data.

Now suppose that we have two disjoint sets of sentences scored by two different annotators: the set S_X of sentences scored by annotator X and the set S_Y of sentences scored by annotator Y. From these raw scores, we can compute μ_X and μ_Y along with σ_X and σ_Y. If $\mu_X > \mu_Y$, can we conclude that annotator X is a more lenient annotator than annotator Y and resolve this by computing z-scores? Not without additional information! Imagine that we could see the "true" mean scores of S_X and S_Y, as annotated by a perfect omniscient annotator. It could be the case that the true means are identical and annotator X is indeed more lenient, but it could also be the case that the true mean of the scores in set S_X is actually higher. In the latter case, the annotators could be equally lenient, or it is even possible that annotator Y could be more lenient! In short, without a shared basis for comparison, we don't know whether computing z-scores is normalizing out annotator differences, differences in the data itself, or a combination.

5 z-scores in practice

This raises the question: what is happening in practice when we compute z-scores on WMT DAs? Are we really normalizing away inter-annotator differences, or is the normalization also doing something else, such as normalizing away real differences in HIT and system quality? If it is the latter, even z-scores for DAs may suffer the same bias from comparisons to better (or worse) systems.

We don't have access to an oracle, and we don't have a direct or reliable way to compute interannotator agreement, because in some collections it is rare that two annotators annotate the same text (and for the Appraise data, we only have HIT information, not annotator information). However, we can still examine this in the existing data and modifications thereof. Bojar et al. (2011) noted that systems might suffer from being compared to the reference too frequently under relative ranking, or might benefit from being compared to particularly poor systems. The same could hold true in DA. Consider the following toy example: a HIT contains 4 sentences, with raw scores of 25, 50, 50, 75, respectively. A sentence with a raw score of 50 in this HIT would have a z-score of 0. If, instead, the raw scores were 0, 25, 50, 75, a sentence with a raw score of 50 would have a z-score of 0.39, while for a HIT with raw scores of 25, 50, 75, 100, a sentence with a raw score of 50 would have a z-score of -0.39. While it is possible that such a set of scores could reflect differences in *annotator* behavior, we could also easily imagine that they might reflect differences in *HIT composition*, with one containing only system scores, one containing system scores and a bad reference, and one containing system scores and a (good) reference.

5.1 HIT Composition

Thus we examine HIT composition, or, more accurately, the composition of data annotated by any given worker/worker ID. In 2018, all systems were SR–DC, and 100% of workers annotated "BAD_REF" data.[11] However, an "Alternate DA HIT Structure" was employed for a subset of researcher HITs (run in Appraise), which used only "BAD_REF" segments for quality assurance, "omitting repeat pairs and good reference pairs" while also attempting to include "the output of all participating systems for each source input" (to have the same annotator produce annotations across systems). The percentage of (non-rejected) workers who annotated data containing "REF" in 2018 ranged from 4.9% (en-et) to 98.8% (zh-en); the former is an outlier, as the next two lowest are 25.8% (en-cs) and 47.4% (en-fi).

In 2019 annotations into English, 100% of workers annotated both "REF" and "BAD_REF" segments. In 2019 annotations out of English, the final output data does not include any "REF" or "BAD_REF" segments (though these *are* described as having been included for QA), but human references are treated as systems, and between 37.8% (en-de) and 61.5% (en-kk) of workers annotated at least some human reference data.

The 2020 Appraise annotations differed from prior years as well: 100% of the 2020 into English (Mechanical Turk) workers annotated both "REF" and "BAD_REF" segments. In 2020 annotations out of English (Appraise), between 95.8% (en-iu) and 100% (en-{ja, ta, zh}) of workers[12] annotated "BAD_REF" data. The percentage of Appraise "workers" that annotated data containing human references (treated as a system) ranged from 8.3% (en-iu) to 73.4% (en-zh).

[11] These values are calculated on ad-good-raw-redup.csv files, so only include annotators who successfully passed QA.

[12] The definition of "worker" is really a bit fuzzy here; the "WorkerID" produced by Appraise is really a HIT ID, so averages are *not* necessarily being computed across all of a given worker's annotations, but rather each HIT is being treated as a unique worker.

5.2 Analysis

In an ideal world where z-score normalization is only correcting for annotator variation, removing one system should not result in changes to the relative rankings of the remaining systems. That is to say, the z-scores themselves may be expected to change (shifting up if a very good system is removed, shifting down if a low-quality system is removed), but we wouldn't expect the relative ranking of two systems to change. After all, one stated motivation of the shift to DA was to avoid the known bias in RR of systems being unfairly penalized or benefiting unfairly from comparisons to stronger/weaker systems (Bojar et al., 2016). Similarly, replacing one system – for example with a much better or much worse system – should not result in other systems switching places in the rankings. We simulate these two scenarios using the existing data, and show that rankings produced in SR+DC settings are much more sensitive to removal or modification of systems than SR–DC.

Year/Type	Δ Rank	Δ Cluster	Δ Both
'18 (–DC)	1/13	0/13	0/13
'19 (–DC, MTurk)	2/5	1/5	0/5
'20 (–DC, MT.)	1/3	0/3	0/3
ALL SR–DC	**4/21**	**1/21**	**0/21**
'19 (+DC, MT.)	1/2	1/2	1/2
'19 (+DC, A.)	6/8	3/8	1/8
'20 (+DC, MT.)	7/7	3/7	3/7
'20 (+DC, A.)	4/8	5/8	3/8
ALL SR+DC	**18/25**	**12/25**	**8/25**

Table 2: Effect of removing human and "REF" scores from annotations and recalculating rankings by year, platform (MTurk or Appraise), and annotation style. Values indicate the fraction of language pairs that had changes in rank, clustering, or both rank and clustering.

We first examine removing human systems and "REF" – acting as though they had never been annotated at all, so all z-scores are calculated without "REF" or human system scores.[13] We then compute rankings and significance clusters. We compare these against the original rankings generated from all available data, with the significance clusters recomputed after removal of human systems.[14] For each pair of rankings, we check whether there is any change in the order of systems (ignoring significance clusters; we call this Δ *Rank*), whether there is any change in clusters (different number or composition of clusters; we call this Δ *Cluster*), and/or changes in both (Δ *Both*). Table 2 shows the results. Rank changes (ignoring significance clusters) are the most common, and many of these occur within significance clusters as we would expect. However, there are also a number of changes to the significance clusters (clusters merging, splitting, or rearranging), as well as pairs for which both rank and cluster changes occur. Most strikingly, all of these changes are *much* more common in the SR+DC settings than in the SR–DC. Removing human and "REF" data results in cluster changes to almost *half* (12/25) of the SR+DC rankings, but less than 5% (1/21) of the SR–DC rankings. No SR–DC rankings exhibit changes in both rank and clusters, but 32% of SR+DC rankings do. This is evidence that the SR+DC rankings are less stable, and consequently less reliable, than the SR–DC rankings. We replicate this result with removing the highest and lowest ranked systems, respectively, as shown in Table 3; the SR+DC rankings are much less robust than the SR–DC rankings to the removal of the best or worst single system.

Removed/Type	Δ Rank	Δ Cluster	Δ Both
Lowest (SR–DC)	4/21	3/21	1/21
Lowest (SR+DC)	18/25	10/25	7/25
Highest (SR–DC)	0/21	1/21	0/21
Highest (SR+DC)	17/25	10/25	5/25

Table 3: Effect of removing single lowest ranked or highest ranked system across all years, by data collection type (–/+DC). Values indicate the fraction of language pairs that had changes in rank, clustering, or both rank and clustering.

One might worry that some of this instability is due to the shrinking number of datapoints available when we remove "REF" and human systems, or the highest/lowest ranked systems. To account for this, we run the same experiment and measure the same changes, but instead of *removing* "REF" and human systems, we degrade their raw scores (dividing each score by 1.25, 1.5, 2, 4, and 10) before computing z-scores, rankings, and significance clusters. This could be viewed as a simulation of what would occur if the high-quality human system were replaced with mediocre (or, in the case of division by 10, very low-quality) systems.[15]

We visualize the result in Figure 1. Once again, the SR+DC evaluations are more brittle to these

[13] We observe similar results if we only remove "REF", but in that setting we cannot examine the 2019 and 2020 Appraise SR+DC rankings, as they do not make use of "REF" at all.

[14] Relevant to clusters containing or above human system(s).

[15] The reverse – inflating scores of low-performing systems – has a similar effect, but requires consideration of how to handle scores of zero.

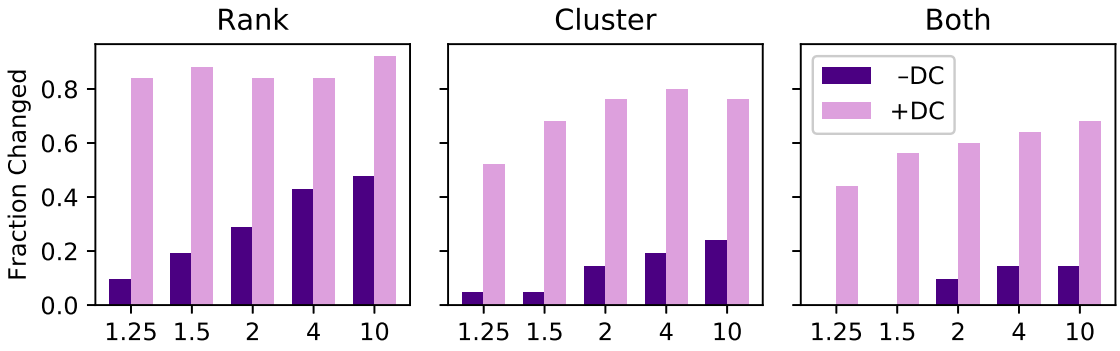


Figure 1: Effect of dividing raw human system and "REF" scores on overall (z-score) rankings for all SR–DC and SR+DC shown in Table 2. The x-axis shows the divisor (ranging from 1.25 to 10) and the y-axis shows the fraction of pairs for which the rankings, clusters, or both ranks and clusters changed.

changes. However, we see that even the SR–DC evaluations are not immune to the effects of extreme outliers on rankings and clusterings – as the divisor used increases, so does the fraction of pairs that have ranking and/or clustering changes. This makes sense intuitively: if most systems are of similar quality, a slight imbalance in which systems are compared to one another likely won't have dramatic effects, but if one system is much worse (or much better) than the rest, systems that are compared against it more or less frequently than others will see their z-scores benefit or suffer accordingly. We also examined monolingual vs. bilingual tasks in the SR+DC context (all SR–DC tasks were monolingual), but note similar rates of changes to ranks, clusters, and both across the two settings.

We have used a very coarse measurement here: counting whether the ranks or clusters changed *at all* rather than whether multiple clusters or large numbers of systems were reranked. Indeed, many of these changes are quite subtle, with just a single new significance line appearing or two clusters merging, or two close systems switching ranks (within or across clusters). If that is the case, why should we be concerned with this? The first reason is to better understand what it is that is actually being measured and whether the WMT annotation protocol is succeeding in its goals. If the inclusion of outliers or the degradation of system scores results in other systems shifting ranks, this indicates that the current approach does suffer from a similar comparison bias to RR. Thus we can't always be confident that what is being measured is a property of the system itself and not closely intertwined with HIT composition – this approach is doing something other than *only* normalizing away interannotator differences. The second reason is to highlight these goals and assumptions so that they can be considered when making future modifications to the annotation process. Many of these issues are currently resulting in small inconsistencies, but if future modifications are made to the annotation process without considering the underlying assumptions and goals, there is no reason to expect that the errors will cancel one another out rather than compound. If we are aware of the underlying assumptions when changes are introduced to the annotation process, we will be better positioned to consider potential problems in the hypothetical and then examine the real data to see if they appear in practice. There is also the question of effects on downstream tasks (§7). Finally, it also helps us to consider ways to mitigate these challenges before they grow, and we discuss some options for future consideration in §8.

5.3 Case Study

We manually select for examination a relatively dramatic case of rankings and clusters changing, from en-de 2020, pictured in Figure 2. This is an unusual case since it contained multiple human-based systems.[16]Nevertheless, it incorporates several issues we raised in hypotheticals, so we discuss it here.

Figure 2 shows the rankings for the original data (human systems were dropped only for the purpose of computing clusters, but *were* used for calculating z-scores), and each of the rankings computed by degrading raw scores by dividing them by 1.25 through 10 (denoted d-n where n is the divisor). We begin by focusing on PROMT_NMT, whose rank increases with increased degradation

[16]See Appendix A for details.

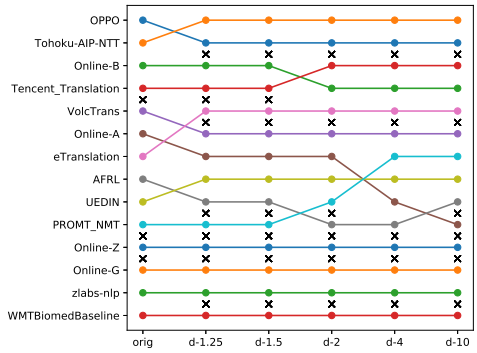


Figure 2: Plot showing z-score rankings (top is best) for 2020 en-de (SR+DC), from original rankings and five divisors for raw human scores. Significance lines are marked with black "x". Human systems were used in calculating z-scores but were removed prior to computing clusters for ease of visualization and comparison.

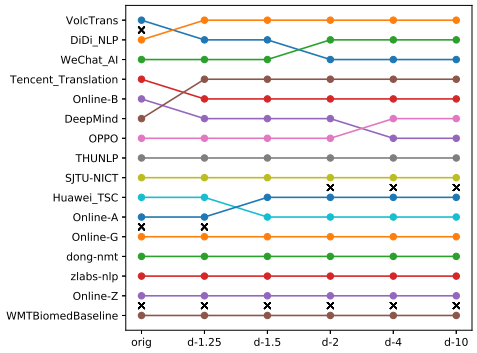


Figure 3: Rankings for 2020 zh-en (SR+DC), from original rankings and with divisors, as in Figure 2.

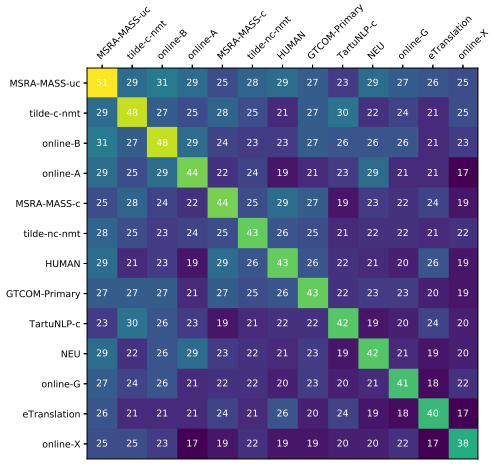


Figure 4: Co-occurrence matrix of systems for en-lt 2019 (SR+DC). Each cell shows the number of HITs that contained segments from the systems at those x and y values. The diagonal shows the total number of HITs that contained each system.

of the human systems. In the original ranking, AFRL and PROMT_NMT appear in the same cluster, with AFRL having a higher score than PROMT_NMT, but not statistically significantly so. When degrading the human raw scores by 1.25 or 1.5, AFRL is in a higher significance cluster than PROMT_NMT, but when dividing by 2, this is reversed: PROMT_NMT is now ranked as significantly *better* than AFRL, while with a divisor of 4 or 10, they return to the same cluster but with PROMT_NMT scoring higher. Thus we see, purely by degrading the raw scores of *other* systems, we observe the full range of possible relative rankings and clusterings for this pair of systems. The same holds true for PROMT_NMT compared with Online-A.

The en-de 2020 rankings may have suffered somewhat from having fewer annotations (1123.6 assessments per system), so we also show results for one of the most-assessed pairs that year: zh-en (2035.1 assessments per system). This is shown in Figure 3.[17] Here we focus on the top system: VolcTrans, which was ranked in Barrault et al. (2020) as significantly better than all systems. As we degrade the human systems, we see it begin to drop in rank, and this significance cluster merges with the one below it, raising the possibility that the initial finding was an artifact of the distribution of data across HITs rather than an inherent property of the MT quality of that particular system.

[17]Note that in the original rankings shown, the human system was omitted when computing significance clusters, and in this case a new significance line (separating Online-A and Online-G appears) where it had been, which was not there in the published rankings that do include human systems.

5.4 System Comparisons

There is a distinct difference in the way that systems are distributed across HITs in the SR–DC and SR+DC annotation styles. In SR–DC, almost all HITs contain segments from every single system (though there is no guarantee that they appear in exactly equal proportions to one another).

In SR+DC, this is not the case, owing to the fact that HITs are limited to 100 segments, there are often 10 or more systems, and documents are often longer than 10 segments. This means that it may be numerically impossible for a given HIT to cover all systems. We see this in Figure 4. A given system may be paired with any other system in less than half of the HITs in which it appears. These kinds

of imbalances mean that systems may be more frequently compared to better or worse systems, resulting in unfair effects on their rankings.

6 Documents

In both SR+DC and SR–DC styles, we don't have a guarantee that every segment-system pair is judged by an annotator, nor that at least one segment from every document-system pair is judged. If we assume approximately uniform translation difficulty across the test set, this isn't necessarily too much of a concern. However, is that really the case, or are some documents "easy" and others "hard"?

Figure 5 shows a matrix of document-system pairs, with each cell showing the average of all of the segment raw scores for that system-document pair.[18] The documents are ranked from highest average raw score to lowest average score (top to bottom), while the systems are ranked by highest average raw score to lowest average raw score (left to right). In the leftmost column, we see the "HUMAN" system, which has high scores across all documents. If all documents were equally difficult to translate, we would expect to see a gradient along the x-axis (i.e., across systems), with minimal variation along the y-axis (i.e., across documents). What we observe instead in this en-lt pair from 2019 (and across a number of other language pairs) is a rough gradient from the top left to the bottom right (with the exception of the "HUMAN" system, which remains strong throughout). This suggests that there are some documents that are "easy" for most systems to translate (top) and some that are "hard" (bottom). This raises a concern: when we attempt to compare two systems of very similar quality, they are not being measured on the same test set. An unlucky sample of documents might see one system judged on a "harder" set of documents, calling the resulting rankings into question.

7 Downstream Consequences

While researchers building MT systems for the shared task may view the human judgment rankings as the end result, the rankings are the *input* to the metrics tasks at WMT. Thus the reliability of the rankings has a direct impact on the reliability of the metrics task – which in the long term feeds into MT research as researchers decide

[18] We can also produce such a matrix using z-scores or automatic metric scores, and results are comparable.

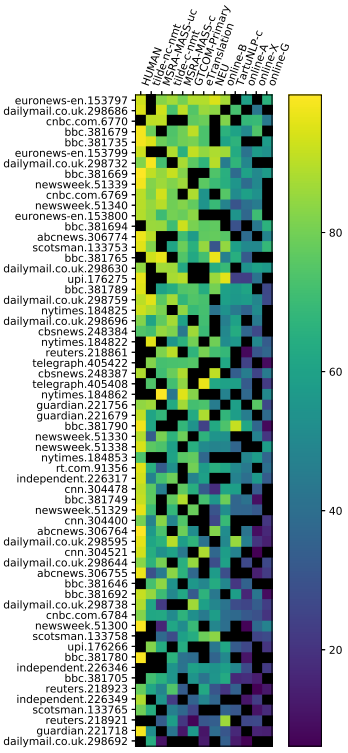


Figure 5: Average raw scores for document-system pairs from en-lt 2019 (SR+DC). Empty cells indicate pair was not judged. Documents are ranked by average raw score (highest: top) as are systems (highest: left).

which automatic metrics to use for evaluating their systems. In system-level metric evaluation at the WMT Metrics shared task, Pearson correlations are computed between metric scores and the z-score human rankings (Mathur et al., 2020b). Note that these correlations are directly between the system average z-scores and the metric scores, and as such do *not* treat all systems within a given cluster as tied. In practice, this means that even rank-only perturbations in the official ranking can be expected to cause changes to metrics task results.

Metrics scores are run on the *full* test set, not the various human-annotated subsets. Citing Graham et al. (2013), the Metrics task papers note that system-level DA scores are "consistent and have been found to be reproducible" even though different sets of segments are assessed for each system. However, that work predates the shift to sampling by document, and our analysis of instability and document difficulty suggest revisiting it.

Recent work has shown that outliers have a concerning impact on metric correlations (Mathur et al., 2020a), and organizers have worked to mitigate this (Mathur et al., 2020b). This paper is a step

towards answering questions raised in Mathur et al. (2020b) regarding outliers and unfair advantages. It may seem tempting to remove outliers from human judgment tasks, but this will not solve the other problems and could instead mask their presence.

8 Proposals for Future Work

The issues discussed in this paper raise concerns about changes to the human evaluation protocols used at WMT and their effects on the validity of WMT system rankings. A partial solution would be to return to SR–DC annotations, perhaps after validation of the 2018 alternate HIT structure that guarantees that for every segment in the HIT, the HIT contains *every* MT system's output for that sentence. But this may be an unsatisfactory conclusion, and fails to address the interest in pushing MT evaluation toward whole documents.

Document-level and context-inclusive evaluations are growing in popularity, but there is limited study on document-level assessment methodologies for MT. Castilho (2021) examines setups comparable to SR–DC, SR+DC, and document rating with document context (which we omitted from this work), and finds in a controlled experiment using Likert scale ratings that a methodology comparable to SR+DC produces higher levels of interannotator agreement and fewer misevaluations than either whole document scores or individual sentences without context. However, that experimental setup does not suffer from the same task composition issues we observe in WMT; in fact these may be orthogonal issues.

If the choice is made to use SR+DC style annotations, there are some improvements to consider, but as noted in Castilho (2020), it remains "essential to test which methodologies will be best suited for different tasks and domains" prior to adopting them. One option would be to create 2018-alternate-structure style HITs with document context, where a HIT contains all systems' output for one or more documents. The downside to this is that it would likely require longer HITs or HITs that only contain a small number of documents; if systems are of similar quality, we might be concerned about annotator fatigue from repetition. The amount of context needed to adequately assess translations is still a question under consideration (Castilho et al., 2020; Castilho, 2021), which ties into issues of document and HIT length.

Another possibility to consider would be to always normalize over annotators (rather than over HITs), but this isn't a solution on its own – it is still necessary to make sure that annotators see comparable distributions of systems and documents, or the same problems will be reintroduced. Having annotators do calibration HITs, i.e., a set of annotations that *all* annotators complete, could also be considered. The calibration HITs would provide a consistent basis for computing the parameters of an annotator-specific z-score transformation, which could then be applied to the remainder of the annotator's judgments. This could untangle the issue of annotator strictness/leniency, but would still merit study before implementation (as annotator behavior may depend on HIT composition, so the z-scores learned in calibration may not be as applicable as one might hope if there is a mismatch between calibration HITs and the remainder of the HITs). One could also consider additional ways of modeling annotator behavior beyond z-score normalization (Paun et al., 2018).

A simpler starting point to deal with the issue of different systems being annotated over different documents would be to guarantee that all systems are scored over the same subset of documents.

All of these are (partially) orthogonal to the questions of what type of annotation tasks result in the most reliable ratings – whether it be direct assessment, ranking, or detailed error annotation – or questions of annotator skills and knowledge (Freitag et al., 2021).

9 Conclusions

We have shown that the current judgment collection methodology at the WMT news translation task results in SR+DC judgments that are more prone to variation on the basis of outliers than SR–DC judgments, and that HIT composition issues have helped reintroduce the relative ranking problem of unfair comparisons to the WMT rankings. We examined issues of document difficulty and how this interacts with the decision to sample documents (rather than sentences) for judgment. These issues risk undermining the validity of WMT rankings, with real consequences for MT research and downstream tasks on automatic metrics. In examining these issues, we've also presented several approaches to diving into the WMT ranking data that may be helpful to consider when planning future changes to WMT human judgment collection procedures.

Acknowledgments

Thank you to the anonymous reviewers for their suggestions and comments. Thank you to Chi-kiu Lo, Nitika Mathur, Gabriel Bernier-Colborne, Roland Kuhn, Huda Khayrallah, Adam Poliak, and George Foster for feedback on various drafts of this work. Thank you to Yvette Graham and task organizers for the 2020 data release and pointers to DA-related code. Thank you also to those listed above and a number of other current and former colleagues – including Rachel Rudinger, Eric Joanis, Darlene Stewart, Samuel Larkin, Michel Simard, Serge Léger, Patrick Littell, and Cyril Goutte – for discussions on related topics.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Valerio Basile, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We need to consider disagreement in evaluation. In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21, Online. 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*, 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, Aurélie Névéol, 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, Berlin, Germany. 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 Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Sheila Castilho. 2020. On the same page? comparing inter-annotator agreement in sentence and document level human machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1150–1159, Online. Association for Computational Linguistics.

Sheila Castilho. 2021. Towards document-level human MT evaluation: On the issues of annotator agreement, effort and misevaluation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 34–45, Online. Association for Computational Linguistics.

Sheila Castilho, Maja Popović, and Andy Way. 2020. On context span needed for machine translation evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3735–3742, Marseille, France. European Language Resources Association.

Markus Freitag, George F. Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *CoRR*, abs/2104.14478.

Yvette Graham, Timothy Baldwin, Meghan Dowling, Maria Eskevich, Teresa Lynn, and Lamia Tounsi. 2016. Is all that glitters in machine translation quality estimation really gold? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3124–3134, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. Trueskill™: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585.

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. Association for Computational Linguistics.

A Notes on Replication

As shown in Table 4, we are able to duplicate the following rankings exactly (or with minor differences, as noted). Code to replicate this work will be available at https://github.com/nrc-cnrc/WMT-Stability/. Language codes are as follows: Chinese (zh), Czech (cs), German (de), English (en), Estonian (et), Finnish (fi), Gujarati (gu), Inuktitut (iu), Japanese (ja), Kazakh (kk), Khmer (km), Lithuanian (lt), Pashto (ps), Polish (pl), Russian (ru), Tamil (ta), Turkish (tr).

- 2018, Mechanical Turk, SR–DC: en-{cs, de, et, fi, ru, tr, zh} and {cs, de, et, fi, ru, zh}-en, but we do not successfully replicate the scores for tr-en (we omit tr-en 2018 from future experiments).
- 2019, Appraise, SR+DC: en-{cs, de, fi, gu, kk, lt, ru, zh}, though we note that en-kk contains a duplicate system that is omitted from the published table.
- 2019, Mechanical Turk, SR–DC: {gu, kk, lt, ru}-en, and fi-en is nearly replicated, but our replication of it is missing a significance line between two clusters due to a rounding difference when computing the significance value.
- 2019, Mechanical Turk, SR+DC: {de, zh}-en are successfully replicated.
- 2019, Turkle, SR–DC: de-cs, de-fr, fr-de, zh-en, are all successfully replicated but are not included in the analyses.
- 2020, Appraise, SR+DC: en-{cs, ja, ru, ta, zh}, are successfully replicated, while en-pl is missing one significance line due to rounding differences. The ranking for en-de has identical scores *except* for Human-A and Human-paraphrase. The original en-de ranking in Barrault et al. (2020) included Human-A, Human-B, and Human-paraphrase. The released en-de data only contained Human-A and Human-B, though Human-A was about twice as large as Human-B, suggesting that it may have incorporated the Human-paraphrase data. Finally, the ranking for en-iu is quite different, though we expect this is because of delays in data collection resulting in a mismatch between the reported scores in the findings paper and the released scores. The en-iu scores also contain an additional low-scoring system that was omitted from the published table.
- 2020, Mechanical Turk, SR+DC: {cs, de, ja, pl, ru, ta, zh}-en were all replicated exactly.
- 2020, Mechanical Turk, SR–DC: {iu, km, ps}-en were all replicated exactly.
- 2020 en-{km, ps} appear to be missing from the released data.

Lang.	Year	–DC	+DC	Mono./Bi.	Tool	Replicated/Notes
en-cs	18	✓		M		✓
en-de	18	✓		M		✓
en-et	18	✓		M		✓
en-fi	18	✓		M		✓
en-ru	18	✓		M		✓
en-tr	18	✓		M		✓
en-zh	18	✓		M		✓
cs-en	18	✓		M		✓
de-en	18	✓		M		✓Matches clusters from Table 15, Appendix A, not Table 8.
et-en	18	✓		M		✓
fi-en	18	✓		M		✓
ru-en	18	✓		M		✓
*tr-en	18	✓		M		*Not successfully replicated.*
zh-en	18	✓		M		✓
en-cs	19		✓	B	Appraise	✓
en-de	19		✓	B	Appraise	✓
en-fi	19		✓	B	Appraise	✓
en-gu	19		✓	B	Appraise	✓
en-kk	19		✓	B	Appraise	✓Contains a duplicate of one system.
en-lt	19		✓	B	Appraise	✓
en-ru	19		✓	B	Appraise	✓
en-zh	19		✓	B	Appraise	✓Matches clusters from Table 33, Appendix A, not Table 11.
de-en	19		✓	M	MTurk	✓
fi-en	19	✓		M	MTurk	Missing a significance line (rounding difference).
gu-en	19	✓		M	MTurk	✓
kk-en	19	✓		M	MTurk	✓
lt-en	19	✓		M	MTurk	✓
ru-en	19	✓		M	MTurk	✓Matches clusters from Table 45, Appendix A, not Table 11.
zh-en	19		✓	M	MTurk	✓Note that this appears in Table 15.
*de-cs	19	✓		M	Turkle	✓
*de-fr	19	✓		M	Turkle	✓
*fr-de	19	✓		M	Turkle	✓
*zh-en	19	✓		M	Turkle	✓This is the Table 11 ranking.
en-cs	20		✓	B	Appraise	✓
en-de	20		✓	B	Appraise	Human-paraphrase missing (subsumed under Human-A?).
en-iu	20		✓	B	Appraise	Different scores/different data? Contains additional system.
en-ja	20		✓	B	Appraise	✓
*en-km	20			?	?	*Does not appear to exist.*
en-pl	20		✓	B	Appraise	Missing a significance line (rounding difference).
*en-ps	20			?	?	*Does not appear to exist.*
en-ru	20		✓	B	Appraise	✓
en-ta	20		✓	B	Appraise	✓
en-zh	20		✓	B	Appraise	✓
cs-en	20		✓	M	MTurk	✓
de-en	20		✓	M	MTurk	✓
iu-en	20	✓		M	MTurk	✓
ja-en	20		✓	M	MTurk	✓
km-en	20	✓		M	MTurk	✓
pl-en	20		✓	M	MTurk	✓
ps-en	20	✓		M	MTurk	✓
ru-en	20		✓	M	MTurk	✓
ta-en	20		✓	M	MTurk	✓
zh-en	20		✓	M	MTurk	✓

Table 4: Notes on Findings paper ranking replications, including information about language pairs, year, SR–DC vs. SR+DC, monolingual vs. bilingual evaluation, tool used for data collection, and success or failure to replicate. Systems marked with * were not included in any additional analysis.

To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation

Tom Kocmi **Christian Federmann** **Roman Grundkiewicz** **Marcin Junczys-Dowmunt** **Hitokazu Matsushita** **Arul Menezes**

Microsoft
1 Microsoft Way
Redmond, WA 98052, USA
{tomkocmi,chrife,rogrundk,marcinjd,himatsus,arulm}@microsoft.com

Abstract

Automatic metrics are commonly used as the exclusive tool for declaring the superiority of one machine translation system's quality over another. The community choice of automatic metric guides research directions and industrial developments by deciding which models are deemed better. Evaluating metrics correlations with sets of human judgements has been limited by the size of these sets. In this paper, we corroborate how reliable metrics are in contrast to human judgements on – to the best of our knowledge – the largest collection of judgements reported in the literature. Arguably, pairwise rankings of two systems are the most common evaluation tasks in research or deployment scenarios. Taking human judgement as a gold standard, we investigate which metrics have the highest accuracy in predicting translation quality rankings for such system pairs. Furthermore, we evaluate the performance of various metrics across different language pairs and domains. Lastly, we show that the sole use of BLEU impeded the development of improved models leading to bad deployment decisions. We release the collection of 2.3 M sentence-level human judgements for 4380 systems for further analysis and replication of our work.

1 Introduction

Automatic evaluation metrics are commonly used as the main tool for comparing the translation quality of a pair of machine translation (MT) systems (Marie et al., 2021). The decision of which of the two systems is better is often done without the help of human quality evaluation which can be expensive and time-consuming. However, as we confirm in this paper, metrics badly approximate human judgement (Mathur et al., 2020b), can be affected by specific phenomena (Zhang and Toral, 2019; Graham et al., 2020; Mathur et al., 2020a; Freitag et al., 2021) or ignore the severity of translation errors (Freitag et al., 2021), and thus may mislead system development by incorrect judgements. Therefore, it is important to study the reliability of automatic metrics and follow best practices for the automatic evaluation of systems.

Significant research effort has been applied to evaluate automatic metrics in the past decade, including annual metrics evaluation at the WMT conference and other studies (Callison-Burch et al., 2007; Przybocki et al., 2009; Stanojević et al., 2015; Mathur et al., 2020b). Most research has focused on comparing sentence-level (also known as segment-level) correlations between metric scores and human judgements; or system-level (e.g., scoring an entire test set) correlations of individual system scores with human judgement. Mathur et al. (2020a) emphasize that this scenario is not identical to the common use of metrics, where instead, researchers and practitioners use automatic scores to compare a pair of systems, for example when claiming a new state-of-the-art, evaluating different model architectures, deciding whether to publish results or to deploy new production systems.

The main objective of this study is to find an automatic metric that is best suited for making a *pairwise ranking of systems* and measure how much we can rely on the metric's binary verdicts that one MT system is better than the other. We design a new methodology for pairwise system-level evaluation of metrics and use it on – to the best of our knowledge – the largest collection of human judgement of machine translation outputs which we release publicly with this research. We investigate the reliability of metrics across different language pairs, text domains and how statistical tests over automatic metrics can help to increase decision confidence. We examine how the common use of BLEU over the past years has possibly negatively affected research decisions. Lastly, we re-evaluate past findings and put them in perspective with our work. This research evaluates not

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 478–494
November 10–11, 2021.

only the utility of MT metrics in making pairwise comparisons specifically – it also contributes to the general assessment of MT metrics.

Based on our findings, we suggest the following best practices for the use of automatic metrics:

1. Use a pretrained metric as the main automatic metric; we recommend COMET. Use a string-based metric for unsupported languages and as a secondary metric, for instance ChrF. Do not use BLEU, it is inferior to other metrics, and it has been overused.
2. Run a paired significance test to reduce metric misjudgement by random sampling variation.
3. Publish your system outputs on public test sets to allow comparison and recalculation of different metric scores.

2 Data

In this section, we describe test sets, the process for collecting human assessments, and MT systems used in our analysis. We publish all human judgements, metadata, calculated metrics scores, and the code with replication of our findings and promoting further research. We cannot release the proprietary test sets and so system outputs for legal reasons. The collection is available at https://github.com/MicrosoftTranslator/ToShipOrNotToShip. Moreover, we plan to evaluate new metrics emerging in the future.

2.1 Test sets

When evaluating our models, we use internal test sets where references are translated by professional translators from monolingual data. Freitag et al. (2020) have demonstrated that the quality of test set references plays an important role in automatic metric quality and correlation with human judgement. To maintain a high quality of our test sets, we create them by a two-step translation process: the first professional translator translates the text manually without post-editing followed by a second independent translator confirming the quality of the translations. The human translators are asked to translate sentences in isolation; however, they see context from other sentences.

The test sets are created from authentic source sentences, mostly drawn from news articles (news domain) or cleaned transcripts of parliamentary discussions (discussion domain). The news domain test sets are used in both directions, where the authentic side is mostly English, Chinese, French, or German. The discussion domain test sets are used in the direction from authentic source to translationese reference, e.g., we have two distinct test sets, one for English to Polish and second for Polish to English. Furthermore, some systems are evaluated using various other test sets.

We evaluate 101 different languages within 232 translation directions.[1] The size of the test sets can vary, and more than one test set or its subsets can be used for a single language direction. The average size of our test sets is 1017 sentences. The distribution of evaluated systems is not uniform, some language pairs are evaluated only a few times and while others repeatedly with different systems. The majority of the language pairs are English-centric, however, we evaluate a small set of French, German, and Chinese-centric systems (together only 90 system pairs). Details about the system counts of evaluated language pairs and average test set sizes can be found in the Appendix in Table 7.

2.2 Manual quality assessment

Our human evaluation is run periodically to confirm translation quality improvements by human judgements. For this analysis, we use human annotations performed from the middle of 2018 until early 2021. All human judgements were collected with identical settings with the same pool of human annotators. Thus, the human annotations should have similar distributions and characteristics.

The base unit of our human evaluation is called *a campaign*, in which we commonly compare two to four systems in equal conditions: We randomly draw around 500 sentences from a test set, translate them with each system and send them to human assessment. Each human annotator on average annotates 200 sentences, thus a system pair is evaluated by five different annotators (each annotating distinct set of sentences translated by both systems).

We use source-based Direct Assessment (*DA*, Graham et al., 2013) for collecting human judgements, where bilingual annotators are asked to rate all translated sentences on a continuous scale between 0 to 100 against source sentence without access to reference translations. This eliminates reference bias from human judgement by design.

We use the implementation of DA in the Appraise evaluation framework (Federmann, 2018),

[1] We compare metrics only over the intersection of languages supported by all evaluated metrics, which means that we use only 39 different target languages when Prism is part of the evaluation.

the same as is used in WMT since 2016 for out-of-English human evaluation (Bojar et al., 2016a).

We do not use crowd workers as human annotators. Instead, we use paid bi-lingual speakers that are familiar with the topic and well-qualified in the annotation process. Moreover, we track their performance, and those who fail quality control (Graham et al., 2013) are permanently removed from the pool of annotators, so are their latest annotations. This increases the overall quality of our human quality assessment.

We have two additional constraints in contrast to the original DA. Firstly, each system is compared on the same set of sentences which removes the problem of a system potentially benefitting from an easier set of randomly selected sentences. Moreover, it allows us to use a stronger paired test that compares differences in scoring of equal sentences instead of an unpaired one that evaluates scores of both systems in isolation. We use the Wilcoxon signed-rank test (Wilcoxon, 1946) in contrast to the Mann-Whitney U-test (Mann and Whitney, 1947) originally suggested for DA (Graham et al., 2017). Secondly, each annotator is assigned the same number of sentences for each evaluated system which mitigates bias from different rating strategies as each system is affected evenly by each annotator.

When calculating the system score, we take the average of human judgements.[2] We analyze human judgements for 4380 systems and 2.3 M annotated sentences. This data is one and a half orders of magnitude larger than the data used at WMT Metric Shared Tasks, which evaluate around 170 systems each year (see Section 6).

2.3 Systems

We evaluate competing systems against human judgement. The system pairs could be separated into three groups: (1) model improvements, (2) state-of-the-art evaluation, and (3) comparisons with third-party models. The first group contains system pairs where one system is a strong baseline (usually our highest quality system so far) and the second system is an improved candidate model; this group evaluates stand-alone models without additional pre- and post-processing steps (e.g., rule-based named entity matching). The second group contains pairs of the candidate for the new best performing system and the current best performing system. The third group compares our best-performing model at the time with a publicly available third-party MT system.

Analyzing the variety of systems, hyperparameters, training data, and even architectures is out of the scope of this paper. However, all models are based on neural architectures.

	Metric	Sentence-level	Use human data	Need reference	Multiple refer.	Languages
string-based	BLEU	—	—	✓	✓	any
	CharacTER	✓	—	✓	—	any
	ChrF	✓	—	✓	—	any
	EED	✓	—	✓	—	any
	TER	✓	—	✓	—	any
pretrained	BERTScore	✓	—	✓	—	104
	BLEURT	✓	✓	✓	—	*
	COMET	✓	✓	✓	—	100
	ESIM	✓	✓	✓	—	104
	Prism	✓	—	✓	—	39
	COMET-src	✓	✓	—	n/a	100
	Prism-src	✓	—	—	n/a	39

Table 1: Comparison of selected string-based and pretrained automatic evaluation metrics. We mark metrics designed to work at sentence-level, fine-tuned on human judgements, requiring reference(s), or supporting multiple references, and report the number of supported languages. *BLEURT is built on top of English-only BERT (Devlin et al., 2019) in contrast to BERTScore and ESIM that use multilingual BERT.

3 Automatic metrics

In this study, we investigate metrics that were shown to provide promising performance in recent studies (see Section 6) and currently most widely used metrics in the MT field.[3] We focus on language-agnostic metrics, therefore we do not include metrics supporting only a small set of languages. The full list of evaluated metrics and their main features is presented in Table 1.

Two categories of automatic machine translation metrics can be distinguished: (1) string-based metrics and (2) metrics using pretrained models. The former compares the coverage of various substrings between the human reference and MT output texts. String-based methods largely depend on the quality of reference translations. However, their advantage is that their performance is predictable as it can be

[2] We do not assume a normal distribution of annotator's annotations; therefore, we do not use z-score transformation.

[3] The YiSi – high correlating metric (Ma et al., 2019) – was not publicly available at the time of our evaluation.

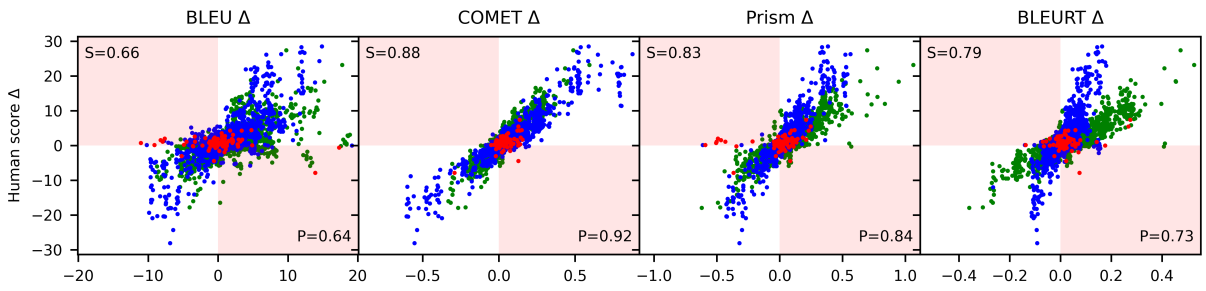


Figure 1: Each point represents a difference in average human judgement (y-axis) and a difference in automatic metric (x-axis) over a pair of systems. Blue points are system pairs translating from English; green points into English; red points are non-English system pairs (a few French, German, or Chinese-centric system pairs). We report Spearman's ρ correlation in the top left corner and Pearson's r in the bottom right corner. Metrics disagree with human ranking for system pairs in pink quadrants. Other metrics are in Figure 2 in the Appendix.

easily diagnosed which substrings affect the score the most. The latter category of pretrained methods consists of metrics that use pretrained neural models to evaluate the quality of MT output texts given the source sentence, the human reference, or both. They are not strictly dependent on the translation quality of the human reference (for example, they can better evaluate synonyms or paraphrases). However, their performance is influenced by the data on which they have been trained. Moreover, the pretrained models introduce a black-box problem where it is difficult to diagnose potential unexpected behavior of the metric, such as various biases learned from training data.

For all metrics, we use the recommended implementation. See Appendix A for implementation details. Most metrics aim to achieve a positive correlation with human assessments, but some error metrics, such as TER, aim for a negative correlation. We simply negate scores of metrics with anticipated negative correlations. Pretrained metrics usually do not support all languages, therefore to ensure comparability, we evaluate metrics on a set of language pairs supported by all metrics.

4 Evaluation

4.1 Pairwise score differences

Most previous works studied the system-level evaluation of MT metrics in an isolated scenario correlating individual systems with human judgements (Callison-Burch et al., 2007; Mathur et al., 2020b). They have mostly employed Pearson's correlation (see Section 6) as suggested by Macháček and Bojar (2014) and evaluated each language direction separately. However, Mathur et al. (2020a) suggest using a pairwise comparison as a more accurate scenario for the general use of metrics.

As the primary unit, we use the difference in metric (or human) scores between system A and B:

$$\Delta = \text{score}(\text{System A}) - \text{score}(\text{System B})$$

We gather all system pairs from each campaign separately as only systems within a campaign are evaluated under equal conditions. All campaigns compare two, three, or four systems, which results in one, three, or six system pairs, respectively.

To understand the relationship between metrics and absolute human differences, we plot these differences and calculate Pearson's and Spearman's correlations in Figure 1. All metrics exhibit a positive correlation with human judgements but differ in behavior. For example, COMET has the smallest deviation which results in the highest correlation with human judgements. However, when we evaluate into-English and from-English language directions separately, we observe that COMET, Prism, and mainly BLEURT have inconsistent value ranges for different language pairs.[4]

Hence, we cannot assume equal scales for one metric and different language pairs, so we can not use Pearson's nor Spearman's correlation in pairwise metrics evaluation. Nonetheless, we provide both correlations in Appendix Table 8 for the complete picture.

4.2 Pairwise system-level metric quality

As standard correlation cannot be used, we investigate a different approach to evaluation. We advocate that the most important aspect of a metric

[4] A possible explanation for BLEURT is that it is trained on English-only. But this does not explain other metrics.

is to make reliable binary pairwise decisions (i.e., which of two systems provides a higher translation quality) without the focus on the magnitude of difference.[5] Therefore, given the size of our data set, we propose to use accuracy on binary comparisons: which system is better when human rankings are considered gold labels.

We define the accuracy as follows. For each system pair, we calculate the difference of the metric scores (metricΔ) and the difference in average human judgements (humanΔ). We calculate accuracy for a given metric as the number of rank agreements between metric and human deltas divided by the total number of comparisons:

$$\text{Accuracy} = \frac{|\text{sign}(\text{metric}\Delta) = \text{sign}(\text{human}\Delta)|}{|\text{all system pairs}|}$$

Assuming human judgements as a gold labels, accuracy gets an intrinsic meaning of how „reliable" a given metric is when making pairwise comparisons. On the other hand, accuracy does not take into account that two systems can have comparable quality, and thus the accuracy of a metric can be over-estimated by chance if a small human score difference has the same sign as the difference in a metric score. To overcome this issue, we also calculate accuracy over a subset of system pairs, where we remove system pairs that are deemed to not be different based on Wilcoxon's signed-rank test over human judgements.

In order to estimate the confidence interval for accuracy, we use the bootstrap method (Efron and Tibshirani, 1994), for more details see Appendix B. We consider all metrics that fall into the 95% confidence interval of the best performing metric to be comparable. We visualize the clusters of best-performing metrics in our analysis with a grey background of table cells.

5 Results

5.1 Which metric is best suited for pairwise comparison?

In this section, we examine all available system pairs and investigate which metric is best suited for making a pairwise comparison.

The results presented in Table 2 show that pre-trained methods (except for Prism-src) generally have higher accuracy than string-based methods,

[5]The value of score difference (e.g., a difference of 2 BLEU) is important mainly to measure the confidence of a ranking decision.

	All	0.05	0.01	0.001	Within
n	3344	1717	1420	1176	541
COMET	**83.4**	**96.5**	**98.7**	**99.2**	**90.6**
COMET-src	83.2	95.3	97.4	98.1	89.1
Prism	80.6	94.5	97.0	98.3	86.3
BLEURT	80.0	93.8	95.6	98.2	84.1
ESIM	78.7	92.9	95.6	97.5	82.8
BERTScore	78.3	92.2	95.2	97.4	81.0
ChrF	75.6	89.5	93.5	96.2	75.0
TER	75.6	89.2	93.0	96.2	73.9
CharacTER	74.9	88.6	91.9	95.2	74.1
BLEU	74.6	88.2	91.7	94.6	74.3
Prism-src	73.4	85.3	87.6	88.9	77.4
EED	68.8	79.4	82.4	84.6	68.2

Table 2: Accuracies for binary comparisons for ranking system pairs. Column "All" shows the results for system pairs. Each following column evaluates accuracy over a subset of systems that are deemed different based on human judgement and a given alpha level in Wilcoxon's test. Column "Within" represents a subset of systems where the human judgement p-value is between 0.05 and 0.001. "n" represents the number of system pairs used to calculate accuracies in a given column. Only the scores in each column are comparable. Results with a grey background are considered to be tied with the best metric.

which confirms findings from other studies (Ma et al., 2018, 2019; Mathur et al., 2020b). COMET reaches the highest accuracy and therefore is the most suited for ranking system pairs. The runner-up is *COMET-src*, which is a surprising result because, as a quality estimation metric, it does not use a human reference. This opens possibilities to use monolingual data in machine translation systems evaluation in an effective way. On the other hand, the second reference-less method *Prism-src* does not reach high accuracy, struggling mainly with into-English translation directions (see Figure 2 in the Appendix). In terms of string-based metrics, the highest accuracy is achieved by ChrF, which makes it a better choice for comparing system pairs than the widely used BLEU.

To minimize the risk of being affected by random flips due to a small human score delta, we also explore the accuracy after removing systems with comparable performance with respect to Wilcoxon's test over human judgements. We incrementally remove system pairs not significantly different with alpha levels of 0.05, 0.01, and 0.001. As expected, removing pairs of most likely equal-quality systems increases the accuracy, however, no metric reaches 100% accuracy even for a set of

n	Everything 1717 ↓	Into EN 922	From EN 768	Non Latin 131	Logograms 44	Non WMT 484	Discussion 78
COMET	**96.5**	**95.3**	**98.3**	**96.2**	**90.9**	**97.3**	**93.6**
COMET-src	95.3	93.5	97.7	95.4	88.6	96.7	**93.6**
Prism	94.5	92.2	98.2	**96.2**	**90.9**	96.9	83.3
BLEURT	93.8	93.8	95.1	93.1	84.1	94.6	89.7
ESIM	92.9	90.6	96.6	93.9	86.4	94.8	76.9
BERTScore	92.2	91.2	94.1	95.4	88.6	92.8	71.8
ChrF	89.5	88.7	91.0	95.4	88.6	89.7	57.7
TER	89.2	87.6	91.7	90.1	72.7	90.9	70.5
CharacTER	88.6	86.4	91.7	88.5	70.5	91.9	69.2
BLEU	88.2	86.9	90.5	92.4	79.5	89.9	61.5
Prism-src	85.3	80.8	91.4	84.0	65.9	91.7	84.6
EED	79.4	75.1	84.8	82.4	54.5	83.1	60.3

Table 3: Accuracies for ranking system pairs. Each column represents a different subset of significantly different system pairs with alpha level 0.05. Results with a grey background are considered to be tied with the best metric. Accuracies across columns are not comparable as they compare different sets of systems.

strongly different systems with an alpha level of 0.001. This implies that either current metrics cannot fully replace human evaluation or remaining systems are incorrectly assessed by human annotators.[6] Moreover, we observe that the ordering of metrics by accuracy remains the same even after removing system pairs with comparable performance, which implies that accuracy is not negatively affected by non-significantly different system pairs. Due to that where we analyze only subsets of the data, we use systems that are statistically different by human judgement with an alpha level of 0.05.

Ma et al. (2019) have observed that system outliers, i.e., systems easily differentiated from other systems, can inflate Pearson's correlation values. Moreso, Mathur et al. (2020a) demonstrated that after removing outliers some metrics would actually have negative correlation with humans. To analyze if outliers might affect our accuracy measurements and the ordering of metrics, we analyze a subset of systems with human judgement p-values between 0.05 and 0.001, i.e. removing system pairs that have equal quality and outlier system pairs that are easily distinguished. From column "Within" in Table 2, we see that the ordering of metrics remains unchanged. This shows that accuracy is not affected by outliers making it more suitable for metrics evaluation than Pearson's ρ.

[6]An alpha level of 0.001 could (mis)lead to the conclusion that 0.1% of human judgements are incorrect. However, the alpha level only determines if two systems are different enough and cannot be used to conclude that a human pairwise rank decision is incorrect.

5.2 Are metrics reliable for non-English languages and other scenarios?

The superior performance of pretrained metrics raises the question if unbalanced annotation data might be responsible; around half of the systems translate into English. Moreover, COMET and BLEURT are fine-tuned on human annotations from WMT on the news domain. This could lead to an unfair advantage when being evaluated w.r.t. human judgements.[7] To shed more light on metrics behavior and robustness, we analyze various subsets, including into and from English translation directions, languages with non-Latin scripts, and non-news domain.

We showed in Section 4.1 that some metrics perform differently for systems translating from and into English. Analyzing this scenario in Table 3 reveals that BLEURT does better (the second best metric) for "into English" translation compared to other metrics. It is surprising that BLEURT has a high accuracy for unseen "from English" pairs which suggests that BLEURT might have learned some kind of string-matching. We also observe in Table 3 gains for Prism for the "from English" directions. The overall ranking of metrics, however, remains similar which confirms that the high accuracy of pretrained methods compared to the string-based ones cannot be attributed to the abundance of system pairs with English as the target.

[7]We double-checked and removed all campaigns containing test sets from WMT 2015 to 2020 from our work and analysis.

When investigating language pairs with non-Latin (Arabic, Russian, Chinese, ...) or logogram-based scripts (Chinese, Korean and Japanese) as the target languages, we observe a slight drop in metric ranks for some pretrained metric in contrast to higher score for ChrF. This indicates that non-Latin scripts might be a challenge for pretrained metrics but more analysis would be required here. For an summary on individual language pairs, refer to Table 9 in the Appendix.

We also investigate if some pretrained methods might have an unfair advantage due to being fine-tuned on human assessments in the news domain. For this, we analyze a subset of news test sets with target languages that were not part of WMT human evaluation (i.e., languages which those methods have not been fine-tuned on) and call this set "non-WMT", and also system pairs evaluated on a proprietary test sets in the EU parliamentary discussions domain covering ten languages. Neither results on non-WMT nor discussion domains in Table 3 show a change in the ranking of metrics, suggesting that COMET is not overfitted to the WMT news domain or WMT languages. Somewhat surprisingly, we actually see a drop in accuracy for the string-based metrics for the discussion domain. We speculate this might be due to their inability to forgivingly match disfluent utterances to expected fluent translations (Salesky et al., 2019).

Overall, the results for various subsets show a similar ordering of metrics based on their accuracy, confirming the general validity of our results.

5.3 Are statistical tests on automatic metric worth it?

Mathur et al. (2020a) studied the effects of statistical testing of automatic metrics and observed that even large metric score differences can disagree with human judgement. They have shown that even for a BLEU delta of 3 to 5 points, a quarter of these systems are judged by humans to differ insignificantly in quality or to contradict the verdict of the metric. In our analysis, we have 203 system pairs deemed statistically significant by humans (p-value smaller than 0.05) for which using BLEU results in a flipped ranking compared to humans. The median BLEU difference for these system pairs is 1.3 BLEU points. This is concerning as BLEU differences higher than one or two BLEU points are commonly and historically considered to be reliable by the field.

	No test	Boot. ↓	Type II Err.
COMET	**83.4**	**95.1**	204 (**17.3%**)
COMET-src	83.2	94.2	242 (19.4%)
BLEURT	80.0	92.0	349 (25.4%)
Prism	80.6	91.3	200 (18.3%)
BERTScore	78.3	87.9	244 (20.9%)
ChrF	75.6	85.4	350 (27.3%)
BLEU	74.6	83.4	378 (27.4%)
Prism-src	73.4	81.5	325 (29.4%)

Table 4: The first column shows accuracy for all system pairs and represent situation, where we would trust any small score difference. The second column shows accuracy, where we ignore systems considered to be tied with respect to the paired bootstrap resampling test. The third column represents the number of system pairs incorrectly decided to be non-significantly different by the paired bootstrap resampling and the percentage from all non-significant systems.

In this section, we corroborate that statistical significance testing can largely increase the confidence of the MT quality improvement and increase the accuracy of metrics. We compare how accurate a metric would be under two situations: either when not using statistical testing and solely trusting in the metric score difference; or when using statistical testing and throwing away systems that are not statistically different.

We evaluated the first situation in Section 5.1 and the results are equal with the first column of Table 2. For the second situation, we calculate accuracy only over the system pairs that are statistically different. We use paired bootstrap resampling (Koehn, 2004), a non-parametric test, to calculate the statistical significance for a pair of systems.[8]

Additionally, the second situation introduces type II errors which represent systems where the statistical significance test rejected a system pair as being non-significant, but humans would judge the given pair as significantly different. In other words, it shows how many system pairs are incorrectly rejected as non-significantly different. See Appendix C for a detailed explanation.

From the results in Table 4, we can see that if we apply paired bootstrap resampling on automatic metrics with an alpha level 0.05 the accuracy increases by around 10% for all metrics in con-

[8] Approximate randomization (Riezler and Maxwell III, 2005) can be used as an alternative test, and for metrics based on the average of sentence-level scores, we can use also tests such as the Student t-test.

trast to not using statistical testing. On the other hand, when using statistical testing, we introduce type II errors, where 17.3%, for COMET, of non-significantly different system pairs are deemed significantly different by humans.[9]

In conclusion, we corroborate that using statistical significance tests largely increases reliability in automatic metric decisions. We encourage the usage of statistical significance testing, especially in the light of Marie et al. (2021) who show that statistical significance tests are widely ignored.

5.4 Does BLEU sabotage progress in MT?

Freitag et al. (2020) have shown that reference translations with string-based metrics may systematically bias against modeling techniques known to improve human-judged quality and raised the question of whether previous research has incorrectly discarded approaches that improved the quality of MT due to the use of such references and BLEU. They argue that the use of BLEU might have mislead many researcher in their decisions.

In this section, we investigate the hypothesis if the usage of BLEU negatively affects model selection. To do so, we compare two groups of system pairs based on the premise if they could be directly affected by BLEU. The first group contains pairs of incremental improvements of our systems. We can assume that incremental models use similar architecture, data, and settings, although we do not study particular changes. We use BLEU as the main automatic metric to guide model development. If BLEU shows improvements, we evaluate models with human judgements to make a final deployment decision. Therefore, systems with degraded BLEU scores which would be deemed improved by humans are missing in this group as we reject them based on BLEU scores during development. The second group contains independent system pairs, which use different architectures, data, settings, and therefore BLEU has not been used to preselect them. In this group, we compare our systems with publicly available third-party MT systems.

We compare three models within the same campaign, two internal[10] and one external system. Thus, the same annotators annotated the same sentences from all three systems under the same conditions. We call system pairs comparisons between two internal models "incremental", and comparisons between the newer internal model and the external model as "independent".

n	Incremental 161 ↓	Independent 246
BLEU	**99.4**	90.7
BERTScore	98.8	91.5
ESIM	98.8	92.3
Prism	98.1	94.3
ChrF	98.1	91.5
COMET	98.1	**98.4**
COMET-src	97.5	98.8
CharacTER	97.5	89.8
Prism-src	96.9	92.7
BLEURT	96.9	93.5
TER	95.7	91.5
EED	78.9	78.0

Table 5: Evaluation of incremental and independent system pairs. We use a subset of 333 system pairs significantly different based on Wilcoxon's test and alpha level of 0.05 over human judgement. Results with grey background are considered tied with the best metric.

Over the past three years we carried out 333 campaigns across 17 language pairs (each campaign comparing three models), resulting in almost 530000 human annotations.

The results in Table 5 show that for independent systems, the ranking of the metrics is comparable with results in Table 3. Pretrained metrics generally outperform string-based ones and COMET is in the lead. However, when inspecting the incremental systems, BLEU wins. This indicates that BLEU influenced our model development and we rejected models that would have been preferred by humans.

Another possible explanation is that systems preselected by BLEU are easy to differentiate by all metrics. This could explain why all metrics have high accuracy in contrast to the "Independent" column and most of them are in a single cluster.

In conclusion, results showing BLEU as the metric with the highest accuracy where we would expect pretrained metrics to dominate, suggests that BLEU affected system development and we rejected improved models due to the erroneous degradation seen in the BLEU score. However, this is indirect evidence as for sound conclusions we would need to evaluate those rejected systems with other metrics and human judgement as well.

[9] Wilcoxon's test on human judgement and alpha level 0.05.

[10] The pair of internal models contains the best model from the last year and our latest improved model.

	WMT Metric task ↓ n	2020b 168 (no outliers)	2020b 184	2019 225	2018 149	2017 152	2016b 120	2015 121	2014 92	2013 135
string-based	BLEU	.740 (.727)	.837 (.832)	.906	.955	.910	.873	.841	**.910**	.845
	CharacTER	.735 (.723)	.873 (.871)	.942	.964	.932	**.938**			
	ChrF	.743 (.730)	.743 (.864)	.948	.959	**.942**	.911	.908		
	EED	.762 (.750)	.888 (.885)	.951						
	METEOR							.900	.884	**.878**
	NIST			.860	.970	.921	.870	.854	.899	.834
	TER	.609 (.668)	.704 (.763)	.922	.953	.918	.863	.837	.860	.788
	WER			.917	.934	.913	.846	.829	.818	.752
pretrained	BEER			.942	**.973**	.938	.925	**.942**		
	BLEURT	.764 (.752)	.902 (.900)							
	COMET	.711 (**.762**)	.853 (**.908**)							
	ESIM	**.770** (.755)	**.906** (.902)							
	Prism	.677 (.710)	.846 (.886)							
	YiSi-1	.759 (.744)	.894 (.890)	**.967**	**.973**					

Table 6: "n" is sum of systems in each study used to calculate aggregated correlation. The results in brackets are without systems on English into Chinese. Correlations are comparable only within columns.

6 Meta Analysis

We analyze findings from past research to put our results in the broader context. We focus on the results on the system-level evaluation, however, a large part of the research studied a sentence-level evaluation. The largest source of metrics evaluation is yearly WMT Metric Shared Task occurring over more than the past ten years (Callison-Burch et al., 2007), where various methods are evaluated with human judgement over the set of submitted systems and language pairs in WMT News Translation Shared Tasks. Recently, Freitag et al. (2021) reevaluated two translation directions from WMT 2020 with the multidimensional quality metric framework and raised a concern that general crowd-sourced annotators used in into-English evaluation in WMT prefer literal translations and have a lower quality than some automatic metrics.

Past studies evaluate system-level correlations with Pearson's correlation calculated for each translation direction separately. We are interested in how metrics correlate with human judgement in general across different language pairs. Thus, to generalize the past findings, we use the Hunter-Schmidt method (Hunter and Schmidt, 2004), which allows combining already calculated correlations with various sizes. We use it to generalize correlations within each study across all language pairs. For this purpose, Hunter-Schmidt is effectively a weighted mean of the raw correlation coefficients.

Although past studies evaluated a larger number of methods and their variants, we have selected a subset of metrics that are evaluated in more than one study or showed promising performance over other metrics in a given study. When a study evaluated several variants of a metric with various parameters, we selected the setting closest to either the recommended setting in the recent years, such as SacreBLEU, or a setting that is used in the later evaluation study, mainly in Mathur et al. (2020b).

Meta-analysis in Table 6 shows that pretrained methods outperform string-based methods as concluded by Mathur et al. (2020b); Ma et al. (2019, 2018). The second important observation is that there was not a single year where BLEU had a higher correlation than ChrF. This supports our conclusions and shows that the MT community had results supporting the deprecation of BLEU as a standard metric for several years. Comparing the pretrained methods, ESIM is the best performing method in general (Mathur et al., 2020b), while COMET is the best performing method when removing the suspicious system.

In the study by Mathur et al. (2020b), COMET under-performed other pretrained metrics. We found out that submitted COMET scores failed to score one English-Chinese system with tokenized output. However, we obtain valid COMET scores on that system output when replicating the results. Moreover, we have not seen any problems with COMET on Chinese. As this one system largely skews Pearson's correlation, we also present analysis without English-Chinese systems in Table 6.

7 Discussion

We corroborate results from past studies that pretrained methods are superior to string-based ones. However, pretrained methods are relatively new

techniques and we can potentially discover significant drawbacks, for example, they could resemble biases from training data, fail on particular domains, or prefer fluency over adequacy. Another problem could arise if an MT system would be trained on the same data as the metric was or if it incorporates the same pretrained model, for example, XLM-R (Conneau et al., 2020) used by COMET. Pretrained methods support only a selected set of languages and the quality can differ for each of them. Thus, we argue that the string-based method should be used as a secondary metric.

An interesting solution to dissipate potential drawbacks of any metric would be if different research groups preselect a different primary pretrained metric in advance to lead their research decisions and to discover improvements not apparent under other metrics. However, we fear that it could lead to "metric-hacking", i.e., picking a metric that confirms results. Therefore, we recommend using COMET as the primary metric. And to use ChrF, the best performing string-based method, as a secondary metric and for unsupported languages.

A surprising results is the high accuracy of COMET-src, a reference-free metric. It allows automatic evaluation over monolingual domain-specific testsets as suggested by Agrawal et al. (2021).

Limitations of BLEU are well-known (Reiter, 2018; Mathur et al., 2020a). Callison-Burch et al. (2006) argued that MT community is overly reliant on it, which Marie et al. (2021) confirmed by showing that 98.8% of MT papers use BLEU. We present indirect evidence that the over-use of BLEU negatively affects MT development and support deprecation of BLEU as the evaluation standard.

We show that the reliability of metrics decisions can be increased with statistical significance tests. However, Dror et al. (2018) point out the assumption of statistical significance tests that data samples are independent and adequately distributed is rarely true. Also, statistical significance tests do not account for random seed variation across training runs. Thus, one should be cautious when making conclusions based on small metrics improvements. Wasserstein et al. (2019) give recommendations for a better use of statistical significance testing.

Marie et al. (2021) have shown that almost 40% of MT papers from 2020 copied score from different papers without recalculating them, which is a concerning trend. Also, new and better metrics will emerge and there is no need to permanently adhering to a single metric. Instead, the simplest and most effective solution to avoid the need to copy scores or stick to obsolete metric is to always publish translated outputs of test sets along with the paper. This allows anyone to recalculate scores with different tools and/or metrics and makes comparisons with past (and future) research easier.

There are some shortcomings in our analysis. We have only a handful of non-English systems, therefore we cannot conclude anything about the behaviour of the metrics for language pairs without English. Similarly, the majority of our language pairs are high-resource, therefore, we cannot conclude the reliability of metrics for low-resource languages. Lastly, many of our translation directions are from translationese into authentic, which as Zhang and Toral (2019) showed is the easier direction for systems to score high by human judgement. These are potential directions of future work.

Lastly, we assume that human judgement is the gold standard. However, we need to keep in mind that there can be potential drawbacks of the method used for human judgement or human annotators fail to capture true assessment as Freitag et al. (2021) observe. For example, humans cannot explicitly mark critical errors in DA and instead they usually assign low assessment scores.

8 Conclusion

We show that metrics can use a different scale for different languages, so Pearson's correlation cannot be used. We introduce accuracy as a novel evaluation of metrics in a pairwise system comparison.

We use and release a large collection of the human judgement confirming that pretrained metrics are superior to string-based. COMET is the best performing metric in our study, and ChrF is the best performing string-based method. The surprising effectiveness of COMET-src could allow the use of large monolingual test sets for quality estimation.

We do not see any drawbacks of the metrics when investigating various languages or domains, especially, for methods pretrained on human judgement. We present indirect evidence that the over-use of BLEU negatively affects MT development.

We show that statistical testing of automatic metrics largely increases the reliability of a pairwise decision based on automatic metric scores.

We endorse the recommendation for publishing translated outputs of research systems to allow comparisons and recalculation of scores in the future.

Acknowledgments

We are grateful for a feedback and review of the paper to many researchers, namely: Shuoyang Ding, Markus Freitag, Hieu Hoang, Alon Lavie, Jindřich Libovický, Nitika Mathur, Mathias Müller, Martin Nejedlý, Martin Popel, Matt Post, Qingsong Ma, Richardo Rei, Thibault Sellam, Aleš Tamchyna, anonymous reviewers, and our colleagues.

References

Sweta Agrawal, George Foster, Markus Freitag, and Colin Cherry. 2021. Assessing reference-free peer evaluation for machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1158–1171, Online. 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, 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 Task Papers*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016b. Results of the WMT16 Metrics Shared Task. In *Proceedings of the First Conference on Machine Translation*, pages 199–231, Berlin, Germany. 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.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *arXiv preprint arXiv:2104.14478*.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. 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 and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 72–81, Online. Association for Computational Linguistics.

John E Hunter and Frank L Schmidt. 2004. *Methods of meta-analysis: Correcting error and bias in research findings*. Sage.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

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: shared task papers*, pages 671–688.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90.

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.

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, Maryland, USA. Association for Computational Linguistics.

Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *arXiv preprint arXiv:2106.15195*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 Metrics Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. 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.

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. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Mark Przybocki, Kay Peterson, Sébastien Bronsart, and Gregory Sanders. 2009. The NIST 2008 Metrics for machine translation challenge—overview, methodology, metrics, and results. *Machine Translation*, 23(2):71–103.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Stefan Riezler and John T Maxwell III. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 57–64.

Elizabeth Salesky, Matthias Sperber, and Alexander Waibel. 2019. Fluent translations from disfluent speech in end-to-end speech translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2786–2792, Minneapolis, Minnesota. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. 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*, volume 200. Citeseer.

Peter Stanchev, Weiyue Wang, and Hermann Ney. 2019. EED: Extended Edit Distance Measure for Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 514–520, Florence, Italy. Association for Computational Linguistics.

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, Lisbon, Portugal. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. 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: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Ronald L. Wasserstein, Allen L. Schirm, and Nicole A. Lazar. 2019. Moving to a world beyond "p<0.05". *The American Statistician*, 73(sup1):1–19.

Frank Wilcoxon. 1946. Individual comparisons of grouped data by ranking methods. *Journal of economic entomology*, 39(2):269–270.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

A Metrics Implementation Details

We use the most common implementation with default or recommended parameters to simulate standard metric usage.

For *BLEU* (Papineni et al., 2002), *ChrF* (Popović, 2015) and *TER* (Snover et al., 2006) metrics, we use SacreBLEU implementation `https://github.com/mjpost/sacrebleu/` version 1.5.0. We use "mteval-v13a" tokenizer for all language pairs except for Chinese and Japanese which use their own tokenizer, as is recommended.

For *CharacTER* (Wang et al., 2016), we use `https://github.com/rwth-i6/CharacTER` commit c4b25cb.

For *EED* (Stanchev et al., 2019), we use `https://github.com/rwth-i6/ExtendedEditDistance` commit f944adc.

For *BERTScore* (Zhang et al., 2020), we use `https://github.com/Tiiiger/bert_score` version 0.3.7.

For *BLEURT* (Sellam et al., 2020), we use the recommended model "bleurt-base-128" and implementation `https://github.com/google-research/bleurt` version 0.0.1. It is important to mention, that BLEURT is fine-tuned for English only. Additionally, we evaluated other variants and "bleurt-large-512" performed better than recommended variant. We add it in Table 8.

For *COMET* (Rei et al., 2020), we use recommended model "wmt-large-da-estimator-1719" and for *COMET-src* we use "wmt-large-qe-estimator-1719". The implementation is `https://github.com/Unbabel/COMET` in version 0.0.6. We evaluated all other COMET models, but neither performed better than recommended model.

For *Prism* and *Prism-src* (Thompson and Post, 2020), we use `https://github.com/thompsonb/prism` commit 06f10da.

For *ESIM* (Mathur et al., 2019), we use `https://github.com/nitikam/mteval-in-context`.

B Confidence Interval for Metric Accuracy

To estimate the confidence interval for the best performing metric, we use the bootstrap method (Efron and Tibshirani, 1994). It creates multiple resamples (with replacement) from a set of observations and calculates accuracy on each of these resamples. We employ modified paired bootstrap resampling (Koehn, 2004), a method which we also use for testing statistical significance of the metric difference in Section 5.3. However, the usage is different.

To calculate the bootstrap resampling. First, we note the best performing metric on all system pairs from the collection as metric α. We create 10 000 resamples by drawing system pairs with replacements from the collection of all. For each resample, we calculate accuracy for all metrics. We note

which metrics have equal or higher accuracy than metric α in a given resample.

If metric α outperforms metric X by less than 95% of the time, we draw the conclusion that metric X performs on par with 95% statistical significance to the winning metric α.

C Comparing Statistical Tests

The problem if two systems have the same MT quality is still an open question. Applying statistical tests over the metric scores allows us to confirm if the difference in score is significant or due to a random change based on the set of translated sentences and a given alpha level. To get the gold truth about system equivalence, we employ Wilcoxon's test on human judgement and alpha level 0.05. We use paired bootstrap resampling approach as the statistical test for automatic metrics. Unfortunately, we cannot directly compare the outputs of two statistical tests (for example, the Wilcoxon test on human judgements with the bootstrap resampling on metric scores) as even with the same alpha level, these tests have a different power. Therefore, we need to investigate it in isolation.

The null hypothesis in our setting is that both evaluated systems have the same translation quality. There are two possible outcomes of a statistical test: accept the null hypothesis (i.e. MT quality of systems is *not significantly* different) or reject the null hypothesis (i.e. MT quality of systems is *significantly* different). When observing outcomes of statistical tests over human judgement and over automatic metric, we get four possible outcomes:

		Statistical test on a metric	
		Signif.	Not signif.
Humans	Signif.	Truly differing system pair	Type II Error
	Not signif.	Type I. Error	Systems with the equal MT Quality

There are two outcomes for the statistical test over a metric that we investigate separately.

In the first scenario, the bootstrap resampling confirms the statistical difference between systems. However, even when both tests agree that systems have statistically different MT quality, it still may happen that humans and metrics disagree on which system is better than the other. The goal is to evaluate how accurate metric decisions are if we employ statistical testing. Therefore, we are interested in the accuracy of a metric over system pairs that are deemed statistically different according to the paired bootstrap resampling, in other words, accuracy for system pairs that are either truly different (top left quadrant) or fall into type I. error (bottom left quadrant).

In the second scenario, we want to find out how many system pairs are diagnosed as non-significant even though human judgements would deem them different. For this scenario, we investigate for how many system pairs bootstrap resampling fails to reject the null hypothesis. However, keep in mind that two statistical tests cannot be directly compared because different tests have different power and the type II error will differ based on that.

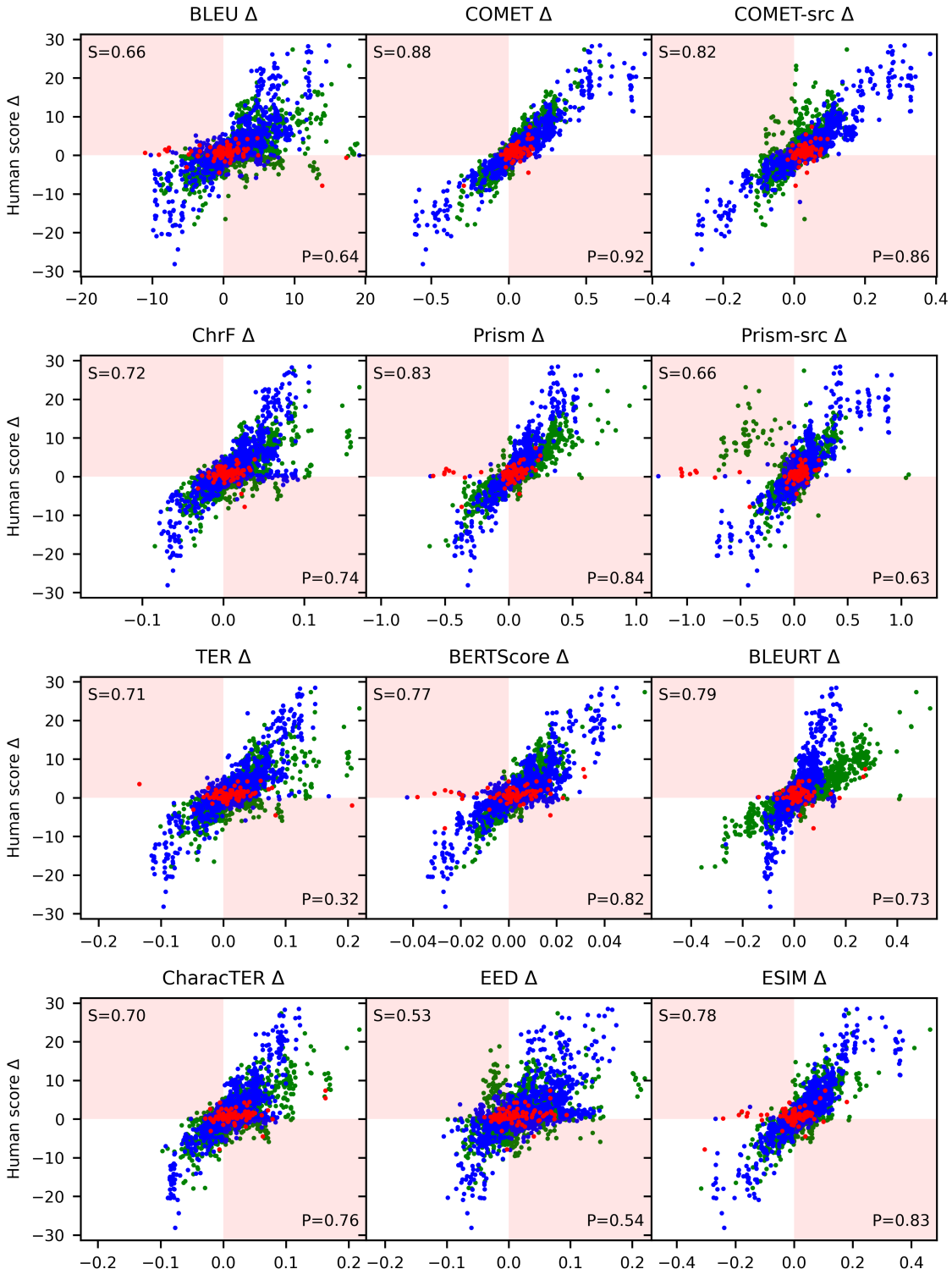


Figure 2: Each point represents a difference in average human judgement (y-axis) and a difference in automatic metric (x-axis) over a pair of systems. Blue points are system pairs translating from English; green points are into English; red points are non-English systems (French, German, and Chinese centric). Spearman's ρ correlation is in top left corner, while Pearson's r is in the bottom right corner. Metrics disagree with human ranking for system pairs in pink quadrants. For better visualization, we have clipped few outliers in BLEU, ChrF, and TER plots.

Language pair	Sys.	Size	Language pair	Sys.	Size	Language pair	Sys.	Size
English - French	145	1034	English - Hindi	58	540	English - Ukrainian	25	988
English - German	139	2544	Polish - English	57	1229	English - Slovak	25	1776
French - English	131	1119	Portuguese - English	57	878	English - Irish	24	463
German - English	122	1212	Swedish - English	57	1116	English - Persian	24	510
Japanese - English	78	925	English - Arabic	56	1054	Slovak - English	23	1476
Chinese - English	74	1029	Korean - English	56	1462	Greek - English	23	1526
Italian - English	71	1156	Czech - English	55	1105	English - Croatian	22	1625
English - Portuguese	70	1679	English - Hungarian	55	1018	English - Welsh	22	497
English - Japanese	67	998	English - Korean	55	550	English - Norwegian	22	1533
English - Swedish	66	1219	English - Turkish	55	1043	English - Hebrew	22	940
English - Chinese	65	2443	English - Thai	54	510	English - Vietnamese	20	1857
English - Danish	64	1186	Hindi - English	54	816	Welsh - English	20	1686
English - Italian	64	1505	Turkish - English	54	1037	Vietnamese - English	20	1697
English - Polish	64	1188	Danish - English	52	986	Catalan - English	20	928
Spanish - English	64	1223	English - Russian	49	1159	English - Urdu	18	448
Dutch - English	63	927	Russian - English	44	736	English - Finnish	17	1802
English - Dutch	61	991	Thai - English	39	457	Tamil - English	16	834
English - Indonesian	61	948	English - Catalan	30	981	English - Lithuanian	16	1997
Indonesian - English	60	703	Hebrew - English	28	870	Lithuanian - English	16	1997
English - Czech	59	1329	English - Romanian	27	1056	English - Maltese	16	489
Arabic - English	59	2674	Romanian - English	27	1094	English - Kiswahili	16	457
English - Spanish	58	1172	English - Greek	27	1936			
Hungarian - English	58	976	Persian - English	26	1372			

Table 7: The column "Sys." represents the number of systems for a given translation direction. We list only translation directions with more than 15 evaluated systems. The column "Size" represents the average test set size for the given direction. We evaluate 232 translation directions in total.

	All	0.05	Within	Spearman	Pearson
n	3344	1717	541	3347	3347
COMET	**83.4**	**96.5**	**90.6**	**0.879**	**0.919**
COMET-src	83.2	95.3	89.1	0.824	0.855
Prism	80.6	94.5	86.3	0.827	0.839
BLEURT-large	80.1	94.4	85.4	0.808	0.748
BLEURT	80.0	93.8	84.1	0.787	0.729
ESIM	78.7	92.9	82.8	0.780	0.835
BERTScore	78.3	92.2	81.0	0.772	0.824
ChrF	75.6	89.5	75.0	0.716	0.739
TER	75.6	89.2	73.9	0.708	0.321
CharacTER	74.9	88.6	74.1	0.700	0.757
BLEU	74.6	88.2	74.3	0.661	0.640
Prism-src	73.4	85.3	77.4	0.661	0.631
EED	68.8	79.4	68.2	0.531	0.541

Table 8: Extended Table 2 with Spearman's and Pearson's correlations over all system pairs. Remaining columns are identical to original table. This table also contain additional BLEURT-large.

	n	COMET	COMET-src	BLEURT	Prism	Prism-src	ESIM	BLEU	ChrF	BERTScore	CharacTER	TER	EED
English - French	62	**98.4**	93.5	90.3	88.7	87.1	82.3	64.5	64.5	62.9	62.9	58.1	41.9
Japanese - English	58	**98.3**	89.7	84.5	**98.3**	89.7	**98.3**	93.1	93.1	96.6	87.9	96.6	82.8
English - Polish	57	**100.0**	**100.0**	98.2	**100.0**	82.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	98.2
Polish - English	55	80.0	**98.2**	83.6	83.6	**98.2**	83.6	83.6	80.0	83.6	80.0	81.8	81.8
Thai - English	54	**100.0**	70.4	96.3	90.7	16.7	88.9	83.3	92.6	90.7	94.4	83.3	20.4
English - German	52	**100.0**	**100.0**	80.8	**100.0**	88.5	**100.0**	75.0	67.3	82.7	73.1	78.8	75.0
Chinese - English	52	96.2	94.2	98.1	92.3	71.2	**100.0**	94.2	96.2	94.2	96.2	86.5	71.2
Turkish - English	51	98.0	**100.0**	**100.0**	96.1	**100.0**	96.1	96.1	96.1	96.1	98.0	96.1	94.1
English - Indonesian	51	**100.0**	**100.0**	96.1	**100.0**	88.2	**100.0**	92.2	98.0	**100.0**	96.1	98.0	96.1
English - Turkish	48	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	87.5	**100.0**	**100.0**	**100.0**	95.8	91.7
English - Swedish	48	91.7	91.7	**97.9**	**97.9**	**97.9**	95.8	93.8	87.5	95.8	95.8	93.8	91.7
Swedish - English	47	97.9	**100.0**	97.9	95.7	89.4	91.5	87.2	91.5	95.7	72.3	91.5	85.1
English - Dutch	47	97.9	97.9	**100.0**	**100.0**	97.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	87.2
Indonesian - English	46	97.8	**100.0**	97.8	97.8	**100.0**	97.8	93.5	97.8	97.8	97.8	91.3	89.1
English - Hungarian	46	**100.0**	**100.0**	97.8	**100.0**	**100.0**	97.8	84.8	82.6	**100.0**	**100.0**	**100.0**	87.0
Czech - English	44	**100.0**	93.2	**100.0**	**100.0**	93.2	**100.0**	**100.0**	**100.0**	95.5	95.5	95.5	20.5
English - Czech	42	**100.0**	**100.0**	92.9	97.6	97.6	97.6	92.9	92.9	97.6	92.9	97.6	**100.0**
English - Danish	42	95.2	95.2	88.1	**97.6**	95.2	92.9	**97.6**	92.9	95.2	92.9	95.2	88.1
Hungarian - English	42	90.5	83.3	78.6	**97.6**	85.7	90.5	90.5	95.2	85.7	92.9	81.0	88.1
Hindi - English	41	95.1	**100.0**	95.1	82.9	73.2	78.0	61.0	61.0	80.5	65.9	73.2	80.5
Dutch - English	40	**100.0**	97.5	97.5	97.5	82.5	97.5	97.5	95.0	97.5	97.5	97.5	75.0
German - English	39	**97.4**	94.9	**97.4**	**97.4**	82.1	94.9	82.1	82.1	**97.4**	82.1	94.9	64.1
Danish - English	39	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	97.4	97.4	**100.0**	89.7	97.4	74.4
Spanish - English	39	94.9	94.9	94.9	**97.4**	92.3	94.9	94.9	94.9	**97.4**	**97.4**	94.9	92.3
English - Russian	37	**100.0**	**100.0**	**100.0**	**100.0**	91.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Portuguese - English	36	**100.0**	94.4	94.4	94.4	94.4	94.4	94.4	88.9	94.4	94.4	94.4	97.2
Korean - English	33	**100.0**	97.0	72.7	87.9	42.4	69.7	63.6	97.0	78.8	90.9	69.7	66.7
English - Portuguese	31	**100.0**	**100.0**	**100.0**	**100.0**	87.1	**100.0**	96.8	**100.0**	**100.0**	**100.0**	96.8	**100.0**
English - Italian	28	**100.0**	**100.0**	**100.0**	**100.0**	96.4	**100.0**	96.4	96.4	**100.0**	**100.0**	96.4	75.0
English - Japanese	28	96.4	89.3	**100.0**	96.4	75.0	96.4	78.6	96.4	96.4	67.9	78.6	46.4
Russian - English	27	92.6	**100.0**	**100.0**	88.9	85.2	81.5	66.7	59.3	88.9	40.7	74.1	59.3
English - Spanish	25	92.0	88.0	88.0	**96.0**	76.0	92.0	92.0	88.0	92.0	84.0	72.0	84.0

Table 9: Each row represents accuracy of system pairs for given language pair. We list language pairs with at least 20 system pairs. Results are calculated over a set of significantly different system pairs with alpha level 0.05. Results with grey background are considered to be tied with the best metric. Interestingly, when we investigated Polish–English results, we found out the test set is likely post-edited MT output.

Just Ask!
Evaluating Machine Translation by Asking and Answering Questions

Mateusz Krubiński[1], Erfan Ghadery[2], Marie-Francine Moens[2], and Pavel Pecina[1]

[1]Charles University Faculty of Mathematics and Physics
{krubinski,pecina}@ufal.mff.cuni.cz
[2]KU Leuven, Department of Computer Science
{erfan.ghadery,sien.moens}@kuleuven.be

Abstract

In this paper, we show that automatically-generated questions and answers can be used to evaluate the quality of Machine Translation systems. Building on recent work on the evaluation of abstractive text summarization, we propose a new metric for system-level Machine Translation evaluation, compare it with other state-of-the-art solutions, and show its robustness by conducting experiments for various translation directions.

1 Introduction

The goal of automatic Machine Translation (MT) evaluation is to automatically evaluate the output quality produced by MT systems. Metrics used for this task assign a score by comparing the MT output to either a reference translation or to the source sentence (the latter is called Quality Estimation).

The main indicator that is used to assess the performance of a specific metric is the correlation with human judgement computed for outputs from several systems. It was recently shown that metrics based on contextualized embeddings, such as YiSi (Lo, 2019) or ESIM (Mathur et al., 2019), are able to achieve better performance than the most widely used BLEU (Papineni et al., 2002).

In this paper, we propose a new method for automatic evaluation of MT systems – MTEQA[1] (**M**achine **T**ranslation **E**valuation with **Q**uestion **A**nswering), building on previous works on evaluating abstractive summaries. We build upon the fact that state-of-the-art (neural) MT systems tend to produce a fluent output but sometimes fail in adequacy of the translation. We leverage the recent progress in Question Generation (QG) and Question Answering (QA) to formulate and answer human readable questions about the MT system output. Our experiments show that the effectiveness of the proposed metric is comparable to performance of other automatic metrics, while considering only a certain amount of information from the whole translation. We also examine the robustness of the metric by considering several translation directions and target languages.

The remainder of this paper is structured as follows. In Section 2, we introduce relevant research on question-based evaluation. In Section 3, we describe our metric in detail. In Section 4, we present and discuss the results of our experiments including s the influence of different human scoring methods. Section 5 presents conclusions.

2 Related Work

Metrics that are most widely used for automatic evaluation of MT outputs produce a score by comparing surface-level forms of hypothesis and reference translation. The most dominant one, BLEU (Papineni et al., 2002), is a version of n-gram precision calculated by averaging over different values of n with penalization for overly short translations (brevity penalty). Another one, CHRF (Popović, 2015), considers the character-level n-grams, making it possible to reward partial token matches. The standardised implementation provided in the sacreBLEU[2] package takes care of pre-processing and enables direct comparison between MT outputs.

Recently, various works (e.g., Lo, 2019; Mathur et al., 2019; Bawden et al., 2020) explored the usage of contextualized word-level or sentence-level embeddings to compare the numerical representations of reference and hypothesis. Such metrics enable explicit regression towards the desired human-produced labels.

2.1 Evaluation of Summarization

The task of automatic text summarization is to produce a concise summary of a given document that

[1]https://github.com/ufal/MTEQA

[2]https://github.com/mjpost/sacrebleu

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 495–506
November 10–11, 2021.

would preserve all the key information from the document. One of the most popular metrics used for evaluating summary quality is ROUGE (Lin, 2004), which compares overlapping n-grams between the model output and the reference summary.

To step beyond the n-grams comparison, Eyal et al. (2019) proposed the APES metric. They used the reference summary to produce fill-in-the-blank type of questions by finding all possible entities using a NER system. The APES score for a given summarization model is the percentage of questions that were answered correctly (using an Question Answering system), averaged over the whole test-set. The authors reported a higher correlation with the Pyramid method (Nenkova et al., 2007) for manual evaluation than the ROUGE metric. Scialom et al. (2019) extended their work into unsupervised settings by generating questions from the source document. Closest to our work are the metrics FEQA (Durmus et al., 2020) and QAGS (Wang et al., 2020), which automatically generate the natural language questions from the summary and/or document.

2.2 Question-based Evaluation of MT

Tomita et al. (1993) were the first to use the reading comprehension tests to measure the quality of MT systems. They translated several passages from TOEFL (Test of English as a Foreign Language) guide book into Japanese, using a selection of MT systems, while corresponding questions and answers were translated into Japanese by professional translators. The MT systems were evaluated by measuring the percentage of questions answered correctly by the Japanese speaking human annotators, using the MT output as a context.

Fuji et al. (2001) used the reading comprehension tests to examine the "usefulness" of machine-translated text. In their experiment, participants take the reading comprehension test in a foreign language (English), while also being presented with the text translated by the MT system into their mother language (Japanese). Authors claim that presenting the MT output yields a higher comprehension performance.

Castilho and Guerberof Arenas (2018) examine the user satisfaction when completing the comprehension type of test, using the context translated by the MT system. They collect the eye-tracking data to analyse the cognitive effort of the participants.

Scarton and Specia (2016) approached the problem of document-level Quality Estimation (QE) by extending the CREG corpus (Ott et al., 2012) of German documents designed for reading comprehension exercises. They use professional translators to translate the questions and answers to English. They examine the document-level translation quality by translating the documents by MT systems and asking the human annotators to complete the reading comprehension test using the MT output as a context. Forcada et al. (2018) used the same corpus to examine the usage of automatically generated gap-filling closure type of testing.

Berka et al. (2011) used the *yes/no* type of questions for manual evaluation of MT systems, examining the English-to-Czech direction. The authors prepared a set of English texts from various domains and used human annotators to come up with three content-based question-answer pairs in Czech for each of the texts. In the next step, the annotators were given the outputs from MT systems (in Czech) and were tasked to answer the questions using the corresponding translation as the context. For each system, the percentage of properly answered questions was measured.

We believe no prior work examines the usage of automatically generated questions and answers to assess the quality of MT systems.

2.3 Keyphrase Extraction

Keyphrases are representative and characteristic phrases from a text that express the key aspects of its content (Papagiannopoulou and Tsoumakas, 2020). In our work, keyphrases play the role of answers, i.e., the pieces of information which we test to be preserved in translation.

In recent years, a wide range of supervised and unsupervised keyphrase extraction methods have been proposed. Unsupervised methods normally perform two main steps to extract keyphrases: 1) select candidate phrases based on some heuristics such as matching with a specific part-of-speech pattern; 2) rank the candidates and select the top ones. Various approaches have been proposed to address this problem such as statistics-based (Won et al., 2019), graph-based (Mihalcea and Tarau, 2004), topic models-based (Liu et al., 2010), and language model-based (Tomokiyo and Hurst, 2003) methods.

On the other hand, supervised methods are relying on labeled data in which keyphrases are annotated in the documents. Supervised methods

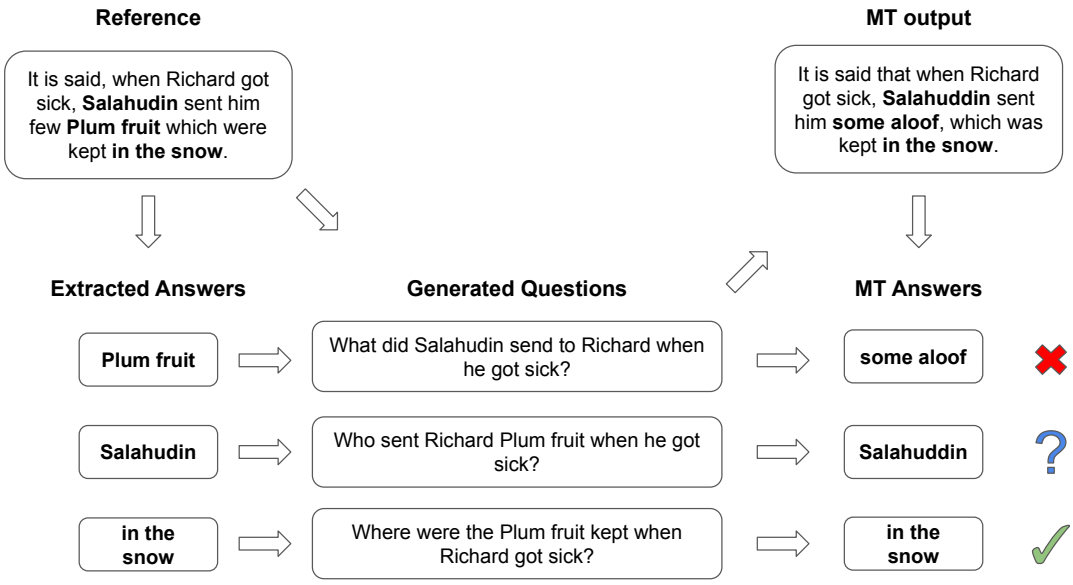


Figure 1: An illustration of the MTEQA pipeline. One of the MT answers is clearly wrong, one is correct but the other differs with just a single character, raising a question about the choice of the answer-comparison metric.

generally model the keyphrase extraction problem as binary classification to predict whether a candidate phrase is a keyphrase or not (Wang and Li, 2017), learning to rank to learn a ranking function that sorts the candidate phrases based on their score (Zhang et al., 2017), and sequence labeling problem (Zhang et al., 2016).

3 MTEQA

Our idea of evaluating MT quality by asking and answering questions is based on the assumption that a good translation should preserve all of the key information that one can extract from the reference. We propose to use a question answering framework as the proxy to measure this.

To check whether a piece of information is preserved, we automatically generate pairs of a question and its (gold-standard) answer from the reference translation and employ a question answering system to provide a new (test) answer given the question and the MT output (translation) used as the context. The generated (test) answer is then compared to the gold-standard answer.

We assume that if it was possible to answer a question looking only at the reference, it should also be possible to answer this question looking only at the MT output and that the two answers should be identical or very similar.

In principle, the proposed MTEQA metric requires solving the following tasks:

1) Answer extraction identifies the key information in a sentence (keyphrases) which should be also present in the MT output. This extraction can be treated in a hierarchical/nested manner. For instance, given the sentence *"Today for dinner I had an organic pasta with garlic."*, the question *"What did you have for dinner today?"* can be correctly answered by all the following phrases *pasta*, *organic pasta* and *organic pasta with garlic*. Thus, answer extraction is performed first and the questions are generated afterwards for each of the answers independently. The same question can be paired with multiple (nested) answers which allows capturing a partial correspondence.

2) Question generation, given a reference translation, produces a human readable question, for which a given keyphrase is the correct answer. For each of the extracted answers, each question is generated independently from the other answers.

3) Question answering generates an answer, given a natural language question and a sentence used as a context. Since we assume that the MT output should carry enough information to answer any question asked based on the reference, we do not consider the non-answerable questions.

4) Answer comparison assesses to what extent the generated answer is correct, given the gold-

Pattern	Extracted Answer	Sentence
NOUN	Coldplay	... the British rock group Coldplay with special guest performers ...
ADJ NOUN	natural grass	As is customary for Super Bowl games played at natural grass stadiums ...
DET NOUN	a fumble	... including a fumble which they recovered for a touchdown ...
NUM NOUN	10 times	The South Florida/Miami area has previously hosted the event 10 times ...
PROPN PROPN	Carolina Panthers	... the National Football Conference (NFC) champion Carolina Panthers ...
DET ADJ NOUN	A professional fundraiser	A professional fundraiser will aid in finding business sponsors ...
DET VERB NOUN	a broken arm	... went down with a broken arm in the NFC Championship Game ...
NUM PUNCT NUM	15–1	The Panthers finished the regular season with a 15–1 record ...
DET NOUN ADP NOUN	the application of electricity	Tesla theorized that the application of electricity to the brain ...

Table 1: Examples of the most frequent POS patterns of gold-standard answers in the XQuAD dataset.

	BLEU	ROUGE-L	F1
Question Answering	-	-	90.27
Question Generation	21.01	43.25	-

Table 2: Performance of the baseline model used in our experiments on the development set of SQuADv1.

standard answer extracted from the reference. Metrics based on exact match should be avoided because they are too strict. For example, given the gold-standard answer *"Tchaikovsky"*, both the *"Tchaikovski"* and *"Beethoven"* would get the same score.

3.1 Scoring Procedure

The entire procedure of MTEQA is illustrated in Figure 1. Formally, for a given segment s_i, reference translation r_i and MT system output t_i, it proceeds as follows:

1. Generate the gold-standard answers $a_{i1}, a_{i2}, \ldots, a_{ik}$ from the reference r_i
2. For each answer a_{ij} and reference r_i, generate a natural language question q_{ij}
3. Answer each question q_{ij} using the MT output t_i as a context, obtaining answer $\tilde{a}_{ij}$
4. The final score for a given translation of a segment s_i, is the average over all generated questions:

$$MTEQA(t_i) = \frac{\sum_1^k D(a_{ij}, \tilde{a}_{ij})}{k},$$

where $D(\cdot, \cdot)$ is a string-comparison metric used to compare the two answers and k is the number of gold-standard answers extracted from the reference.

For the task of comparing MT systems on the entire test-set (i.e. system-level comparison) or at the document-level, we simply report the average of the segment-level scores. When more than one reference $\hat{r}_i$ is available for a given segment, we can use it to generate additional questions and answers.

3.2 Baseline Implementation

Our implementation of the proposed MTEQA metric is based on the state-of-the-art system capable of solving the initial three tasks of the procedure: answer extraction, question generation, question answering. It is the T5 model (Raffel et al., 2020) fine-tuned on the SQuADv1 dataset (Rajpurkar et al., 2016) by Patil (2020) and available from GitHub[3]. Performance on the development set of SQuADv1 in Table 2. We report word-level F1 for question answering and BLEU and ROUGE-L for question generation.

The SQuAD dataset was created manually by tasking the crowd-workers to create up to five questions-answer pairs from a single paragraph from Wikipedia. While the crowd-workers were encouraged to formulate the questions in their own words, the answers were restricted to be continuous sub-sequences of words from the given paragraph. In MTEQA, the answers generated by this model are also continuous sub-sequences of words from the reference and test translations.

The same system is also used for question answering and question generation by prompting the model with a different initial token in the input – for Question Answering:

```
"question: {question_text}
 context: {context_text}"
```

for Question Generation:

```
"answer: {answer_text}
 context: {context_text}".
```

3.3 Generating Additional Answers

Since the QG system generates a single question for each sub-sequence of words marked as an extracted answer, the limit factor is the number of gold-standard answers we extract. To generate more questions, we need more keyphrases to formulate a question about.

[3] https://github.com/patil-suraj/question_generation

	cs-en 12	de-en 12	zh-en 16	avg	en-de 14	en-cs 12
MTEQA F1	0.782*	0.997*	0.952*	0.893*	0.946*	0.845*
MTEQA CHRF KEYPHRASE	**0.890***	**0.998***	0.951*	**0.905***	0.952*	0.859*
SENTBLEU	0.844	0.978	0.948	0.859	0.934	0.840
BLEU	0.851	0.985	0.956	0.854	0.928	0.825
PRISM	0.818	**0.998**	0.957	0.880	**0.958**	**0.949**
YISI-2	0.764	0.988	**0.964**	0.821	0.899	0.714

Table 3: System-level Pearson correlation for selected metrics used for measuring MT quality with DA human assessment over MT systems using the *newstest2020* references. Average (avg) is computed over all to-English directions available. Number below the language pair indicates the number of systems considered. Figures without * are taken from Mathur et al. (2020a).

Considering the whole predictive power of our metric is based on questions, we propose two methods of generating additional questions.

1) We exploit the MT output as an additional source of question/answer pairs. After following the standard procedure, we swap the roles of MT output and reference – we generate gold-standard answers and questions from the MT output, and use reference as a context to answer it. As a final score we take the sum of the two scores.

2) We add keyphrases extracted by linguistic processing of the sentences based on Part-of-Speech (POS) pattern matching and Named Entity Recognition (NER). Given a sentence as the input, first, we parse the sentence using UDPipe (Straka et al., 2016) to extract part of speech (POS) tags. Then, we extract phrases that are matched with one of the patterns in our POS pattern bank. The POS pattern bank is created by parsing the sentences from XQuAD (Artetxe et al., 2020) dataset, extracting the POS patterns corresponding to the gold-standard answers, and taking the most frequent patterns. This dataset contains professional translations of the development set of SQuADv1, translated into various languages from different language families and using different scripts. Table 1 shows some examples of the extracted POS patterns. Second, we extract named entities mentioned in the input sentence using a combination of two multilingual NER models, POLYGLOT-NER (Al-Rfou et al., 2015), and Stanza (Qi et al., 2020). Finally, we output the union of the extracted phrases and named entities as the potential answers.

3.4 Choice of the $D(\cdot, \cdot)$ Metric

As already pointed, selection of the $D(\cdot, \cdot)$ might be crucial for optimal performance of the proposed metric and thus we consider several options. Motivated by QA evaluation, we employ the word-level F1 (Rajpurkar et al., 2016; Trischler et al., 2017; Chen et al., 2019; Durmus et al., 2020). Motivated by MT evaluation we also consider the BLEU (Papineni et al., 2002) metric and the CHRF (Popović, 2015) metric. Finally we also employ "exact match" (Rajpurkar et al., 2016) score, mainly for comparison. All of the metrics we use operate on a surface level and assign a similarity score for a pair of strings. In the future, it may be worth to explore e.g. cosine similarity between word embeddings.

4 Experiments

We evaluate the proposed MTEQA metric using the submissions to the WMT20 News translation task (Barrault et al., 2020) and their (direct) human assessments (DA). For each of the MT systems participating in the task, we compute a single score as the average of segment-level scores and report the system-level Pearson correlation with the human assessment. We report individual results for selected translation directions into English plus aggregated results (averages) for all to-English directions which were part of the WTM20 Metric Task (Mathur et al., 2020b) evaluation campaign[4].

4.1 Baseline

The baseline implementation is described in Section 3. It is based on the T5 model tuned on the SQuADv1 dataset and used to generate: 1) the gold-standard answers from the reference translations, 2) a question for each gold-standard answer, 3) a test answer for each question and MT output (context) pair. The test answers are compared by the word-level F1 score (Section 3.4).

The results of this system are shown in Table 3 labeled as MTEQA F1 together with other metrics for comparison. We experiment with the to-English direction, since the SQuADv1 dataset used for fine-tuning is in English. On average, the baseline

[4] cs, de, ja, pl, ru, ta, zh, iu, km, ps → en

	cs-en	de-en	zh-en	ja-en	ru-en	ps-en	avg	en-de 14	en-cs 12
MTEQA F1	0.782	0.997	0.952	0.982	0.908	0.982	0.893	0.946	0.845
MTEQA CHRF	0.796	0.996	**0.959**	0.982	0.901	0.980	0.887	0.950	0.815
MTEQA BLEU	0.762	0.998	0.954	**0.983**	0.925	0.985	0.894	**0.957**	0.840
MTEQA EXACT	0.762	0.998	0.954	0.966	0.910	0.986	0.883	0.950	0.874
MTEQA F1 OUT	0.808	0.998	0.949	0.980	0.917	0.984	0.891	-	-
MTEQA CHRF OUT	0.835	0.997	0.957	0.979	0.910	0.986	0.891	-	-
MTEQA BLEU OUT	0.809	0.998	0.950	0.981	0.929	0.984	0.896	-	-
MTEQA EXACT OUT	0.827	**0.999**	0.948	0.969	0.902	0.983	0.884	-	-
MTEQA F1 KEYPHRASE	0.851	0.998	0.944	0.978	0.930	0.986	0.896	0.941	0.877
MTEQA CHRF KEYPHRASE	**0.890**	0.998	0.951	0.978	0.927	0.981	**0.905**	0.952	0.859
MTEQA BLEU KEYPHRASE	0.844	0.998	0.939	0.973	**0.945**	0.991	0.900	0.943	0.873
MTEQA EXACT KEYPHRASE	0.858	0.997	0.938	0.959	0.936	0.990	0.893	0.948	**0.915**
MTEQA F1 OUT KEYPHRASE	0.831	0.998	0.942	0.978	0.914	**0.992**	0.893	-	-
MTEQA CHRF OUT KEYPHRASE	0.851	0.998	0.947	0.977	0.917	0.990	0.902	-	-
MTEQA BLEU OUT KEYPHRASE	0.842	0.998	0.938	0.971	0.913	0.990	0.895	-	-
MTEQA EXACT OUT KEYPHRASE	0.838	0.998	0.936	0.960	0.918	**0.992**	0.887	-	-

Table 4: System-level Pearson correlation for various variants of the proposed metric with DA human assessment over MT systems using the *newstest2020* references. Average is computed over all to-English directions available.

outperforms the traditional MT evaluation metrics (SENTBLEU, BLEU) as well as the recently proposed ones that performed very well in the WTM20 Metric Task (PRISM (Thompson and Post, 2020), YISI-2), though for some of the translation directions (e.g. Czech-English) MTEQA F1 is much worse (but for Czech-English, YISI-2 also does not beat BLEU).

4.2 Variants of the $D(\cdot, \cdot)$ metric

To assess the effect of choice of the $D(\cdot, \cdot)$ metric, we modified the baseline to exploit other options (see Section 3.4). The results are shown in the first section of Table 4. Unsurprisingly, the worst results are achieved by MTEQA EXACT which requires exact match of the test answer and the gold-standard one. But overall, the differences here are not large.

4.3 Generating Additional Answers

In general, the T5 model fine-tuned on the SQuADv1 dataset does not generate plentiful question/answer pairs. In fact, the average number of such pairs that are generated for an English sentence is only around two. Table 5 (row *baseline*) presents exact figures from our experiments, i.e., the average numbers of questions generated from a single segment of the *newstest2020* reference files for selected translation directions and the average computed for all directions into English.

To increase the number of question/answer pairs, we implemented the two methods described in Section 3.3 and present the results in Table 4. The systems denoted as OUT exploit question/answer pairs extracted from the references and MT outputs and the systems denoted as KEYPHRASE extract the pairs by POS pattern matching and NER.

The average correlation obtained using the MT output to generate questions (denoted as OUT) was very similar, but slightly worse than the one using just the questions from the reference. However, the method based on POS pattern matching and NER (denoted as KEYPHRASE) yielded improvements over various translation directions and answer comparison methods. The average numbers of question/answer pairs obtained by this method is shown in Table 5. It increased by the factor of 4 (approximately). Together with the CHRF metric used for answer comparison, it forms the best-performing configuration of the proposed metric. We also include its results in Table 3 . From now on, we will report our results using this variant. See Appendix A for examples of usage of different evaluation methods.

4.4 Non-English Reference

So far, all the experiments were conducted for the translations directions into English. This is given by the limitation of the T5 model which was trained on English data and most importantly by the SQuADv1 dataset which was used for fine-tuning and which is in English.

To overcome that, we used the multilingual mT5 model (Xue et al., 2021) and fine-tuned it on machine translation of SQuADv1 dataset into German by Lewis et al. (2020) and into Czech by (Macková and Straka, 2020). The results for English-Czech and English-German are included in both

	cs-en	de-en	ja-en	pl-en	zh-en	avg	en-cs	en-de
BASELINE	2.87	2.75	1.74	1.36	1.65	1.76	1.66	1.41
KEYPHRASE	13.36	12.01	6.66	5.10	8.79	6.98	9.45	8.71

Table 5: Average number of questions generated from a single segment in the *newstest2020* reference file by the baseline system (fine-tuned T5) and the keyphrase extraction method (POS pattern matching and NER). The average is computed over all to-English directions.

Tables 3 and 4. Overall, MTEQA still performs very well. It is better than the traditional metrics (SENTBLEU, BLEU) and also YISI-2 and comparable with PRISM for English-German. However, it is substantially worse than PRISM for English-Czech. Given the fact, that the system is multilingual and fine-tuned on machine-translated data, the results are encouraging and open doors for a cross-lingual setting which would not require reference translations.

4.5 Comparison with MQM Scores

Recently, Freitag et al. (2021) demonstrated that the WMT DA method traditionally used for human evaluations has actually lower correlation with expert-based labels than the Multidimensional Quality Metrics (MQM) scoring method developed in the EU QTLaunchPad and QT21 projects.

To provide a more complete picture of the performance of the proposed MTEQA metric, we also report correlation with the MQM assessments. Table 6 presents the system-level Pearson correlation of the proposed metric with both the MQM and DA labels for 8 systems that were re-annotated by Freitag et al. (2021) and are available from GitHub[5].

The results are surprising and to a large extent unintuitive. Metrics performing well in comparison with MQM are bad in comparison with DA. This issue was already discussed by Freitag et al. (2021) and we leave deeper analysis of the difference for the future when MQM labels will be available for more data and for more translation directions.

5 Conclusions

In this paper we introduced a new metric for automatic evaluation of Machine Translation systems. We showed that the degree to which the MT output can be used to answer questions about the reference can be used as a proxy to evaluate the translation quality. We proved that our metric is robust by conducting experiments over multiple translation directions.

We examined a linguistically motivated way of extracting key phrases from the sentence and showed that it boosts the final performance. We checked the influence of various word-level comparison metrics used to compare the test and gold-standard answers, and reported how it affects the correlation with human scores. In our work, we focused on translation directions into English. The only limiting factor in applying our metric to other translation directions is the availability of Question Generation and Question Answering systems in a given language. However, automatic translation of SQuAD can be an effective way to obtain data for training such systems.

Finally, we examined the performance against the MQM labels and compared the performance against the DA labels. While for the DA labels our metric performs close to state-of-the-art solutions, for the MQM labels there is a noticeable drop in performance.

In the future, we plan to examine the cross-lingual approach – instead of generating questions and answers from the reference, one may instead use the source directly.

[5] https://github.com/google/wmt-mqm-human-evaluation

Acknowledgements

This work was supported by the European Commission via its H2020 Program (contract no. 870930) and CELSA (project no. 19/018), and has been using data and tools provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (project no. LM2018101).

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama.

	zh-en		en-de	
	MQM	DA	MQM	DA
MTEQA CHRF KEYPHRASE	0.630	**0.818**	0.761	0.394
PRISM	0.778	0.351	**0.989**	0.607
COMET	**0.889**	0.188	0.965	**0.628**
PARBLEU	0.380	0.565	0.722	0.218
CHRF	0.523	0.579	0.853	0.576
TER	0.352	0.511	0.810	0.477

Table 6: System-level Pearson correlation for selected metrics used for measuring MT quality with the DA and MQM labels, computed for the *newstest2020* references and the 8 MT systems re-annotated by Freitag et al. (2021).

2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Rachel Bawden, Biao Zhang, Andre Tättar, and Matt Post. 2020. ParBLEU: Augmenting metrics with automatic paraphrases for the WMT'20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 887–894, Online. Association for Computational Linguistics.

Jan Berka, Martin Černý, and Ondřej Bojar. 2011. Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95.

Sheila Castilho and Ana Guerberof Arenas. 2018. Reading comprehension of machine translation output: What makes for a better read? In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 79–88, Alacant/Alicante, Spain. European Association for Machine Translation.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Mikel L. Forcada, Carolina Scarton, Lucia Specia, Barry Haddow, and Alexandra Birch. 2018. Exploring gap filling as a cheaper alternative to reading comprehension questionnaires when evaluating machine translation for gisting. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 192–203, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *arXiv preprint arXiv:2104.14478*.

Masaru Fuji, Hatanaka N, Ito E, Kamei S, Kumai H, Sukehiro T, Yoshimi T, and Isahara Hitoshi. 2001. Evaluation method for determining groups of users who find mt useful. In *MT Summit VIII: Machine Translation in the Information Age*, pages 103–108.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 366–376.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with

different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Kateřina Macková and Milan Straka. 2020. Reading comprehension in czech via machine translation and cross-lingual transfer. In *23rd International Conference on Text, Speech and Dialogue*, pages 171–179, Cham, Switzerland. Springer.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020a. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4–es.

Niels Ott, Ramon Ziai, and Detmar Meurers. 2012. Creation and analysis of a reading comprehension exercise corpus. *Multilingual corpora and multilingual corpus analysis*, 14:47.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2020. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1339.

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.

Suraj Patil. 2020. Question generation. https://github.com/patil-suraj/question_generation.

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. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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'16)*, pages 3652–3658, Portorož, Slovenia. European Language Resources Association (ELRA).

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Brian Thompson and Matt Post. 2020. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Masaru Tomita, Shirai Masako, Tsutsumi Junya, Matsumura Miki, and Yoshikawa Yuki. 1993. Evaluation of mt systems by toefl. In *Proceedings of the*

Theoretical and Methodological Implications of Machine Translation (TMI-93), pages 252–265.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, pages 33–40.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Liang Wang and Sujian Li. 2017. Pku_icl at semeval-2017 task 10: Keyphrase extraction with model ensemble and external knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 934–937.

Miguel Won, Bruno Martins, and Filipa Raimundo. 2019. Automatic extraction of relevant keyphrases for the study of issue competition. In *Proceedings of the 20th international conference on computational linguistics and intelligent text processing, Berkeley, La Rochelle, France*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuan-Jing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 836–845.

Yuxiang Zhang, Yaocheng Chang, Xiaoqing Liu, Sujatha Das Gollapalli, Xiaoli Li, and Chunjing Xiao. 2017. Mike: keyphrase extraction by integrating multidimensional information. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1349–1358.

A Appendix

A.1 Answer extraction

Below we show the difference in the answer extraction process using the baseline approach as opposed to the proposed method based on POS patterns and NER tags. In both cases the same system is used for question generation.

Answer	Question
Answers extracted using the method based on POS sequences and NER tags	
the stadium	Where did the cat fall from?
an American football match	At what event did spectators catch a cat?
upper deck	What part of the stadium did the cat fall from?
A cat	What animal was caught by spectators at an American football match in Miami Gardens?
Florida	Where is Miami Gardens located?
spectators	Who caught a cat at an American football match in Miami Gardens?
Miami Gardens	Where was a cat caught by spectators at an American football match?
Answers extracted using the baseline model	
cat	What animal was caught by spectators at a football match in Miami Gardens?
Miami Gardens	Where was a cat caught by spectators at an American football match?

Table 7: Extracted keyphrases and generated corresponding questions for the sentence: `A cat was caught by spectators at an American football match in Miami Gardens, Florida, after it fell from the stadium's upper deck.`

Answer	Question
Answers extracted using the method based on POS sequences and NER tags	
Liberal	What party did Ed Davey belong to?'
vaccine passports	What did Ed Davey call 'divisive, unworkable and expensive'?
opposition	What type of opposition was there on the Covid Recovery Group?
the Covid Recovery Group	What group did Tory MPs oppose?
Ed Davey	Which Liberal Democrat leader called vaccine passports 'divisive, unworkable and expensive'?
Tory	What political party opposed vaccine passports?
leader	Who is Ed Davey?
Democrats	Along with Tory MPs, what party opposed vaccine passports?
Answers extracted using the baseline model	
Ed Davey	Which Liberal Democrat leader called vaccine passports 'divisive, unworkable and expensive'?
vaccine passports	What did Ed Davey call 'divisive, unworkable and expensive'

Table 8: Extracted keyphrases and generated corresponding questions for the sentence: `There had been opposition from Tory MPs on the Covid Recovery Group as well as the Liberal Democrats, whose leader Ed Davey called vaccine passports 'divisive, unworkable and expensive'.`

Answer	Question
Answers extracted using the method based on POS sequences and NER tags	
russischen	Welche Nationalität sind die Pelmeni?
Pelmeni	Wie ist der russische Name für Pirggen?
Piroggen	Was wird manchmal mit gebrannten Zwiebeln angerichtet?
gebratenen Zwiebeln	Mit welchen Arten von Zwiebeln werden die russischen Pelmeni angerichtet?
Answers extracted using the baseline model	
-	-

Table 9: Extracted keyphrases and generated corresponding questions for the sentence: `Ähnlich wie die russischen Pelmeni werden Piroggen manchmal mit gebratenen Zwiebeln angerichtet.`

A.2 Answer comparison

Below we show the difference between gold-standard answers extracted from the reference and test answers obtained with the Question Answering system, using the MT output as context.

Question	Gold-standard Answer	Test Answer
MT Output: The men‘s 100 metres semi-final begins at Sunnybrown Haquim (left).		
In what distance is Sani Brown Hakim in the men‘s semifinals?	100m	100 metres
Who is Sani Brown Hakim in the 100m semifinals?	the men	Sunnybrown Haquim
Who started in the men‘s 100m semifinals?	Sani Brown Hakim	Sunnybrown Haquim
MT Output: Sani Brown Hakeem (left) will start the men‘s 100 metres semi-final.		
In what distance is Sani Brown Hakim in the men‘s semifinals?	100m	100 metres
Who is Sani Brown Hakim in the 100m semifinals?	the men	Sani Brown Hakeem
Who started in the men‘s 100m semifinals?	Sani Brown Hakim	Sani Brown Hakeem

Table 10: Extracted keyphrases, generated corresponding questions and answers extracted from MT output for the reference: `Sani Brown Hakim (left) starting in the men‘s 100m semifinal.`

Question	Gold-standard Answer	Test Answer
MT Output: Recently I flew from Moscow, where I was trained," Andrei Borovikoff said.		
Who said that he flew from Moscow to study?	Andrei Borovikov	Andrei Borovikoff
Where was I studying?	Moscow	Moscow
MT Output: Recently, I flew from Moscow, where he was trained ", Andrey Borovikov told.		
Who said that he flew from Moscow to study?	Andrei Borovikov	Andrey Borovikov
Where was I studying?	Moscow	Moscow

Table 11: Extracted keyphrases, generated corresponding questions and answers extracted from MT output for the reference: `Recently I flew from Moscow where I was studying," said Andrei Borovikov.`

A Fine-Grained Analysis of BERTScore

Michael Hanna
Charles University, MFF UFAL
hannami@o365.cuni.cz

Ondřej Bojar
Charles University, MFF UFAL
bojar@ufal.mff.cuni.cz

Abstract

BERTScore (Zhang et al., 2020), a recently proposed automatic metric for machine translation quality, uses BERT (Devlin et al., 2019), a large pre-trained language model to evaluate candidate translations with respect to a gold translation. Taking advantage of BERT's semantic and syntactic abilities, BERTScore seeks to avoid the flaws of earlier approaches like BLEU, instead scoring candidate translations based on their semantic similarity to the gold sentence. However, BERT is not infallible; while its performance on NLP tasks set a new state of the art in general, studies of specific syntactic and semantic phenomena have shown where BERT's performance deviates from that of humans more generally.

This naturally raises the questions we address in this paper: what are the strengths and weaknesses of BERTScore? Do they relate to known weaknesses on the part of BERT? We find that while BERTScore can detect when a candidate differs from a reference in important content words, it is less sensitive to smaller errors, especially if the candidate is lexically or stylistically similar to the reference.

1 Introduction

While manual, human evaluation of machine translation (MT) systems is still the gold standard, automatic evaluation metrics have long been used for their relative speed and inexpensiveness. Early automatic metrics were easy to implement and somewhat correlated with human judgements, but have clear limitations: BLEU (Papineni et al., 2002) relies on n-gram overlap, and is thus not robust to differing word order or choice. In contrast, METEOR (Lavie and Agarwal, 2007) requires training, but depends on token alignment, which is also a fraught task.

With the advent of deep learning, new automatic metrics have arisen, both in response to and making use of the technical advances brought by deep learning. In particular, metrics like COMET (Rei et al., 2020) and BERTScore use large pre-trained language models (LLMs) to generate scores for candidate sentences. The use of these LLMs allows for metrics that take advantage of the linguistic capabilities of these LLMs, and no longer rely solely on surface-level features such as n-grams.

The expressiveness of these models is both a boon and a danger. While they can (and do, based on correlation with human judgments) generate more useful scores for translations, how they arrive at the score, and which types of sentences they will score accurately is not immediately obvious.

Moreover, these LLMs are known to have flaws. BERT in particular has been shown to be, in certain scenarios, insensitive to negation (Ettinger, 2020) and word order (Pham et al., 2020). BERT also has inexact representations of numbers (Wallace et al., 2019) and fails to be robust to named entities (Balasubramanian et al., 2020). All of these phenomena could result in poor-quality scores from BERTScore. However, it is difficult to say for certain how these issues might manifest in BERTScore, as it employs BERT in an unsupervised scenario distinct from that of these analyses.

Thus, in this paper, we analyze BERTScore. We first formally define desiderata for a MT metric. Then, we consider how BERTScore fulfills these requirements under conditions of interest. We find that BERTScore violates some of these requirements, specifically the requirement that incorrect translations be rated below correct ones; this occurs most often when the incorrect translation is lexically similar to the reference, or especially if the difference is only in function (not content) words.

2 Desiderata for MT metric quality

The most common method of measuring the quality of a MT metric is correlation with human judgments (Fomicheva and Specia, 2019); however, these correlations provide little information regard-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 507–517
November 10–11, 2021.

ing when and why an MT metric differs from human judgment. In this paper, we consider three ways of examining MT metric quality, with the aim of determining the failure cases of MT metrics.

In all of our experiments, we assume the following setup. We have a MT metric, which we take to be a function M that takes as input a reference and candidate translation, and outputs a score in $[0, 1]$. We also have a dataset $\mathcal{D}$, consisting of triples (x, Y, B) where x is a source sentence, Y is a list of at least two reference translations, and B is a list of at least one "bad" translation, which contains errors.

Then we state that a good MT metric[1] M fulfills, for any triple $(x, Y, B) \in \mathcal{D}$, where $Y = \{y_1, y_2, \ldots, y_n\}$, and $B = \{b_1, b_2, \ldots, b_m\}$, the following conditions:

(i) For any pair (y, y') of reference translations from Y, $M(y, y') \approx M(y, y') \approx 1$.

(ii) For any reference translation $y \in Y$ and candidate translation $b \in B$, $M(y, b) < 1$ and $M(b, y) < 1$. It follows that given another reference translation $y' \in Y$, we should have that $M(y, b) < M(y, y')$ and so on.

(iii) If we know the relative quality of the bad translations in B, let B be a list sorted in decreasing order of translation quality, such that b_1 is better than b_2, and so on. Then for any reference translation y, and bad translations b_i, b_j from B, where $i < j$, $M(y, b_i) > M(y, b_j)$.

Put simply, (i) reference translations should be scored near 1 when compared to each other, (ii) bad candidate translations should be scored worse than reference translations, and (iii) the scoring of bad candidate translations should reflect their relative quality.

To use this framework to investigate the failure points of MT metrics, we simply need a dataset that contains phenomena of interest; for example, we might be interested in knowing if a MT metric is able to distinguish between translations that do and do not correctly render negation. Then, we simply compute the quantities discussed in conditions (i) through (iii) for each example, and see which, if any, conditions are violated. If, for example, condition (i) is violated when two equivalent references employ different types of negation, this might imply that our metric is not robust to this sort of negation phenomenon.

Note that these desiderata only concern the scores given to the reference and candidate sentences; the source-language sentence is ignored. This is because we define these desiderata keeping in mind that many MT metrics (including BERTScore) operate only in the target language[2]. We can thus avoid assuming the existence of a source sentence whatsoever, allowing the construction of datasets consisting only of reference and bad candidate translations that exhibit phenomena of interest. However, it may be desirable to use real translations, so that the dataset reflects the distribution of real-world translation errors.

3 BERTScore and BERT

3.1 BERTScore

In this paper, our metric of interest is BERTScore (Zhang et al., 2020). To compute BERTScore, we first feed a reference and candidate translation for a given sentence into BERT, and retrieve their token level vector representations. Let z be the representations of the reference and $\hat{z}$ those of the candidate. Then we compute the precision and recall metrics for BERTScore by comparing each token representation z_i of the reference translation to each token representation $\hat{z}_j$ of the candidate translation as follows:

$$P_{BERT} = \frac{1}{|\hat{z}|} \sum_{\hat{z}_j \in \hat{z}} \max_{z_i \in z} z_i^\top \hat{z}_j$$

$$R_{BERT} = \frac{1}{|z|} \sum_{z_i \in z} \max_{\hat{z}_j \in \hat{z}} z_i^\top \hat{z}_j$$

The F_1 score can be defined as usual. As BERTScore can range from -1 to 1, but most often inhabits the upper end of that range, its creators suggest the use of baseline scaling, which generally leaves BERTScore in the range [0,1], as desired for use with our prior formalization. Baseline rescaling is performed for P_{BERT} as

$$\hat{P}_{BERT} = \frac{P_{BERT} - a}{1 - a}$$

and likewise for R_{BERT}; a is an empirical lower bound on observed BERTScore.

[1] We assume WLOG that the metric's scores are normalized such that better translations receive higher mean scores, and equally good candidates receive a score near 1.

[2] In the long term, explicit inclusion of the source sentence in MT evaluation would be useful but that is not the concern of this work.

The form of BERTScore naturally leads to its interpretation as a similarity-based metric. It penalizes candidates containing words whose representations are not similar to any of the reference's words' representations (precision), and vice-versa (recall). As a result, the quality and characteristics of these representations, derived from BERT, will play a key role in the quality of BERTScore.

3.2 BERT

What, then, is known about BERT, and its syntactic and semantic capabilities? Of the two, it is syntax that BERT is most widely claimed to capture within its internal representations: Hewitt and Manning (2019) use structural probing to find dependency trees in BERT's vector geometry, while Tenney et al. (2019) use probing to find part of speech tags and dependency arc labels, among other types of syntactic information. Analysis of BERT's attention has shown that certain heads attend to not only relevant linguistic units such as determiners of nouns and coreferent mentions (Clark et al., 2019), but also dependency relations (Htut et al., 2019).

However, these analyses of internal representations and the information contained therein occasionally come at odds with targeted evaluations of BERT's syntactic abilities. Despite BERT's supposed knowledge of syntax, its predictions often remain the same, even when its inputs are shuffled (Pham et al., 2020). Moreover, BERT does not seem to understand negation (Ettinger, 2020); this may be due to BERT encoding syntactic information, but not necessarily using it in its predictions (Glavas and Vulić, 2021).

For semantics, the situation is even more complicated. While BERT's performance on natural language understanding tasks set a new state of the art, more targeted tests of its semantic abilities have yielded less positive results. BERT has limited knowledge of lexical semantic relations such as hypernymy (Ravichander et al., 2020) and antonymy (Staliunaite and Iacobacci, 2020). Moreover, it has fragile representations of named entities (Balasubramanian et al., 2020), and imprecise representations of numbers (Wallace et al., 2019). These flaws comprise specific linguistic phenomena that BERTScore, due to its use of BERT, might be unable to handle, and thus merit investigation.

German Source: Ich habe mich konzentriert.
Good Translation 1: I focused.
Good Translation 2: I've been concentrating.
Bad Translation 1: I've focused me.
Bad Translation 2: I have focussed.
Broad phenomenon: Verb Tense / Aspect / Mood
Specific phenomenon: Reflexive - Perfect

Figure 1: Example data from TQ-AutoTest.

4 Experiments

For our experiments, we utilize the framework described in Section 2 to investigate the questions about BERT described in Section 3. As the number of datasets that fit our framework is few, we limit ourselves to three primary experiments.

4.1 TQ-AutoTest

First, we apply our framework to the TQ-AutoTest dataset (Macketanz et al., 2018). Originally used for targeted evaluation of MT systems, it includes German source sentences that each exhibit one of 14 different linguistic phenomena, such as ambiguity, composition, and subordination.

Each example contains a source sentence, annotated with the broader and more specific phenomenon it exhibits, as well as 1-2 reference translations and 0-2 incorrect translations; see Figure 1 for a sample. As the dataset released is small, we filter out phenomena containing fewer than 5 examples.

We can thus test condition (i) by computing the BERTScore between the two references, and verifying it is close to 1. We also test condition (ii) by comparing the BERTScore assigned (with respect to a given reference) to the other reference, and that which is assigned to a bad translation; the former should be greater than the latter. We cannot test condition (iii), as the dataset does not provide multiple bad translations sorted by quality.

4.2 PE^{2}rr

Second, we use the PE^{2}rr dataset (Popović and Arčan, 2016), which is a manually annotated error analysis of MT output. Each example in the dataset consists of a source sentence, one MT output, and two correct translations, along with two error annotations. These annotations are word-level annotations into 8 broadly non-linguistic classes, such as

German Source: Frauen , die in Burkina Faso zu Hexen abgestempelt werden , weisen in der Regel einige gemeinsame gesellschaftliche Merkmale auf .
Original Translation (Annotated): Women in Burkina Faso `[miss]` are branded as witches , usually `[miss]` some common social features .
Post-edit: Women in Burkina Faso who are branded as witches usually have some common social features .
Original Reference: Women declared as witches in Burkina Faso usually have several common characteristics .

Figure 2: Example data from PE^{2}rr. The "`[miss]`" tokens inserted in the original translation correspond to one of the annotator's error notations, indicating a missing word.

"addition", "lexical error", or "untranslated". See Figure 2 for an example.

Despite the difference in annotation type, we apply our framework just as with the prior experiment. For consistency with the prior experiment, we use only the portion of the dataset where the target language is English (i.e., the German-English portion). We also filter out any examples in which the machine translation output is totally correct, as this would leave no bad examples with which to test conditions other than (i). This once more allows us to test conditions (i) and (ii).

4.3 Grammatical Error Correction

Finally, we use two non-MT datasets for grammatical error correction (GEC): the CoNLL 2014 shared task dataset (Ng et al., 2014), as well as additional annotations released by Bryant and Ng (2015). The former consists of non-native English speakers' essays on genetic testing, paired with annotated corrections from two annotators for each sentence. The latter adds additional annotators' corrections for each sentence, yielding 11 in total.

Of special interest in this dataset is the presence of, for each example, an incorrect sentence and 11 sets of error-annotated corrections that can be applied to obtain correct sentences. Thus, for each original, ungrammatical sentence, we have 10 (often distinct) grammatical sentences, whose meaning should be roughly the same; these act as reference sentences. This allows us to test conditions (i) and (ii).

We can also test condition (iii) by using the following moderate assumption: a sentence, originally grammatically incorrect, is more correct if more corrections have been applied to it. That is, applying one or more corrections (from the same annotator) brings an incorrect candidate sentence closer to the shared meaning of the reference sentences. This assumption can be false: sometimes a group of corrections, rather than one alone, is needed to increase the grammaticality of a sentence. But, this assumption allows us to apply an arbitrary number of corrections to an initially incorrect sentence, to generate intermediate incorrect sentences, with controlled, graded, levels of incorrectness.

So, we generate two incorrect candidates by applying different numbers of edits from the same annotator to one original sentence. We then generate a reference sentence by applying all of another annotator's edits. Finally, we calculate the BERTScore of each candidate with respect to this reference; the candidate that received more edits should receive a higher BERTScore. Note that the reference must be generated by a different set of edits; comparing partially-corrected sentences to a correct sentence generated from same edits would make this a trivial string comparison problem. Figure 3 provides an example of this process.

5 Results

In the following section, we detail the experiments performed and their results. In all experiments, the original authors' implementation of BERTScore[3] is used, with default baseline rescaling.

5.1 TQ-Autotest

As discussed in Section 4.1, we test conditions (i) and (ii) with the TQ-Autotest dataset. First, we filter the dataset to include only those examples for which there are at least two good translations (y, y'). Then, to test condition (i), we compute the BERTScore (BERTScore(y, y')) assigned to the pair of good translations, then compute the mean score for each category.

Note that one should not make absolute comparisons between the mean BERTScore assigned to each category and the desired value (1.0) of the mean score. Because BERTScore can easily be rescaled, it is more useful to verify that good translation pairs receive similar scores across categories;

[3] Available at https://github.com/Tiiiger/bert_score

Original: As a result , if the situation keep go on in this unexpected trend , it will cause a bad effect on the young generation .
Corrections (Annotator 2):
(7, 8, 'keeps', subject-verb agreement)
(8, 10, 'following', word choice)
(10, 11, '', preposition)
(17, 18, 'have', word choice)
(23, 24, 'younger', word form)
Original + 1 correction (b_1): As a result , if the situation keeps go on in this unexpected trend , it will cause a bad effect on the young generation .
Original + 4 corrections (b_4): As a result , if the situation keeps following this unexpected trend , it will have a bad effect on the young generation .
Alternate reference (Annotator 7) (y): As a result , if the situation keeps going on in this unexpected trend , it will have a bad effect on younger generations .

Figure 3: Creation of graded ungrammatical sentences from GEC data. Each correction is a tuple of (start index, end index, new text, error made). We apply a correction via string replacement (i.e. in Python: `original[start_index:end_index] = new_text`). Having received more corrections, b_4 should be closer to y than b_1 is. Thus, we should have BERTScore(y, b_4) > BERTScore(y, b_1).

Linguistic Phenomenon	BERTScore (F_1)
LDD & interrogatives	0.849
Composition	0.852
Punctuation	0.864
Function word	0.712
Subordination	0.763
Non-verbal agreement	0.830
Ambiguity	0.804
Verb tense/aspect/mood	0.795
Coordination & ellipsis	0.885
Named entitiy & term.	0.859
MWE	0.874
Average	0.815

Table 1: TQ-AutoTest: Mean BERTScore (F_1) assigned to gold reference, gold candidate pairs, by linguistic phenomenon

Linguistic Phenomenon	Accuracy
LDD & interrogatives	82.4
Composition	60.0
Punctuation	40.0
Function word	42.9
Subordination	75.0
Non-verbal agreement	60.0
Ambiguity	100.0
Verb tense/aspect/mood	73.6
Coordination & ellipsis	80.0
Named entity & terminology	85.7
MWE	100.0
Average	75.5

Table 2: TQ-AutoTest: Average Accuracy by broad linguistic phenomenon. Credit is given when BERTScore(y, y') is greater than BERTScore(y, b), and BERTScore(y, b'), if a b' is provided.

these scores could then easily be rescaled to 1.0. Thus, in this section, we explore only which types of correct sentence pairs BERTScore is more or less able to assign a high score, compared to the average score given to correct sentence pairs.

We see in table Table 1 that the average BERTScore for each category falls near the average. There are some notable exceptions: for example, the "function word" category, which falls well below the mean. This indicates that BERTScore gives different correct translations of the same sentence lower scores, when that sentence contains difficult function words. In contrast, the "coordination & ellipses" and "multi-word error" categories fall well above the mean, suggesting that alternate correct translations of sentences containing these phenomena are scored more highly.

In the second experiment, we test condition (ii) via the following procedure. Once more, we filter the dataset, such that every example includes two good translations and one bad translation, (y, y', b), with potentially another bad translation b' as well. Then, we test the accuracy of BERTScore on these examples. BERTScore is deemed to correctly answer an example if BERTScore(y, y') > BERTScore(y, b). If b' is present, as in 71% of examples, it is also necessary that BERTScore(y, y') > BERTScore(y, b'). We then report the mean accuracy for each category of linguistic phenomenon.

In this second experiment, see Table 2, the differences are more pronounced. In some categories, namely those such as "ambiguity", and "multi-

Error Type	Count	BERTScore (F_1)
reordering	135	0.560
untranslated	55	0.553
lexical	171	0.584
inflection	72	0.571
derivation	16	0.537
missing	164	0.578
contraction	14	0.560
Average	90	0.587

Table 3: PE^{2}rr: Mean BERTScore (F_1) for reference sentence pairs where the original machine translation contains at least one error of the given type.

Error Type	Acc. (Easy)	Acc. (Hard)
reordering	97.0	44.4
untranslated	100.0	72.7
lexical	98.2	46.2
inflection	100.0	50.0
derivation	100.0	50.0
missing	97.0	46.3
contraction	100.0	50.0

Table 4: PE^{2}rr: Average accuracy by error type, for both the easy and hard problem scenarios. Credit is given when BERTScore(y, y') is greater than BERTScore(y, b).

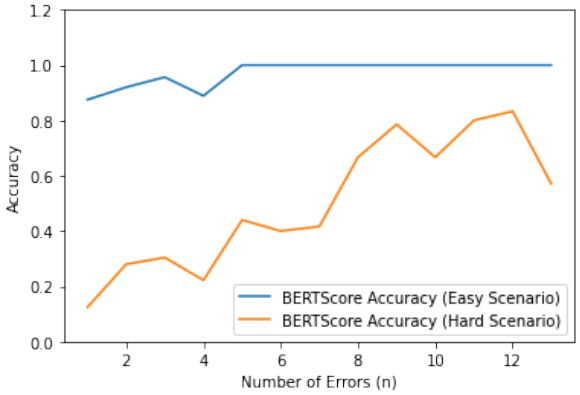


Figure 4: PE^{2}rr: BERTScore accuracy on sentences with n errors, in the easy and hard scenarios.

word error" which are likely to result in totally incorrect word choice (i.e. an obvious lexical difference), BERTScore has a high accuracy. In contrast, BERTScore struggles with difficult punctuation as well as composition errors and function words; notably, the last category also has the lowest score in Table 1. These errors are all somewhat subtle. Errors in function words are by definition not errors in more obvious content words. Similarly, in compositional phenomena like phrasal verbs, an error can appear simply as the omission or incorrect substitution of a mere preposition.

In order to provide context for these results, we also run the second experiment (which corresponds to Table 2) using BLEU score[4] as our metric. As with BERTScore, we mark an example as correct when BLEU(y, y') is greater than BLEU(y, b), and BLEU(y, b'), if a b' is provided. We omit the category-level results for brevity, but find that the mean accuracy for BLEU is significantly lower, at 36.2%, compared to 75.5% in Table 2. This suggests that although BERTScore is flawed, it is at least more accurate than BLEU.

5.2 PE^{2}rr

Using the PE^{2}rr dataset, we again test conditions (i) and (ii). We use an approach like that taken with the TQ-AutoTest dataset. First, we filter out examples in which the two references translations provided are identical; in this case, the BERTScore will be trivially 1. For all other examples (y, y'), we compute BERTScore(y, y'). Then, for each error category, we compute the mean BERTScore (F_1) among all examples that contain at least one of that error. Results are reported in Table 3.

We find that the average BERTScore assigned to pairs of correct translations in this dataset, 0.587, is much lower than in TQ-AutoTest, where the average BERTScore assigned to correct translations pairs was 0.815. Moreover, PE^{2}rr examples with no errors in their machine translation have a higher BERTScore assigned to their correct translations.

Testing condition (ii) with the PE^{2}rr dataset is somewhat more challenging. First, we filter out any of the machine translations that have no errors (as we need a bad translation to test condition (ii)). Normally, the next step would be to compute and compare, using our two references translations y, y' and one bad translation b, BERTScore(y, y') and BERTScore(y, b).

However, a difficulty arises: while each example has two correct translations, and one incorrect machine translation, the two correct translations are not generated in the same way. One is a post-edit of the machine translation, while the other is an original reference, generated from the German text without the machine translation.

[4]We calculate BLEU as implemented in the *multi-bleu.perl* script at https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

If we choose the original reference to be y, and the post-edit to be y', this is a fair comparison; however, if we choose the post-edit to be y, the task becomes much more challenging. This is because BERTScore(y, b) will be comparing a post-edited machine translation to the original machine translation, and there will naturally be a good deal of overlap, even though b contains errors. So, we report results in two cases in Table 4: the easy case, where y is the original reference, and the hard case, where y is the post-edit.

This choice has a major effect on the ability of BERTScore to distinguish good translations from bad ones. When we choose the original reference as y, BERTScore has a high accuracy in all categories. In contrast, in the hard problem setting, BERTScore does poorly (at or below chance) in all categories except sentences with an untranslated word, which is likely easy for BERTScore to detect.

We again compare BERTScore to BLEU to contextualize our results for condition (ii) (in Table 4). We find that once more BERTScore outperforms BLEU, which achieves accuracies below 50% in all categories in the easy setting, and below 13% in the hard setting. The disparity between the easy and hard setting performance reflects the fact that BLEU also struggles to penalize bad translations with high n-gram overlap, compared to those with less overlap.

Finally, to provide an alternate explanation for the trend in BERTScore accuracies, we plot in Figure 4 the BERTScore accuracy for examples with n errors, in both the easy and hard scenarios. While in the easy scenario, BERTScore has an accuracy near 1 for $n >= 5$, in the hard scenario, the accuracy is lower, but increases with n. This suggests that it is easier for BERTScore to distinguish between translations that are good and those that are bad, but lexically similar to the postedited reference, when the latter contain more errors. Alternatively, we can view this as BERTScore being less sensitive to translation errors (except when they are numerous), and relatively more sensitive to the stylistic differences that exist between the post-edited and original reference sentences.

5.3 Grammatical Error Correction

For GEC, we test all three conditions. To test the first, we use the 11 annotators' corrections to create post-edited versions $(y_1, \ldots, y_{11})$ of the original sentences; these should all have the same meaning. Then, we compute the mean BERTScore (F_1) for each pair of annotators' post edits; these are reported in Figure 5. The mean BERTScore falls in a very narrow range (0.84-0.88), except for Annotators 1 and 8, who provided similar annotations.

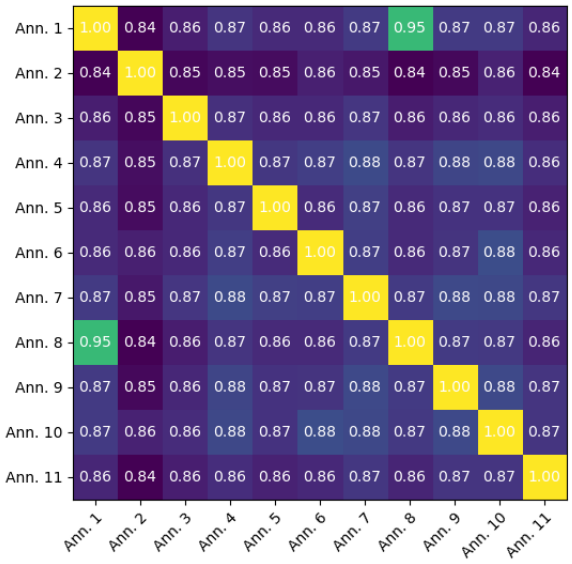


Figure 5: GEC: Heatmap of Average BERTScore (F_1) assigned when comparing references from a given pair of annotators

n: # of Errors	b_n: With Errors	b_0: Without Errors	y': Alternate Reference
1	0.841	0.850	0.865
2	0.830	0.844	0.860
3	0.820	0.839	0.856
4	0.814	0.828	0.847
5	0.812	0.821	0.842

Table 5: GEC: Average BERTScore (F_1) assigned when comparing a reference y to a) b_n, a sentence with n errors in it, b) b_0, the same sentence, but with all errors corrected, and c) y', an alternate reference sentence.

We test conditions (ii) and (iii) jointly. To do so, we need two correct translations, created independently of one another. We also need two partially correct translations (of different levels of correctness) and one correct translation; these must be created from the same set of edits, independent of the first two. To create these, we first select three annotators, and assign them roles: reference annotator, alternate reference annotator, and "error" annotator. Next, for $n = 1, \ldots, 5$, we filter out all examples where the error annotator made fewer than n error annotations. Then, for each example, the reference and alternate reference annotators' an-

notations are used to create one reference sentence (y) and alternate reference sentence (y'). The error annotator's annotations are used to create both a third, fully corrected sentence (b_0), and a sentence with n errors in it (b_n); the errors that remain are chosen at random.

Finally, we compute BERTScore(y, y'), BERTScore(y, b_0), and BERTScore(y, b_n); the mean values for each of these is reported in Table 5. Note that although b_0 and y' have no errors, their mean BERTScore still changes with n because we filter out examples that have fewer than n errors according to the error annotator. Thus, the downward trend with growing n reflects the fact that, as the number of errors in a sentence increases, the annotators' corrections diverge.

We can see that BERTScore does respect conditions (ii) and (iii) within the GEC data. The alternate reference and error-free sentence are always assigned a higher score than the sentence with errors (ii). Moreover, the sentences with more errors are assigned a lower score than those containing fewer errors (iii).

Also of note is the absolute magnitude of the BERTScores assigned in this experiment, which is much higher than those in the prior PE^{2}rr experiment. The mean BERTScore assigned to even examples containing 5 errors (0.812, Table 5), is much higher than the mean BERTScore assigned to PE^{2}rr reference pairs (0.587, Table 3). As before, we suggest that this occurs due to sensitivity to style. Although b_5 contains errors, it is stylistically similar to the reference, as they are both edits of the same sentence; in contrast, the two references in the PE^{2}rr dataset are very distinct from one another.

Finally, we perform this same evaluation with BLEU, again omitting full results for brevity. We find that BLEU, regardless of n, gives the alternate reference y' a higher score than the other sentences. However, the BLEU scores assigned to b_n and b_0 were similar, and b_0 did not consistently receive higher scores. Thus, we conclude that BERTScore also performs better on this task.

6 Discussion

From our experiments on three datasets, we draw three main findings. First, we find that the performance of BERTScore with respect to conditions (i) and (ii) can vary based on linguistic phenomena. Second, while BERTScore is generally capable, it has difficulties on a challenge dataset that tasks it with penalizing lexically similar incorrect translations and preferring translations that are lexically different but more correct. Third, BERTScore does, in certain circumstances, respect our third condition—it ranks worse bad translations below better bad translations.

With respect to the first finding, penalizing translations that incorrectly render function words seems to be the most difficult for BERTScore. In TQ-Autotest, this includes sentences with tag questions; in one example, the reference is "You're crazy, aren't you?", and secondary good translation is "You're crazy, right?", while incorrect sentences are "You're crazy, or?" and "You're crazy, are not you?". That is, the differences do not affect the main content of the sentence. In contrast, sentences with incorrectly resolved word ambiguity or larger, multi-word errors were easily penalized.

The second finding also confirms that BERTScore can more easily detect bad translations when there is less lexical overlap. In the easy problem setting, where both the good and bad candidate translations were lexically distinct from the reference, BERTScore easily distinguished the good translation from the bad. But, in the hard problem setting, where the bad translation has high lexical overlap with and was stylistically similar to the reference, BERTScore struggled. Due to this style sensitivity, BERTScore may be better-suited to scoring candidates from widely-differing systems, as opposed to closely-related systems, or multiple candidates from one system.

Our third finding, provides more positive results. In the GEC scenario, BERTScore was able to fulfill all three of the conditions we defined. It not only gave high scores to similar sentences and worse scores to sentences with errors; it also gave better scores to more grammatically correct sentences, even when they were not perfectly correct.

Unfortunately, the GEC dataset does not necessarily reflect the kinds of errors that occur in machine translation. It focuses primarily on grammatical errors, and thus contains fewer semantic errors. Moreover, since the GEC annotators had only the incorrect text, and no source text to work with, their annotations are occasionally in disagreement, as they each independently inferred the intended meaning of the incorrect text.

7 Related Work

The original paper introducing BERTScore (Zhang et al., 2020) naturally compared BERTScore's correlations with human judgments to that of other metrics. However, various other surveys of MT metrics, as well as datasets and methodologies have been conducted, offering insights into how MT system and metric performance should be measured.

Naturally, the WMT Metrics task, most recently run in 2020 (Mathur et al., 2020b) is one such forum for the evaluation of metrics. In this last iteration, metrics were evaluated based on their correlation with human judgment scores on the sentence, paragraph, and document level. BERTScore was not included even in the most recent iteration of the metrics task.

More recently, Kocmi et al. (2021) run a large-scale comparison of MT metrics, including BERTScore using a large dataset of translations with human judgments; they find that BERTScore's performance is middle-of-the-road, though better than BLEU, and recommend COMET (Rei et al., 2020) for general use.

Unfortunately, while these studies evaluate MT metrics, using human judgments alone cannot tell us when or why they may succeed or fail. In response to the results of 2019 WMT Metrics Task (Ma et al., 2019), Mathur et al. (2020a) note that correlations of metrics with human judgment can be highly sensitive to the number of systems in question, as well as outliers.

Fomicheva and Specia (2019) propose moving beyond correlation with human judgment alone as a standard for MT metric evaluation. To this end, they conduct a comprehensive study of MT metrics, and review datasets that have more fine-grained error and quality annotations. Despite this, datasets for MT metric evaluation with linguistic annotations or other annotations regarding sentence content are few.

8 Conclusion

BERTScore is a new metric for MT evaluation that uses BERT, and is as a result difficult to interpret. We define desiderata for BERTScore and other such metrics' performance, and use targeted datasets to find when BERTScore fails. We find that BERTScore fails to assign low scores when a bad candidate sentence has high lexical overlap with the reference in terms of content words. Despite this, in less challenging scenarios, BERTscore does well, and is able to rank sentences in order of their quality. Moreover, BERTScore outperforms BLEU score across the datasets and conditions we tested. However, these experiments are limited in scope, due to limited available data with appropriate annotations. Development of datasets for MT metric evaluation with linguistic annotation would aid in further work on this topic.

Acknowledgments

The authors would like to acknowledge the support of the grant 825303 (Bergamot) of the European Union's Horizon 2020 research and innovation programme (Michael Hanna), and 19-26934X (NEUREM3) of the Czech Science Foundation (Ondřej Bojar). Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ ID:90140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. 2020. What's in a name? are BERT named entity representations just as good for any other name? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 205–214, Online. Association for Computational Linguistics.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? 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 697–707, Beijing, China. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Marina Fomicheva and Lucia Specia. 2019. Taking MT Evaluation Metrics to Extremes: Beyond Correlation with Human Judgments. *Computational Linguistics*, 45(3):515–558.

Goran Glavas and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *EACL*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in bert track syntactic dependencies? *ArXiv*, abs/1911.12246.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation.

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, page 228–231, USA. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

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 (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, 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 of the Association for Computational Linguistics*, pages 311–318.

Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?

Maja Popović and Mihael Arčan. 2016. PE2rr corpus: Manual error annotation of automatically pre-annotated MT post-edits. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 27–32, Portorož, Slovenia. European Language Resources Association (ELRA).

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ieva Staliunaite and Ignacio Iacobacci. 2020. Compositional and lexical semantics in roberta, bert and distilbert: A case study on coqa. In *EMNLP*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Evaluating Multiway Multilingual NMT in the Turkic Languages

Jamshidbek Mirzakhalov[a,b], **Anoop Babu**[a,b], **Aigiz Kunafin**[a], **Ahsan Wahab**[a], **Behzod Moydinboyev**[a,b], **Sardana Ivanova**[a,c], **Mokhiyakhon Uzokova**[a,d], **Shaxnoza Pulatova**[a,e], **Duygu Ataman**[a,f], **Julia Kreutzer**[a,g], **Francis Tyers**[a,h], **Orhan Firat**[a,g], **John Licato**[a,b], **Sriram Chellappan**[a,b]

[a]Turkic Interlingua, [b]University of South Florida,
[c]University of Helsinki, [d]Tashkent State University of Uzbek Language and Literature,
[e]Namangan State University, [f]NYU, [g]Google Research, [h]Indiana University

Abstract

Despite the increasing number of large and comprehensive machine translation (MT) systems, evaluation of these methods in various languages has been restrained by the lack of high-quality parallel corpora as well as engagement with the people that speak these languages. In this study, we present an evaluation of state-of-the-art approaches to training and evaluating MT systems in 22 languages from the Turkic language family, most of which being extremely under-explored (Joshi et al., 2019). First, we adopt the TIL Corpus (Mirzakhalov et al., 2021) with a few key improvements to the training and the evaluation sets. Then, we train 26 bilingual baselines as well as a multi-way neural MT (MNMT) model using the corpus and perform an extensive analysis using automatic metrics as well as human evaluations. We find that the MNMT model outperforms almost all bilingual baselines in the out-of-domain test sets and finetuning the model on a downstream task of a single pair also results in a huge performance boost in both low- and high-resource scenarios. Our attentive analysis of evaluation criteria for MT models in Turkic languages also points to the necessity for further research in this direction. We release the corpus splits, test sets as well as models to the public[1].

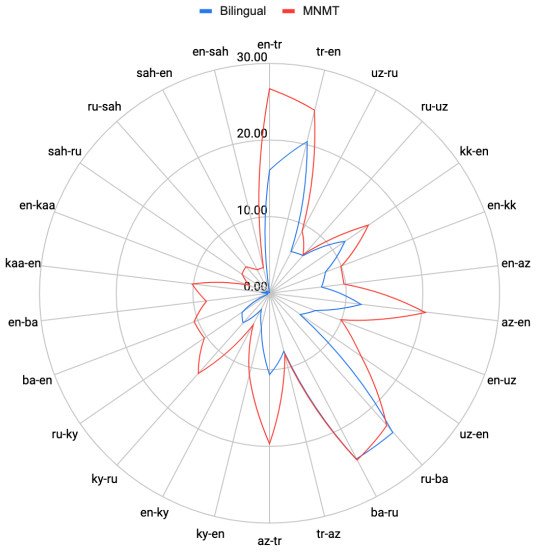


Figure 1: Performance comparison between bilingual baselines and the MNMT model on X-WMT test set.

1 Introduction

The last few years have seen encouraging advances in low-resource MT development with the increasing availability of public multilingual corpora (Agić and Vulić, 2019; Ortiz Suárez et al., 2019; Schwenk et al., 2019; El-Kishky et al., 2020; Tiedemann, 2020; Goyal et al., 2021; ∀ et al., 2020) and more inclusive multilingual MT models (Arivazhagan et al., 2019; Tiedemann and Thottingal, 2020; Fan et al., 2020). In this study, we take the Turkic language family into focus, which has not been studied at large in MT research (detailed review in Section 2). Most recently, in a wide evaluation of translation between hundreds of languages with a multilingual model (M2M-124) trained on large web-mined parallel data, translation into, from, and between Turkic languages was shown to be very challenging compared to other language families (Goyal et al., 2021). With the promise of strong transfer capabilities of multilingual models especially for related languages, we hope that the inclusion of a wider set of Turkic languages into a joint model can unlock automatic translation even for the very low-resourced Turkic languages where no prior translation models exist (Koehn, 2005; Choudhary and Jha, 2011; Post et al., 2012; Nomoto et al., 2018; Esplà-Gomis et al., 2019; ∀ et al., 2020).

To this aim, we adopt the TIL Corpus (Mirzakhalov et al., 2021) compiled by the Turkic Inter-

[1] https://github.com/turkic-interlingua/til-mt

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 518–530
November 10–11, 2021.

Name	Codes	Speakers	Data	MT?
English	en, eng	400.0M	38.6M	✓
Russian	ru, rus	258.0M	23.3M	✓
Turkish	tr, tur	85.0M	52.6M	✓
Kazakh	kk, kaz	13.2M	5.3M	✓
Uzbek	uz, uzb	27.0M	2.9M	✓
Azerbaijani	az, aze	23.0M	2.2M	✓
Tatar	tt, tat	5.2M	1.8M	✓
Kyrgyz	ky, kir	4.3M	1.7M	✓
Chuvash	cv, chv	1.0M	1.5M	✓
Turkmen	tk, tuk	6.7M	910.4K	✓
Bashkir	ba, bak	1.4M	880.5K	✓
Uyghur	ug, uig	10.0M	334.8K	✓
Karakalpak	kaa	583.0K	253.8K	✗
Khakas	kjh	43.0K	219.0K	✗
Altai	alt	56.0K	192.6K	✗
Crimean Tatar	crh	540.0K	185.3K	✗
Karachay-Balkar	krc	310.0K	162.8K	✗
Gagauz	gag	148.0K	157.4K	✗
Sakha	sah	450.0K	157.1K	✓
Kumyk	kum	450.0K	156.8K	✗
Tuvinian	tyv	280.0K	100.3K	✗
Shor	cjs	3.0K	2.3K	✗
Salar	slr	70.0K	766	✗
Urum	uum	190.0K	491	✗

Table 1: (The table indicates the language codes used for the Turkic languages along with the number of L1 speakers, amount of available data (in sentences) in our corpus. The column MT? indicates if there are currently available online machine translation systems for the language. K: thousand, M: million.)

lingua[2] community (Mirzakhalov, 2021) including X-WMT test sets with a few key improvements (Section 3). We train a multi-way NMT (MNMT) model on the entire parallel corpus, which constitutes the first large-scale multilingual translation model specifically for Turkic languages (Section 4). We perform an extensive analysis of the strengths and weaknesses of this model, comparing it to the bilingual baselines and evaluating it under a domain shift. We find that the MNMT model outperforms almost all bilingual baselines in the out-of-domain tests while it performs comparably or underperforms in the in-domain tests. We further analyze its capacity for transfer learning by fine-tuning the model on several language pairs all of which experience gains, both in- and out-of-domain scenarios. In addition, we complement the automatic evaluation with a human evaluation study for multiple languages (Section 5), gaining insights into types of common mistakes that the model makes and the suitability of different automatic metrics for Turkic languages. We plan on releasing the improved corpus, evaluation sets, and all the models to the public.

This work will not only enrich the landscape of languages currently considered in MT research and spur future research on NLP for Turkic languages but will hopefully also inspire the building of new translation engines and derived technologies for populations with millions of native speakers (Table 1).

2 Related Work

This section discusses the previous work on MT of these languages including the available corpora and languages resources. The 19 Turkic languages covered in the study are: Altai, Azerbaijani, Bashkir, Crimean Tatar, Chuvash, Gagauz, Karachay-Balkar, Karakalpak, Khakas, Kazakh, Kumyk, Kyrgyz, Sakha, Turkmen, Turkish, Tatar, Tuvan, Uyghur, and Uzbek. There are several other widely spoken languages that are left out from our study such as Shor, Salar, Urum, Nogai, Khorasani Turkic, Qashqai, and Khalaj, due to the lack (or very limited amount) of any available parallel corpora. Future work will focus on extending the corpus to these languages as well.

2.1 MT of Turkic Languages

The need for more comprehensive and diverse multilingual parallel corpora has sped up the creation of such large-scale resources for many language families and linguistic regions (Koehn, 2005; Choudhary and Jha, 2011; Post et al., 2012; Nomoto et al., 2018; Esplà-Gomis et al., 2019; ∀ et al., 2020). Tiedemann (2020) released a large-scale corpus for over 500 languages covering thousands of translation directions. The corpus currently includes 14 Turkic languages and provides bilingual baselines for all translation directions present in the corpus. However, most of the 14 Turkic languages contain a few hundred or a dozen samples. In addition, the varying and limited size of the test sets does not allow for the extensive analysis and comparisons between different model artifacts, linguistic features, and translation domains. More recently, Goyal et al. (2021) extended the previous Flores benchmark by providing human translated evaluation sets for 101 languages, among which 5 of them are from the Turkic family: Azerbaijani, Kazakh, Kyrgyz, Turkish, and Uzbek. Similarly, they train a large MNMT model and evaluate its performance

[2] https://turkicinterlingua.org/

using the benchmark.

A Russian-Turkic parallel corpus was curated for 6 different Turkic languages, and their bilingual baselines have been reported for both directions using different NMT-based approaches Khusainov et al. (2020). However, the dataset, test sets, and models are not released to the public which limits its use to serve as a comparable benchmark. Additionally, a rule-based MT framework for Turkic languages has been presented with 4 language pairs Alkım and Çebi (2019). Also, several rule-based MT systems have been built for Turkic languages which are publicly available through the Apertium[3] website Washington et al. (2019).

For individual languages in our corpus, there are several proposed MT systems and linguistic resources: Azerbaijani (Hamzaoglu, 1993; Fatullayev et al., 2008), Bashkir (Tyers et al., 2012), Crimean Tatar (Gökırmak et al., 2019; Altıntaş, 2001), Karakalpak (Kadirov, 2015), Kazakh (Assylbekov and Nurkas, 2014; Sundetova et al., 2015; Littell et al., 2019; Briakou and Carpuat, 2019; Tukeyev et al., 2019), Kyrgyz (Çetin and Ismailova), Sakha (Ivanova et al., 2019), Turkmen (Tantuğ and Adalı, 2018), Turkish (Turhan, 1997; El-Kahlout and Oflazer, 2006; Bisazza and Federico, 2009; Tantuğ et al., 2011; Ataman et al., 2017), Tatar (Salimzyanov et al., 2013; Khusainov et al., 2018; Valeev et al., 2019; Gökırmak et al., 2019), Tuvan (Killackey, 2013), Uyghur (Mahsut et al., 2004; Nimaiti and Izumi, 2012; Song and Dai, 2015; Wang et al., 2020), and Uzbek (Axmedova et al., 2019). Yet to our knowledge, there has not been a study that covers Turkic languages to such a large extent as ours, both in terms of multilingual parallel corpora and multiway NMT benchmarks across these languages.

3 TIL Corpus

As we adopt the TIL Corpus as the training data, we perform a few key modifications to better the quality of the datasets.

First, we notice that the alignments for the Bible[4] and TedTalks[5] datasets were not optimal as most "sentences" were actually comprised of multiple sentences in order to preserve the quality of the alignment with target sequence. For example, in the case of TedTalks, the original speech utterance may have been 2-3 sentences in text but the translation of that speech may end up differing by 1 or even more sentences depending on the translator. Common practice in this situation, as seen through multiple corpora across OPUS[6], is to leave the entire utterance as is to preserve the quality of the alignment even if the number of sentences do not match. Instead, we drop the examples where the total number of sentences do not match and split (and realign) the cases where they do. This naturally increased the overall number of sentence alignments in both the Bible and TedTalks corpora for all language pairs.

Second, we perform a corpus-wide length and length-ration filtering where we drop sentence pairs that are single words as well as the entries where source and target ratio is over 2.

Third, we re-curate the in-domain evaluation sets following the improvements to the corpus. Details on the evaluation sets are described further in Section 3.1.

3.1 Curation of evaluation sets

The original TIL Corpus introduced three evaluation sets with different domains (Bible, TedTalks, and X-WMT). To simplify the analysis of the models, we re-curate the in-domain evaluation sets by randomly sampling from each corpora. X-WMT is used as the out-of-domain test set since it is from the news domain with substantial amount of new words/terms that most of the language pairs lack. The curation steps for the test sets are presented below.

3.1.1 In-domain Evaluation Sets

In-domain development and test sets are randomly sampled from each language pair and can serve as evaluation sets for both bilingual and multilingual models. The size of the development and test sets depends on the amount of training data available. More specifically, development and test sizes are 5k each if the train size is over 1 million parallel sentences, 2.5k if over 100k, 1k if over 10k, and 500 if over 2.5k. All test and development samples are removed from the training corpus for that language pair. Overall, this yields development and test sets for exactly 400 language pairs.

[3] https://www.apertium.org/

[4] https://bible.is/

[5] https://www.ted.com/participate/translate

[6] https://opus.nlpl.eu/

	en	ru	ba	tr	uz	ky	kk	az	sah	kaa
en	—									
ru	**1000**	—								
ba	1000	**1000**	—							
tr	**800**	800	800	—						
uz	**900**	900	900	600	—					
ky	500	**500**	500	400	500	—				
kk	700	700	700	500	**700**	500	—			
az	**600**	600	600	500	600	500	500	—		
sah	300	**300**	300	300	300	300	300	300	—	
kaa	300	300	300	**300**	300	300	300	300	300	—

Table 2: X-WMT test sets. Bolded entries indicate the original translation direction.

3.1.2 X-WMT Test Set

X-WMT is a challenging and human-translated test set in the news domain based on the professionally translated test sets in English-Russian from the WMT 2020 Shared Task (Mathur et al., 2020). It was originally introduced in the TIL Corpus and we adopt the test sets as they are. Currently, the test set extends into 8 Turkic languages (Bashkir, Uzbek, Turkish, Kazakh, Kyrgyz, Azerbaijani, Karakalpak, and Sakha) paired with English and Russian. Table 2 highlights the currently available test set directions. Bolded entries in the table indicate the original direction of the translation.

4 Experimental Setup

4.1 Bilingual Experiments

To serve as initial baselines, we train 26 bilingual baselines using the corpus and report the performance on the in-domain test set as well as the X-WMT set (out-of-domain) as described in Section 3.1.2. The selection of the language pairs was constricted by the availability of both in-domain and out-of-domain test sets to enable more meaningful insights from the experiments.

4.1.1 Model details

All models are Transformers (*transformer-base*) (Vaswani et al., 2017b) and are trained using the JoeyNMT framework (Kreutzer et al., 2019). In the preprocessing stage, we use Sacremoses for tokenization and apply byte pair encoding (BPE) (Sennrich et al., 2015; Dong et al., 2015) with a joint vocabulary size of 4k and 32k. Models use 512-dimensional word embeddings and hidden layers and are trained with the Adam optimizer (Kingma and Ba, 2015). A learning rate of $3{*}10^{-4}$ is applied along with a dropout rate of 0.3. We use a batch size of 4096 BPE tokens with 8 accumulations to simulate training on 8 GPU machines. All models, except English-Turkish and Turkish-English, are trained on Google Colab's freely availably preemptible GPUs.

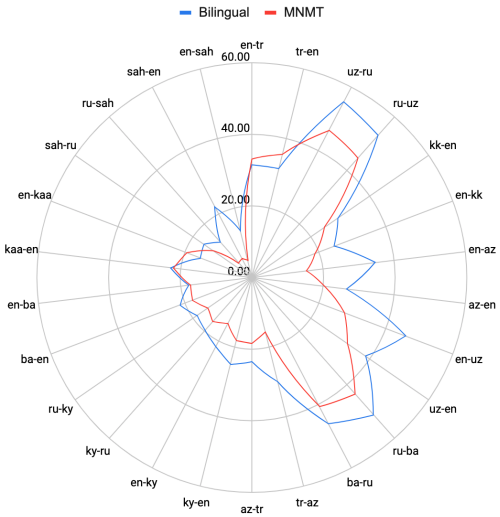


Figure 2: Performance comparison between bilingual baselines and the MNMT model on the in-domain test set.

4.2 Multilingual Experiments

To examine the extent of transfer learning and generalization within our corpus, we train a multiway multilingual NMT model on the entire dataset covering almost 400 language directions. We then compare the performance of the model on the in-domain and out-of-domain test sets across a range of language pairs.

4.2.1 Data Preprocessing

Similar to the bilingual data preprocessing, the entire corpus has been tokenized using Sacremoses[7] and samples longer than 300 words have been filtered out. In addition, we perform cross-filtering of test and dev sets of all language pairs from the training corpus, as it is very necessary to do so in any MNMT model using a multiway corpus. Since the corpus is relatively unbalanced, we perform a temperature-based sampling with a value of 1.25. Although a higher temperature value between 2 and 3 would further balance our corpus, it would increase the dataset size by 8x with t=2 and 25x with t=3. This increase would limit our ability to train the model due to the restrained compute resources. Originally, the overall training set size is at around 133 million samples and this increases to 244 million after the sampling procedure. We ap-

[7] https://github.com/alvations/sacremoses

Pairs	Train size	In-Domain Test					X-WMT Test				
		Bilingual		MNMT			Bilingual		MNMT		
		BLEU	Chrf	BLEU	Chrf	PPL	BLEU	Chrf	BLEU	Chrf	PPL
en-tr	35.8M	31.45	0.51	**33.09**	**0.51**	8.18	16.04	0.55	**26.74**	**0.56**	12.76
tr-en	35.8M	31.37	0.50	**35.48**	**0.52**	7.19	20.39	0.51	**24.66**	**0.55**	10.88
ru-uz	1.3M	**53.12**	**0.76**	44.73	0.71	3.02	6.58	0.41	**6.70**	**0.42**	82.20
uz-ru	1.3M	**55.39**	**0.76**	46.42	0.71	3.27	6.08	0.36	**9.16**	**0.39**	16.70
en-kk	564.8K	**24.53**	**0.54**	18.92	0.49	10.45	7.82	0.40	**9.92**	**0.43**	10.02
kk-en	564.8K	**29.17**	**0.51**	24.67	0.48	7.47	12.00	0.42	**15.71**	**0.44**	26.02
az-en	548.9K	**26.65**	**0.48**	20.47	0.42	7.70	12.01	0.41	**20.41**	**0.49**	14.46
en-az	548.9K	**34.73**	**0.56**	15.27	0.42	8.74	6.79	0.38	**9.71**	**0.43**	10.59
en-uz	529.6K	**45.95**	**0.66**	27.80	0.51	6.04	6.34	0.40	**9.89**	**0.42**	47.45
uz-en	529.6K	**38.72**	**0.58**	32.44	0.50	6.15	4.81	0.24	**14.45**	**0.45**	19.08
ba-ru	523.7K	**46.02**	**0.69**	40.59	0.64	3.75	24.39	**0.58**	**24.57**	0.57	5.49
ru-ba	523.7K	**51.26**	**0.74**	43.44	0.67	3.24	**24.31**	**0.59**	23.13	0.56	6.29
az-tr	410.1K	**23.47**	**0.48**	18.40	0.43	8.87	10.61	0.43	**19.63**	**0.48**	23.42
tr-az	410.1K	**29.97**	**0.53**	15.71	0.42	8.37	7.78	0.39	**8.21**	**0.42**	14.51
en-ky	312.6K	**21.66**	**0.44**	14.54	0.38	10.77	2.33	0.27	**4.64**	**0.34**	19.57
ky-en	312.6K	**24.96**	**0.42**	18.01	0.38	11.02	4.65	0.29	**10.87**	**0.39**	35.64
ky-ru	293.7K	**19.63**	**0.40**	16.30	0.38	10.04	5.23	0.30	**14.08**	**0.44**	9.43
ru-ky	293.7K	**18.57**	**0.43**	14.82	0.40	9.58	4.42	0.35	**10.35**	**0.45**	11.52
ba-en	34.3K	**21.51**	0.36	17.79	**0.37**	10.81	0.32	0.19	**10.55**	**0.40**	37.89
en-ba	34.3K	**17.78**	0.33	17.29	**0.35**	10.52	0.16	0.14	**8.35**	**0.34**	21.43
en-kaa	17.1K	15.34	0.40	**19.42**	**0.46**	8.83	0.31	0.19	**2.82**	**0.27**	77.93
kaa-en	17.1K	**22.82**	0.43	21.95	**0.48**	8.56	1.04	0.21	**10.21**	**0.38**	38.17
ru-sah	9.2K	**13.26**	**0.35**	5.46	0.19	30.82	0.12	0.16	**4.64**	**0.17**	58.01
sah-ru	9.2K	**16.35**	**0.36**	13.11	0.26	23.00	0.42	0.18	**4.41**	**0.25**	40.68
en-sah	8.1K	**13.45**	**0.36**	4.98	0.18	34.31	0.04	**0.14**	**3.46**	0.12	75.38
sah-en	8.1K	**22.19**	**0.40**	5.90	0.23	24.58	0.16	0.21	**3.38**	**0.24**	110.50

Table 3: Experiments results from bilingual baselines and MNMT model evaluated on the in-domain and out-of-domain test sets. *BLEU* and *Chrf* uses the SacreBLEU implementation and *PPL* refers to the internal perplexity of the MNMT model.

ply the sentencepiece[8] implementation of the byte pair encoding (BPE) (Sennrich et al., 2016) with a joint vocabulary size of 64k. Following the method from Ha et al. (2016) and Johnson et al. (2017), we prepend a target language token to the source sentences to enable many-to-many translation.

4.2.2 Model details

We train the model using the Transformer architecture in the *transformer-base* configuration. More specifically, we use the *transformer_wmt_en_de* version from Fairseq (Ott et al., 2019) implementation[9] with 6 layers both in the encoder and decoder. Configuration of the model closely follows the original implementation of the Transformer (Vaswani et al., 2017a) with the model dimension set at 512 and hidden dimension size at 2048. We apply a dropout rate of 0.3, the learning rate of $5 * 10^{-4}$, and warm-up updates of 40k. The effective batch size is 16,384 BPE tokens. The model is trained using 4 NVIDIA V100 GPU machines for a little over 1 million steps which takes about 36-48 hours.

4.3 Evaluation of Models

Automatic evaluation metrics used to compare the performance of bilingual baselines and MNMT are token-based corpus BLEU (Papineni et al., 2002) and character-based Chrf (Popović, 2015). While corpus BLEU is the de-facto standard in MT (Marie et al., 2021), Chrf might work better for morphologically rich languages because it can reward partially correct words. We also report the MNMT model's internal perplexity to better highlight the language pairs in which the model struggles most. We evaluate the models on the in-domain and X-WMT evaluation sets. The gap between scores on in-domain

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

[9] https://github.com/pytorch/fairseq/tree/master/examples/translation

	Bilingual		MNMT		Gain	
	BLEU	**Chrf**	**BLEU**	**Chrf**	**BLEU**	**Chrf**
XX-En	6.92	0.31	13.78	0.42	+6.86	+0.11
En-XX	4.57	0.30	9.37	0.36	+4.80	+0.06
XX-Ru	9.03	0.36	13.06	0.41	+4.03	+0.06
Ru-XX	8.86	0.38	11.21	0.40	+2.35	+0.02
XX-XX	8.99	0.38	12.49	0.41	+3.49	+0.04

Table 4: Performance comparison with different language groups and their overall gains in the MNMT setup. XX refers to the Turkic languages in the corpus.

	Adequacy				Fluency			
	Avg	**k**	**LL**	**UL**	**Avg**	**k**	**LL**	**UL**
en-tr	2.97	0.33	0.23	0.43	3.20	0.12	0.04	0.21
tr-en	2.95	0.45	0.36	0.55	3.18	0.40	0.30	0.50
en-uz	2.77	0.18	0.10	0.26	2.93	0.28	0.17	0.38
uz-en	3.05	0.28	0.20	0.37	3.19	0.29	0.18	0.39
ba-ru	2.74	0.58	0.48	0.67	3.34	0.63	0.54	0.73
ru-ba	2.81	0.27	0.17	0.37	3.06	0.19	0.09	0.29

Table 5: **Avg** represents the average score for either Adequacy or Fluency given by the annotators for each language pair. **k** represents the Cohen's Kappa score. **LL** represents the Lower Limit within 95% confidence. **UL** represents the Upper Limit within 95% confidence.

versus out-of-domain translations is particularly interesting since it gives us an estimate of domain robustness and generalization, as well as mimics a realistic shift from the training domain to the domain of interest for potential users or downstream applications.

4.4 Bilingual baselines vs MNMT

Table 3 shows all the results for the bilingual baselines and MNMT as evaluated on two test tests. The first obvious trend in the table is the dominance of the bilingual baselines on the in-domain test sets as they overperform the MNMT model in most of the high- to mid-resource language pairs. As the train size decreases, the results become more comparable in terms of BLEU and even better for MNMT when evaluated in Chrf. When tested under a domain shift with the X-WMT set, MNMT results in gains across almost all pairs. However, it is important to note that there is a noticeable performance drop that follows the domain shift as can be seen in Figures 1 and 2. This highlights a realistic phenomenon of generalization and sets an expectation of the model's capabilities in real-world use cases.

Another observation in Table 3 is that all of the language pairs having fewer than 100k training samples (8 total) in our bilingual baselines barely pass the mark of 1 BLEU score or 0.2 Chrf in the out-of-domain test. However, in the MNMT setup, the average BLEU and Chrf score for those 8 low-resource pairs are 5.98 and 0.27 respectively. While these scores indicate that these pairs are still extremely low in quality and potentially unusable in practice, gains are promising given the amount of resources and a moderately-sized MNMT model.

To examine the generalization of the MNMT model into different language groups, we calculate the average gains for all pairs translating into English (XX-En), from English (En-XX), into Russian (XX-Ru), from Russian (Ru-XX), and direct pairs (XX-XX). Table 4 shows the average gains per category in terms of BLEU and Chrf. As it looks, translating from and into English sees the most gains, which is very consistent with the findings from the community (Arivazhagan et al., 2019; Goyal et al., 2021). A positive trend is the increasing quality of direct pairs which are very comparable to the non-Turkic pairs. We hypothesize that one of the main reasons for this is that the TIL Corpus is a multi-centric dataset with training data between almost all language pairs which allows us to train a complete Multilingual Neural Machine Translation (cMNMT) (Freitag and Firat, 2020). As shown in (Freitag and Firat, 2020; Fan et al., 2021), MNMT models trained on multi-centric parallel corpora tend to result in performance gains between non-English pairs.

4.5 BLEU vs Chrf

Figure 3 compares BLEU and Chrf for all bilingual and multilingual models on X-WMT. We distinguish between translating into and from Turkic languages since all Turkic languages feature agglutination. As hinted above, we suspect that BLEU might underestimate translation quality when translating into Turkic languages. The graph shows a clear distinction that confirms this: For translations into non-Turkic languages, the relation between Chrf and BLEU is almost linear, with a Pearson correlation of 0.98 and a rank correlation of 0.98 as well. For translation into Turkic, the trend follows a more curved line, with a largely higher Chrf-to-BLEU ratio. The Pearson correlation is much lower at 0.87, but the rank correlation is only slightly lower than for non-Turkic languages at 0.92. Consequently, we can expect the same BLEU score to correspond to a higher Chrf score when translating into Turkic languages than from them. This means that while Chrf and BLEU are likely to pro-

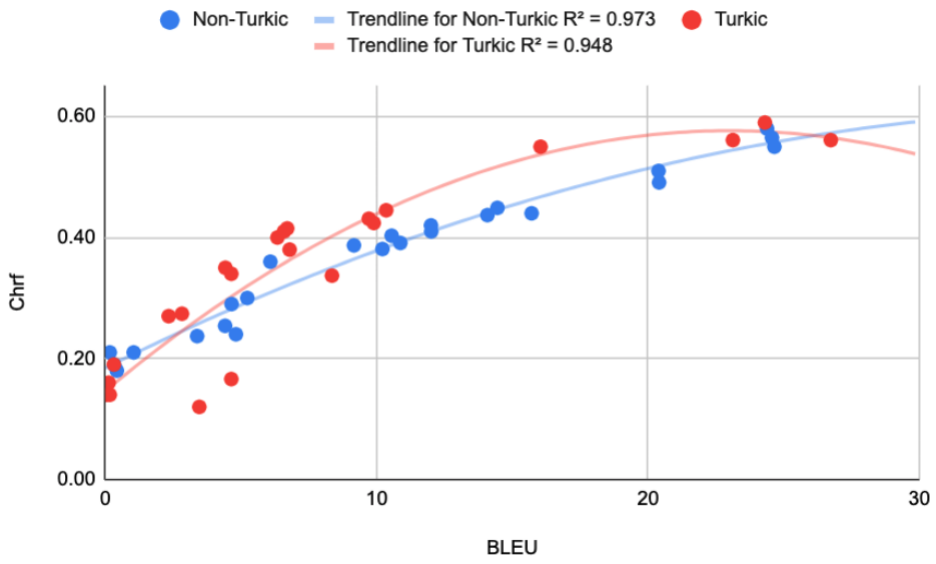


Figure 3: Correlations between BLEU and Chrf scores when the target language is Turkic and non-Turkic.

duce similar rankings of systems (at least in our scenario with standard comparable Transformer models), the Chrf score might better characterize the absolute translation quality. Our human evaluation does not cover sufficient language pairs (three from and three into Turkic languages) to yield a reliable empirical confirmation for this hypothesis. Future studies of larger scale as in the WMT metrics shared task (Mathur et al., 2020) will be needed.

5 Human Evaluation of MNMT

5.1 Human evaluation setup

To facilitate analysis on how well evaluation metrics measure the quality of the translations, we conduct human evaluations using the outputs from the MNMT model on the X-WMT set. We use Direct Assessment (DA) and follow the TAUS guidelines[10] with the only exception being the number of annotators per language pair, where we employ 2 annotators per language pair instead of 4[11]. In our DA, two hundred sentences of the MNMT model's output per language pair are evaluated based on its adequacy and fluency on respective 1-4 point scales. Annotators received an explanation of the rating scales with the task (e.g. "Adequacy: On a 4-point scale rate how much of the meaning is represented in the translation: 4: Everything 3: Most 2: Little 1: None"). To measure the inter-annotator agreement (IAA) between the two annotators of each language pair, we compute the Weighted Cohen's Kappa statistic (Cohen, 1960).

The language pairs involved in this human study are English-Turkish, Turkish-English, Bashkir-Russian, Russian-Bashkir, Uzbek-English, and English-Uzbek. These pairs were selected on the basis of language and script diversity, their performance on the X-WMT test set, and the availability of annotators.

5.2 Discussion and Results

The results of the average adequacy and fluency for each language pair are shown by Table 5. Most of the chosen language pairs received an average score of around 3 for both adequacy and fluency. This indicates that the model was largely able to convey most intended meaning in a good grammatical sense to a native speaker. Fluency is consistently rated higher than adequacy, which is a common theme in NMT evaluation (Martindale et al., 2019). The large difference in BLEU (5 BLEU points) between en-uz and uz-en is still noticeable, but much smaller according to the human evaluation. Chrf estimates a quality difference of 0.3 here, which is closer to the human estimate.

The Cohen's Kappa scores for each language pair are present in Table 5. As Cohen's Kappa is a measure from 0–1 of how well the two annotators agreed with their evaluations while removing possible agreements by chance, Cohen's Kappa score serves as one metric in deciding the reliability of

[10] https://rb.gy/eqlgbm
[11] Due to limited resources.

en-tr	
Adequacy: 3.00 — **Fluency:** 4.00 — *Fluent Output with Inadequate Verbal Tense*	
`Reference` Toyota, Subaru'daki hissesini 'den fazla artıracağını söyledi.	`Hypothesis` Toyota, Subaru'daki hisseyi 'den fazla artırdığını söyledi.
Adequacy: 4.00 — **Fluency:** 3.00 — *Lexical choice preserves meaning, still not the natural construction*	
`Reference` Başka birisi ağır yaralandı.	`Hypothesis` Başka bir kişi kötü yaralandı.
tr-en	
Adequacy: 3.00 — **Fluency:** 2.00 — *Some of the translations made lost the original meaning*	
`Reference` The schoolgirl who died from catastrophic injuries following a **suspected** hit-and-run in Newcastle has been **pictured** for the first time.	`Hypothesis` After a **suspicious** hit-and-run in Newcastle's, the student who died badly was first **seen**.
Adequacy: 3.00 — **Fluency:** 4.00 — *Maintains grammatical form, but changes the meaning*	
`Reference` He further **dismissed** the embargo as an **attack** on the rights of citizens.	`Hypothesis` He also **denied** the ambargo by defending an **attack** on citizens' rights.
ba-ru	
Adequacy: 2.00 — **Fluency:** 3.00 — *"kiss" translates to "kill" and changes the meaning completely*	
`Reference` В ночь после выборов, пишет Ло Бьянко в своей книге, Карен Пенс отказалась поцеловать мужа.	`Hypothesis` В свою книгу Ло Бьянко, в ночь после выборов, Карен Пенс отказывается от смерти мужа.
Adequacy: 3.00 — **Fluency:** 2.00 — *Incorrect pronoun ("she" to "he"). Few awkward translations*	
`Reference` Поэтому она откликнулась на вакансию в Fast Trak Management, маленькой компании, которая называет себя "маркетинговой фирмой номер один в Северной Вирджинии".	`Hypothesis` Поэтому он согласился на вакансию Fast Trak Management в малой компании, которая называла себя "Первую маркетинговую фирму в Северной Вирджинии".
ru-ba	
Adequacy: 3.00 — **Fluency:** 2.00 — *When several verbs are present, some are omitted from the translation*	
`Reference` Видео Пирзаданың бер нисә йылан һәм аллигаторҙо тотоп торғанын күрһәтә.	`Hypothesis` Дәүләт еренә йәмәғәт **access** Видеоға ярашлы, Пирзада бер нисә йылан һәм алгигатор менән нисек эш итә.
Adequacy: 2.00 — **Fluency:** 3.00 — *A whole part of the original sentence is omitted from the translation*	
`Reference` iHandy тарафынан киң билдәле эмодзи-ҡушымталар серияһы сығарылды, әммә улар ҙа Google Play Store системаһынан шунда уҡ юйылды.	`Hypothesis` iHandy Google Play Store-ҙан сығарылған популяр эмодзи-приложениялар серияһы булдырылды.
uz-en	
Adequacy: 2.00 — **Fluency:** 2.00 — *Changed the order events*	
`Reference` Antonio Brown has indicated he's not retiring from the NFL, only a few days after announcing he was done with the league in a rant.	`Hypothesis` Antonio Braun said that after a few days after the NFL, he won't leave after he announced that he was engaged in league.
Adequacy: 2.00 — **Fluency:** 4.00 — *Improper changes from original nouns, and different sense of "hold"*	
`Reference` **Harker** says **Fed** should '**hold** firm' on interest rates	`Hypothesis` **Everyone** thinks that this is how to **hold** the **Federal** rate percentages.
en-uz	
Adequacy: 3.00 — **Fluency:** 3.00 — *"Gumonlanuvchi":"a suspect"."Shubhachi":"someone who suspects"*	
`Reference` Keyin ushbu mashinadan uch nafar **gumonlanuvchi** tushayotganini ko 'rishdi.	`Hypothesis` Keyinchalik uchta **shubhachi** mashinadan chiqib ketganini ko'rishdi.
Adequacy: 2.00 — **Fluency:** 2.00 — *Use of a correct but a foreign word (başarısız)*	
`Reference` WeWork's Neumann muvaffaqiyatsiz IPO o 'tkazilgandan so'ng o 'zini bosh direktor lavozimidan chetlatishga ovoz berdi	`Hypothesis` **Biz Work"s** Neumann IPO **başarısız** bo'lganidan so'ng **O'zbekiston** Bosh direktori sifatida ovoz berdi

Table 6: Qualitative Analysis of the MNMT model output for 6 language pairs. The `Reference` sentence shows the intended translation while the `Hypothesis` shows the MNMT model output.

Pairs	Train Size	In-Domain		X-WMT	
		BLEU	Chrf	BLEU	Chrf
ru-ba	523.7K	54.48 (+11.04)	0.743 (+0.07)	24.85 (+0.54)	0.569 (-0.02)
ky-en	312.6K	24.21 (+6.2)	0.42 (+0.05)	10.26 (+5.61)	0.38 (+0.09)
en-ba	34.3K	30.43 (+13.14)	0.46 (+0.11)	4.56 (+4.4)	0.22 (+0.08)
ru-sah	9.2K	49.46 (+44.00)	0.585 (0.4)	22.05 (+21.93)	0.348 (+0.19)

Table 7: Experiment results from the finetuning of the MNMT model.

the evaluations. We see that the reliability varies across language pairs and between adequacy and fluency. Translation into English or Russian has a higher agreement on average than in the opposite direction (en/tr is a tie).

5.3 Qualitative Analysis

To gain better qualitative insight into the model outputs in each of the 6 language directions, we asked the annotators to identify 2 examples that highlight the most commonly witnessed mistakes during their review. Table 6 showcases those examples along with a brief explanation for their scores. From this analysis, it seems that the severity of mistakes that the MNMT model makes in *adequacy* tends to range from certain words being translated to a slightly different meaning to the original intention of the sentence being lost. As for *fluency*, the errors seem to range from awkward wording to clear grammatical mistakes. There are a few cases where there is an off-target translation for a word or a segment of the sentence.

6 The Promise of MNMT: Cross-lingual Knowledge Transfer

One of the biggest advantages of a large MNMT model is its capacity for transfer learning as can be accomplished through fine-tuning. Since we plan on releasing the model to the public, we believe many understudied and underperforming language pairs could benefit from cross-lingual knowledge transfer. This phenomenon is well-known in the broader NLP community as well as in MT research. To test this hypothesis, we fine-tune our MNMT model on 4 language pairs ranging from high(er)-resource to extremely low-resource in training data available. Table 7 shows the results of the experiments. As it can be seen, the performance of the models improves steadily across all resource types, low-resource cases experiencing gains up to 44 BLEU points (or 0.4 Chrf) from the bilingual baselines in the in-domain evaluation. However, in out-of-domain scenarios, gains are not as significant. Mid- to high-resource pairs improve modestly in the range of 1–5 BLEU points (or 0–0.1 Chrf) while a low-resource pair, Russian-Sakha gains up to 22 BLEU points (0.19 Chrf).

7 Future Work and Conclusion

In this work, we train and evaluate the first large-scale MNMT model for the Turkic language family which consists of many underexplored languages. Among many results, we find it very promising to train and finetune a MNMT model with a language family corpus as it boosts the cross-lingual knowledge transfer between the related languages and consistently improves over the strong bilingual baselines in out-of-domain scenarios. Our analysis also shows that Chrf and BLEU do not correlate in the same when the target language group if different: BLEU underestimates the translations for the Turkic languages.

In the future work, we hope to include more of the underrepresented Turkic language pairs in the study and explore the potential of transfer learning into the translation of unseen languages and language pairs (”zero-shot”).

Acknowledgements

We thanks all of the members and partners of the Turkic Interlingua (TIL) community for their contributions to the project. Namely, we would like to thank our dedicated translators and annotators: Nurlan Maharramli, Sariya Kagarmanova, Iskander Shakirov, Aydos Muxammadiyarov, Ziyodabonu Qobiljon qizi, Alperen Cantez, Doniyorbek Rafikjonov, Mukhammadbektosh Khaydarov, Madina Zokirjonova, Erkinbek Vokhabov, Petr Popov, Abilxayr Zholdybai and Akylbek Khamitov. We also acknowledge and appreciate significant dataset contributions from Rasul Karimov, Khan Academy O’zbek[12], and the Foundation for the Preservation and Development of the Bashkir Language[13].

[12] https://uz.khanacademy.org/

[13] https://bsfond.ru/

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Emel Alkım and Yalçın Çebi. 2019. Machine translation infrastructure for Turkic languages (MT-Turk). *The International Arab Journal of Information Technology*, 16(3):380–388.

Kemal Altıntaş. 2001. *Turkish to Crimean Tatar machine translation system*. Ph.D. thesis, Bilkent University.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges.

Zhenisbek Assylbekov and Assulan Nurkas. 2014. Initial explorations in kazakh to english statistical machine translation. In *The First Italian Conference on Computational Linguistics CLiC-it 2014*, page 12.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Xolisa Axmedova, Guzal Abdujalilova, and Umida Abdurahmonova. 2019. Algorithm based on linguistic models in machine translation between russian and uzbek. *ACADEMICIA: An International Multidisciplinary Research Journal*, 9(12):16–21.

Arianna Bisazza and Marcello Federico. 2009. Morphological pre-processing for turkish to english statistical machine translation. In *Proceedings of IWSLT 2009*.

Eleftheria Briakou and Marine Carpuat. 2019. The university of Maryland's Kazakh-English neural machine translation system at WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 134–140.

Mustafa Alp Çetin and Rita Ismailova. Assisting tool for essay grading for Turkish language instructors. *MANAS Journal of Engineering*, 7(2):141–146.

Narayan Choudhary and Girish Nath Jha. 2011. Creating multilingual parallel corpora in indian languages. In *Language and Technology Conference*, pages 527–537. Springer.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

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. Association for Computational Linguistics.

Ilknur Durgar El-Kahlout and Kemal Oflazer. 2006. Initial explorations in english to turkish statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 7–14.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Miquel Esplà-Gomis, Mikel L Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Rauf Fatullayev, Ali Abbasov, and Abulfat Fatullayev. 2008. Dilmanc is the 1st MT system for Azerbaijani. *Proc. of SLTC-08, Stockholm, Sweden*, pages 63–64.

∀, Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Tajudeen Kolawole, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddee Hassan Muhammad, Salomon Kabongo, Salomey Osei, et al. 2020. Participatory research for low-resourced machine translation: A case study in african languages. *Findings of EMNLP*.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. *arXiv preprint arXiv:2010.10239*.

Memduh Gökırmak, Francis Tyers, and Jonathan Washington. 2019. Machine Translation for Crimean

Tatar to Turkish. In *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 24–31.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Ilker Hamzaoglu. 1993. *Machine translation from Turkish to other Turkic languages and an implementation for the Azeri language*. Ph.D. thesis, MSc Thesis, Bogazici University, Istanbul.

Sardana Ivanova, Anisia Katinskaia, and Roman Yangarber. 2019. Tools for supporting language learning for sakha. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 155–163, Turku, Finland. Linköping University Electronic Press.

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.

Pratik Joshi, Christain Barnes, Sebastin Santy, Simran Khanuja, Sanket Shah, Anirudh Srinivasan, Satwik Bhattamishra, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali. 2019. Unsung challenges of building and deploying language technologies for low resource language communities. *arXiv preprint arXiv:1912.03457*.

Azizbek Kadirov. 2015. The algorithm of machine translation from uzbek to karakalpak. *TurkLang-2015*, page 24.

Aidar Khusainov, Dzhavdet Suleymanov, Rinat Gilmullin, and Ajrat Gatiatullin. 2018. Building the Tatar-Russian NMT system based on re-translation of multilingual data. In *International Conference on Text, Speech, and Dialogue*, pages 163–170. Springer.

Aidar Khusainov, Dzhavdet Suleymanov, Rinat Gilmullin, Alina Minsafina, Lenara Kubedinova, and Nilufar Abdurakhmonova. 2020. First Results of the "TurkLang-7" Project: Creating Russian-Turkic Parallel Corpora and MT Systems.

Rachel Killackey. 2013. Statistical Machine Translation from English to Tuvan.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Patrick Littell, Chi-kiu Lo, Samuel Larkin, and Darlene Stewart. 2019. Multi-source transformer for Kazakh-Russian-English neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 267–274.

Muhtar Mahsut, Yasuhiro Ogawa, Kazue Sugino, Katsuhiko Toyama, and Yasuyoshi Inagaki. 2004. An experiment on Japanese-Uighur machine translation and its evaluation. In *Conference of the Association for Machine Translation in the Americas*, pages 208–216. Springer.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online. Association for Computational Linguistics.

Marianna Martindale, Marine Carpuat, Kevin Duh, and Paul McNamee. 2019. Identifying fluently inadequate output in neural and statistical machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 233–243, Dublin, Ireland. European Association for Machine Translation.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Jamshidbek Mirzakhalov. 2021. *Turkic Interlingua: A Case Study of Machine Translation in Low-resource Languages*. Ph.D. thesis, University of South Florida.

Jamshidbek Mirzakhalov, Anoop Babu, Duygu Ataman, Sherzod Kariev, Francis Tyers, Otabek Abduraufov, Mammad Hajili, Sardana Ivanova, Abror Khaytbaev, Antonio Laverghetta Jr., Behzodbek

Moydinboyev, Esra Onal, Shaxnoza Pulatova, Ahsan Wahab, Orhan Firat, and Sriram Chellappan. 2021. A large-scale study of machine translation in the turkic languages.

Maimitili Nimaiti and Yamamoto Izumi. 2012. A rule based approach for japanese-uighur machine translation system. In *2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing*, pages 124–129.

Hiroki Nomoto, Kenji Okano, David Moeljadi, and Hideo Sawada. 2018. Tufs asian language parallel corpus (talpco). In *Proceedings of the Twenty-fourth Annual Meeting of the Association for Natural Language Processing*, pages 436–439.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: 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, Philadelphia, Pennsylvania, 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, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409.

Ilnar Salimzyanov, J Washington, and F Tyers. 2013. A free/open-source Kazakh-Tatar machine translation system. *Machine Translation Summit XIV*, pages 175–182.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the WEB. *CoRR*, abs/1911.04944.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

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.

JL Song and L Dai. 2015. Construction of Uighur-Chinese parallel corpus. In *Multimedia, Communication and Computing Application: Proceedings of the 2014 International Conference on Multimedia, Communication and Computing Application (MCCA 2014), Xiamen, China, October 16-17, 2014*, page 353. CRC Press.

Aida Sundetova, Mikel Forcada, and Francis Tyers. 2015. A free/open-source machine translation system from english to kazakh. In *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE" TURKIC LANGUAGES PROCESSING" TurkLang-2015*, pages 78–90.

Ahmet Cüneyd Tantuğ, Eşref ADALI, and Kemal OFLAZER. 2011. Türkmenceden türkçeye bilgisayarlı metin çevirisi. *İTÜDERGİSİ/d*, 7(4).

A. Cüneyd Tantuğ and Eşref Adalı. 2018. Machine translation between turkic languages. In Kemal Oflazer and Murat Saraçlar, editors, *Turkish Natural Language Processing*, pages 237–254. Springer.

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge–Realistic Data Sets for Low Resource and Multilingual MT. *arXiv preprint arXiv:2010.06354*.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ualsher Tukeyev, Aidana Karibayeva, and Balzhan Abduali. 2019. Neural machine translation system for the kazakh language based on synthetic corpora. In *MATEC Web of Conferences*, volume 252, page 03006. EDP Sciences.

Cigdem Keyder Turhan. 1997. An English to Turkish machine translation system using structural mapping. In *Fifth Conference on Applied Natural Language Processing*, pages 320–323.

Francis M Tyers, Jonathan North Washington, Ilnar Salimzyanov, and Rustam Batalov. 2012. A prototype machine translation system for Tatar and Bashkir based on free/open-source components. In *First Workshop on Language Resources and Technologies for Turkic Languages*, page 11.

Aidar Valeev, Ilshat Gibadullin, Albina Khusainova, and Adil Khan. 2019. Application of Low-resource Machine Translation Techniques to Russian-Tatar Language Pair. *arXiv preprint arXiv:1910.00368*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Dongqi Wang, Zihan Liu, Qingnan Jiang, Zewei Sun, Shujian Huang, and Jiajun Chen. 2020. NJUNLP's Machine Translation System for CCMT-2020 Uighur - Chinese Translation Task. In *China Conference on Machine Translation*, pages 76–82. Springer.

Jonathan North Washington, Ilnar Salimzianov, Francis M. Tyers, Memduh Gökırmak, Sardana Ivanova, and Oğuz Kuyrukçu. 2019. Free/open-source technologies for Turkic languages developed in the Apertium project. In *Proceedings of the International Conference on Turkic Language Processing (TURKLANG 2019)*.

Extending Challenge Sets to Uncover Gender Bias in Machine Translation Impact of Stereotypical Verbs and Adjectives

Jonas Troles
Cognitive Systems
University of Bamberg
jonas.troles@uni-bamberg.de

Ute Schmid
Cognitive Systems
University of Bamberg
ute.schmid@uni-bamberg.de

Abstract

Human gender bias is reflected in language and text production. Because state-of-the-art machine translation (MT) systems are trained on large corpora of text, mostly generated by humans, gender bias can also be found in MT. For instance when occupations are translated from a language like English, which mostly uses gender neutral words, to a language like German, which mostly uses a feminine and a masculine version for an occupation, a decision must be made by the MT System. Recent research showed that MT systems are biased towards stereotypical translation of occupations. In 2019 the first, and so far only, challenge set, explicitly designed to measure the extent of gender bias in MT systems has been published. In this set measurement of gender bias is solely based on the translation of occupations. With our paper we present an extension of this challenge set, called *WiBeMT*[1], which adds gender-biased adjectives and sentences with gender-biased verbs. The resulting challenge set consists of over $70,000$ sentences and has been translated with three commercial MT systems: DeepL Translator, Microsoft Translator, and Google Translate. Results show a gender bias for all three MT systems. This gender bias is to a great extent significantly influenced by adjectives and to a lesser extent by verbs.

1 Introduction

The problem of unfair and biased models has been recognized as an important problem for many applications of machine learning (Mehrabi et al., 2019). The source of unfairness typically is based on biases in the training data. In many domains, unfairness is caused by sampling biases. In machine translation (MT), however, the main source of unfairness is due to historical or social biases when there is a misalignment between the world as it is and the values or objectives to be encoded and propagated in a model (Suresh and Guttag, 2019; Bolukbasi et al., 2016). One source of imbalance in natural language is the association of specific occupations with gender. Typically, occupations in more technical domains as well as occupations with high social status are associated with male gender (Cheryan et al., 2017).

In natural language processing (NLP), gender bias has been investigated for word embeddings (Bolukbasi et al., 2016). Analogy puzzles such as "man is to king as woman is to x" generated with *word2vec*[2] yield $x = queen$ while for "man is to computer programmer as woman is to x", the output is $x = homemaker$. Only few publications exist that directly address gender bias and MT. Although gender bias is particularly relevant for translations into gender-inflected languages, for instance from English to German, where biased models can result in translation errors (Saunders and Byrne, 2020). For the English sentence "The doctor told the nurse that she had been busy.", a human translator would resolve the co-reference of 'she' to the doctor, correctly translating it to 'die Ärztin'. However, a neural machine translation model (NMT) trained on a biased dataset, in which most doctors are male might incorrectly default to the masculine form, 'der Arzt'. Hovy et al. (2020) investigated the prediction of age and gender of a text's author before and after translation. They found that *Bing Translator*, *DeepL Translator*, and *Google Translate* all skew the predictions to be older and more masculine, which shows that NMT systems and the expression of gender interact.

According to Saunders and Byrne (2020), the first systematic analysis of gender bias in MT is from Stanovsky et al. (2019). They introduce *WinoMT* which is the first challenge set explicitly designed to quantify gender bias in MT sys-

[1] Our test set and related data is available at: www.github.com/JDtroles/WiBeMTdata.git

[2] www.code.google.com/archive/p/word2vec/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 531–541
November 10–11, 2021.

tems. Furthermore, they introduce an automatic evaluation method for eight different languages. Evaluating six MT systems, they found a strong preference for masculine translations in all eight gender-inflected languages. The above example sentence, where 'doctor' has been errouneously translated into its male German form (*Arzt*) is one of the $3,888$ sentences used in *WinoMT*.

WinoMT focuses solely on the translation of occupations to evaluate gender bias. In this paper, we present the extended data set *WiBeMT* to uncover gender bias in neural machine translation systems. Gender stereotypical occupations are augmented by gender stereotypical adjectives and verbs and we investigate how congruent and incongruent combinations impact translation accuracy of the NMT systems DeepL, Microsoft Translator and Google Translate. In the next section, the original *WinoMT* data set is introduced together with our extensions. Afterwards, results for the three NMT systems based on 70,686 sentences are presented, followed by a discussion and an outlook.

2 Constructing the Extended Challenge Set WiBeMT

To construct a more diverse challenge set, we extend *WinoMT*, respectively its core data base *WinoBias*. Therefore, we identify verbs and adjectives of high stereotypicality with respect to gender. A gender score is determined by the cosine similarity of these words to a list of gender specific words. To calculate the similarity different pretrained word embeddings are used, where each of these words is represented by a vector. With the resulting most feminine and masculine adjectives the original sentences of *WinoBias* are extended. Furthermore, new sentences are created combining occupations with gender stereotypical verbs.

2.1 WinoBias and its Extension

WinoMT is based on two previous challenge sets, *Winogender* (Rudinger et al., 2018) and *WinoBias* (Zhao et al., 2018). Both were introduced to quantify gender bias in co-reference resolution systems. Since *WinoBias* constitutes 81.5% of *WinoMT*, we use it as basis for our extension. In total *WinoBias* consists of $3,168$ sentences, of which $1,582$ are feminine and $1,586$ are masculine sentences. The sentences are based on 40 occupations. An example sentence of *WinoBias* involving a cleaner and a developer is:

- WinoBias sentence: *The cleaner hates the* ***developer*** *because* ***she*** *always leaves the room dirty.*
- DeepL translation: *Die Reinigungskraft hasst* ***den Entwickler****, weil* ***sie*** *das Zimmer immer schmutzig hinterlässt.*

DeepL fails to correctly translate **developer** to the female inflection **die Entwicklerin**, but instead favors the stereotypical male inflection **der Entwickler**.

The *WinoBias* set is constructed such that each sentence is given in a stereotypical and an anti-stereotypical version. Stereotypical in this context means that the gender of the pronoun matches the predominant gender in the sentences occupation. The example sentence above is the anti-stereotypical version for the occupational noun 'developer', where the stereotypical version contains a 'he' instead of a 'she'.

We argue that a measurement of gender bias solely based on the translation of occupational nouns does not do justice to the complexity of language. Therefore, we want to diversify the given approach by taking gender-stereotypical adjectives and verbs into account. This is realized in two steps: First, *WinoBias* sentences are extended such that each occupational noun is preceded with an adjective which is congruent or incongruent with respect to the gender stereotypical interpretation of the occupation. For instance, a developer might be *eminent* (male) or *brunette* (female). By adding feminine and masculine adjectives to each WinoBias sentence, we create a new subset of extended WinoBias sentences.

Second, we create completely new sentences based on feminine and masculine verbs. One example with a feminine verb being: "The X **dances** in the club", and one with a masculine verb: "The X **boasts** about the new car.". Those base sentences are then extended with 99 occupations from Zhao et al. (2018), Rudinger et al. (2018), and Garg et al. (2018), resulting in a new subset of verb sentences. All 100 occupational nouns used in *WinoBias* and in the new verb sentences are listed in Table 1 which is based on numbers from the US Bureau of Labor Statistics[3]. After concatenating both subsets, the complete extended gender-bias challenge set consists of $70,686$ sentences. Overall, 100 occupa-

[3]Labor Force Statistics from the Current Population Survey: www.bls.gov/cps/cpsaat11.htm

Table 1: All 100 occupations, used for this work with the corresponding percentage of women in the occupation in the US. Numbers with an asterisk are taken from the WinoBias paper and are from 2017 (Zhao et al., 2018). All other numbers are from 2019 from the US Bureau of Labor Statistics. Occupations in bold font are used in the WinoBias challenge set. All occupations without corresponding percentage were not listed by the US Bureau of Labor Statistics.

Occupation	%	Occupation	%	Occupation	%	Occupation	%	Occupation	%
electrician	2	athlete	35	pharmacist	60	**receptionist**	**89**	examiner	–
carpenter	**3**	**lawyer**	**36**	**accountant**	**62**	**nurse**	**90***	gardener	–
firefighter	3	**janitor**	**37**	**auditor**	**62**	paralegal	90	geologist	–
plumber	3	musician	37	**editor**	**63**	dietitian	92	hygienist	–
construction-worker	**4**	**CEO**	**39***	**writer**	**63***	**hairdresser**	**92**	inspector	–
laborer	**4***	**analyst**	**41***	author	64	nutritionist	92	investigator	–
mechanic	**4***	**physician**	**41**	instructor	65	**secretary**	**93**	mathematician	–
driver	**6***	surgeon	41	veterinarian	68	administrator	–	officer	–
machinist	6	**cook**	**42**	**cashier**	**71**	advisor	–	pathologist	–
painter	9	chemist	43	**clerk**	**72***	appraiser	–	physicist	–
mover	**18***	**manager**	**43***	**tailor**	**75**	broker	–	practitioner	–
sheriff	**18**	**supervisor**	**44***	**attendant**	**76***	CFO	–	professor	–
developer	**20***	**salesperson**	**48***	**counselor**	**76**	collector	–	sailor	–
programmer	20	photographer	49	**teacher**	**78***	conductor	–	scientist	–
guard	**22***	bartender	53	planner	79	CTO	–	soldier	–
architect	25	dispatcher	53	**librarian**	**80**	dancer	–	specialist	–
farmer	**25**	judge	53	psychologist	80	doctor	–	student	–
chief	**28**	artist	54	**assistant**	**85***	economist	–	surveyor	–
dentist	34	**designer**	**54**	**cleaner**	**89***	educator	–	technician	–
paramedic	34	**baker**	**60**	**housekeeper**	**89***	engineer	–	therapist	–

tions are used in the set, 42 gender-verbs, and 20 gender-adjectives.

We hypothesize that gender-stereotypical verbs and adjectives influence the gender of the translation of the occupational nouns, when translating from English to the gender-inflected language German:

Hypothesis 1

1 Sentences with a feminine verb result in significantly more translations into the female inflection of an occupation than sentences with a masculine verb.

Hypotheses 2

2a WinoBias sentences extended with a feminine adjective result in significantly more translations into the female inflection of an occupation than original WinoBias sentences (without a preceded adjective).

2b WinoBias sentences extended with a masculine adjective result in significantly more translations into the masculine inflection of an occupation than original WinoBias sentences (without a preceded adjective).

2.2 Finding Gender-Stereotypical Verbs and Adjectives

To determine gender-stereotypicality of adjectives and verbs, large collections of these word types have been scored with respect to their similarity to a list of gender specific words given by Bolukbasi et al. (2016). As input we used a list of 3.250 verbs from patternbasedwriting.com[4] and a combined list of 4.889 adjectives from patternbasedwriting.com and from Garg et al. (2018).

Calculation of gender score is based on word embeddings and the cosine-similarity, following the work of Bolukbasi et al. (2016) and Garg et al. (2018). In their work similarity of the words to the pronouns "she" and "he" has been used. We extended scoring using a longer list of feminine and masculine words such as "mother", "uncle", "menopause" or "semen" to enhance robustness of the gender-score. This list, containing 95 feminine and 108 masculine words, is taken from Bolukbasi et al. (2016), who used it for their debiasing methods of word embeddings.

Two families of *word embeddings* were used: two pre-trained versions of *fastText*[5] from Mikolov et al. (2017) and two pre-trained versions of *GloVe*[6] from Pennington et al. (2014). All four word embeddings have a vector size of 300 dimensions.

[4] www.patternbasedwriting.com offers teaching materials for primary school children.

[5] Downloaded from: www.fasttext.cc/docs/en/english-vectors.html

[6] Downloaded from: www.nlp.stanford.edu/projects/glove/

Table 2: Summary of the origin of training corpora and their corresponding size for each word embedding.

	Size [billion]			
corpora	fastText-small	fastText-large	GloVe-small	GloVe-large
Wikipedia 2014	—	—	1.6	—
Gigaword 5	—	—	4.3	—
Wikipedia meta-page 2017	9.2	—	—	—
Statmt.org News	4.2	—	—	—
UMBC News	3.2	—	—	—
Common Crawl	—	630	—	840

Table 2 gives an overview of all word embeddings and their training data.

Cosine Similarity is a measure of similarity between two normalized and non-zero vectors and can take values in the range between -1 and 1. Equation 1 shows the calculation of the cosine similarity between the vectors $\mathbf{a}$ and $\mathbf{b}$ of two words:

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \quad (1)$$

where a_i and b_i are components of vector $\mathbf{a}$ and $\mathbf{b}$ respectively.

Since word embeddings inherit to some extend the meaning of words, it is possible to use the cosine similarity as a measure for the similarity in meaning or, furthermore, the relationships between words. This enables mathematical operations on the vectors representing words such that: $cos(\overrightarrow{brunette} \cdot \overrightarrow{her}) \geq cos(\overrightarrow{brunette} \cdot \overrightarrow{him})$ becomes true for "brunette" and other gender-biased adjectives. If the feminine-gender value ($cos(\overrightarrow{brunette} \cdot \overrightarrow{her})$) is then subtracted from the masculine-gender value ($cos(\overrightarrow{brunette} \cdot \overrightarrow{him})$) the resulting single float value indicates whether a word is gender-biased in the word embedding with which the cosine-similarity was computed.

The total gender-score is the sum of eight single scores resulting from the combination of the four different word embeddings with the cosine similarity with "she" and "he" and with the list of feminine and masculine words. Since different word embeddings vary in the strength of the inherited gender-bias and all word embeddings should have equal impact on the overall score, the interim results were normalized to fit a range between $a = -1$ and $b = 1$, as Equation 2 shows:

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)} \quad (2)$$

To validate the procedure to determine a gender score for adjectives and verbs, the same method has been applied to the occupations given in Table 1. The gender-score for these occupations shows a strong correlation to the percentage of women working in each given occupation (see Figure 1).

Gender-score has been calculated for all verbs and adjectives for which word embeddings existed. They were sorted ascending from the most negative – and therefore most feminine – score value, to the most positive, i.e., most masculine, value.

Verbs: Of the $3,250$ *verbs*, $3,210$ could be ranked ($med = 0.151$, $std = 0.165$). After sorting the verbs by their gender-score, the most stereotypical verbs which could be used in a sentence in which person P actively does action A were picked. The gender score of the 21 selected feminine verbs ranges from -0.772 to -0.233 and of the masculine verbs from 0.445 to 0.733:

- Feminine verbs: crochet, sew, accessorize, bake, embroider, primp, gossip, shriek, dance, undress, milk, giggle, marry, knit, twirl, wed, flirt, allure, shower, seduce, kiss.
- Masculine verbs: draft, tackle, swagger, trade, brawl, reckon, preach, sanction, build, boast, gamble, succeed, regard, retire, chuck, overthrow, rev, resign, apprehend, appoint, fool.

Adjectives: Of the $5,441$ *adjectives*, $4,762$ could be ranked ($med = 0.189$, $std = 0.142$). After calculating the gender-score for the $4,762$ adjectives, beginning with the most feminine, respectively, most masculine, adjectives were tested for their suitability considering the extension of existing WinoBias sentences until 10 most feminine and suitable as well as 10 most masculine and

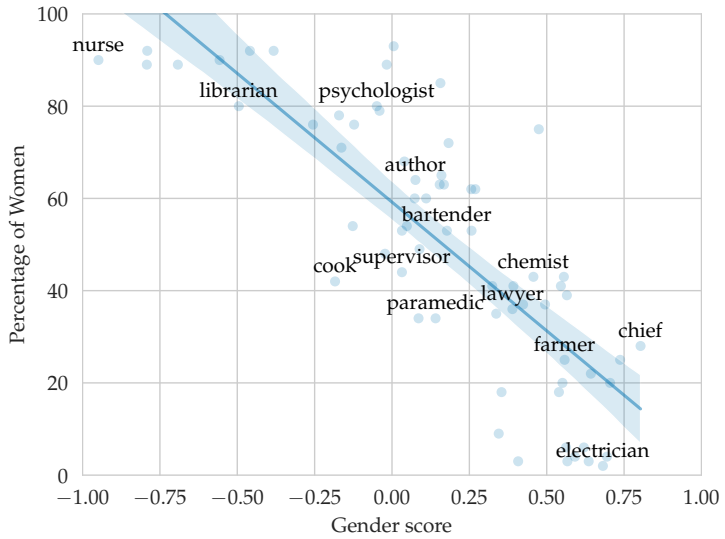


Figure 1: Scatter plot of the 99 single word occupations with the percentage of women in the profession y-axis and the gender score on the x-axis.

suitable adjectives had been selected. Words that semantically could not be combined with occupations were discarded (examples: hormonal, satin, luminous, philosophical, topographical). Furthermore, adjectives like "pregnant" – which only apply to persons with a uterus – were also discarded. The gender score of the 10 selected feminine adjectives ranges from -0.600 to -0.205 and of the masculine adjectives from 0.480 to 0.654:

- Feminine adjectives: sassy, perky, brunette, blonde, lovely, vivacious, saucy, bubbly, alluring, married.
- Masculine adjectives: grizzled, affable, jovial, suave, debonair, wiry, rascally, arrogant, shifty, eminent.

2.3 Translation of *WiBeMT* and Evaluation Design

To test the extent to which machine translation systems inherit a gender-bias, three different services were tested with *WiBeMT*: *Google Translate*, *Microsoft Translator*, and *DeepL Translator*. All three NMT systems were accessed via their API, and all translation processes took place in July 2020.

To test our hypotheses that adjectives and verbs significantly influence the gender in translations of NMT systems, translations from English to German have to be categorized as either (correctly) feminine, masculine, neutral, or wrong. Stanovsky et al. (2019) use an automated method in the form of different morphological analyzers such as *spaCy* to determine the gender of the occupation in the translated sentence. While this method is convenient for an automated approach that other researchers can use with different data, it also suffers from a certain degree of inaccuracy. To measure the accuracy of their automated evaluation method, Stanovsky et al. (2019) compared the automated evaluations with a random sample of samples evaluated by native speakers. They found an average agreement of 87% between automated and native speaker evaluations. To evaluate NMT systems with respect to *WiBeMT*, we preferred to augment automated evaluation by "manual" evaluation to gain higher accuracy.

Evaluating the gender of all translations is based on a nested list, we refer to as *classification-list*, and a set of rules for automated evaluation and manual evaluation for all remaining ambiguous translations. The classification-list contains four sublists for each occupation: a sublist of correct feminine translations, a sublist of correct masculine translations, a sublist of correct neutral translations, and a sublist of inconclusive or wrong translations. The gender of the translated occupation is then classified by checking in which sublist the occupation is listed and the gender is labeled as the sublist's category.

To control classifications for possible errors, $N = 665$ (1%) of the WinoBias sentences extended by adjectives were manually controlled for

each translation system (in total $N = 1,995$ sentences). Not a single one was miscategorized. Of the verb-sentences translations, even 5% were manually controlled by the authors, and here also, not a single one of the 618 translations was miscategorized.

3 Results

After the creation of the *WiBeMT* challenge set and the translation of all $70,686$ sentences with each NMT system, the translations were categorized as *feminine*, *masculine*, *neutral*, or *inconclusive / wrong*. The latter will be referred to as "wrong". Due to these discrete categories the Chi^2 test of independence will be used for all statistical tests. As the extended *WinoBias* sentences include a pronoun that defines the gender, the results considering these data can be divided into *true* and *false*, and *feminine* and *masculine* translations. On the other hand, the verb sentences do not include any cue for a *correct* gender in the translation. Therefore, they are just analyzed for feminine and masculine translations. The calculated *feminine-ratio* ($\%TFG$) results from all feminine-translations divided by the sum of feminine- and masculine-translations. The calculated *correct-gender-ratio* ($\%TCG$) results from all translations with correct gender divided by the sum of all translations with correct and incorrect gender. All Chi^2-tests were, if necessary, Bonferroni corrected. First, the results for the verb-sentences; Second, the results for the extended WinoBias sentences; And third, further results are presented.

3.1 Hypothesis 1: Verb Sentences

The statistical tests regarding Hypothesis 1 yielded mixed results, depending on which NMT system is looked at. Of the 2.079 sentences with a feminine verb DeepL translated the occupations in 242 (11.9%) sentences into the female gender ($\%TFG$) compared to Microsoft Translator with 149 (7.5%), and Google Translate with 120 (6.0%). Of the 2.079 sentences with a masculine verb DeepL (DL) translated the occupations in 135 (6.6%) sentences into the female gender compared to Microsoft Translator (MS) with 104 (5.2%), and Google Translate (GT) with 109 (5.4%). For DeepL and Microsoft Translate the difference becomes significant with $\Delta\%TFG_{DL} = 5.3\%$, $Chi^2_{DL} = 33.7$ and $p_{DL} < 0.001$, and $\Delta\%TFG_{MS} = 2.3\%$, $Chi^2_{MS} = 8.4$ and $p_{MS} = 0.004$. For Google Translate the difference does not become significant. As two of three NMT systems inherit a gender bias that is influenced as expected by verbs, meaning that translations of sentences with feminine verbs result in a significantly higher $\%TFG$ than translations of sentences with masculine verbs, $H1$ is accepted. Figure 2 gives an overview of the translations to the female gender ($\%TFG$) for all three NMT systems and the two categories of verb sentences.

3.2 Hypothesis 2: Extended WinoBias Sentences

Both hypotheses assume that adjectives that inherit a gender-bias in word embeddings also influence the gender of translations from English to German. While the results confirm the assumptions of Hypothesis $H2_a$, they also yield unexpected results for Hypothesis $H2_b$. Of the 3,168 original WinoBias sentences DeepL translated $1,259$ (41.7%), Microsoft Translator translated $1,041$ (35.8%), and Google Translate translated 976 (33.2%) to the female inflection of the occupation. These percentages are used as the baseline to test, whether feminine and masculine adjectives skew the outcome of translations into the expected direction.

Of the sentences extended with a feminine adjective DeepL translated 49.5%, Microsoft Translator translated 42.5%, and Google Translate translated 40.1% to the female inflection. The difference to the percentage of translations to the female inflection of the original WinoBias sentences becomes significant for all three NMT systems: $\Delta\%TFG_{DL} = 7.8\%$, $Chi^2_{DL} = 66.3$ and $p_{DL} < 0.001$; $\Delta\%TFG_{MS} = 6.7\%$, $Chi^2_{MS} = 49.0$ and $p_{MS} < 0.001$; and $\Delta\%TFG_{GT} = 6.9\%$, $Chi^2_{GT} = 53.8$ and $p_{GT} < 0.001$. Therefore Hypothesis $H2_a$ is accepted.

Of the sentences extended with a masculine adjective DeepL translated 44.6%, Microsoft Translator translated 39.3%, and Google Translate translated 37.9% to the female inflection. The difference to the percentage of translations to the female inflection of the original WinoBias sentences becomes significant for all three NMT systems: $\Delta\%TFG_{DL} = 2.9\%$, $Chi^2_{DL} = 9.1$ and $p_{DL} = 0.008$; $\Delta\%TFG_{MS} = 3.5\%$, $Chi^2_{MS} = 13.3$ and $p_{MS} < 0.001$; and $\Delta\%TFG_{GT} = 4.7\%$, $Chi^2_{GT} = 25.1$ and $p_{GT} < 0.001$. While all differences are significant, they contradict our assumption that preceding masculine adjectives to Wino-

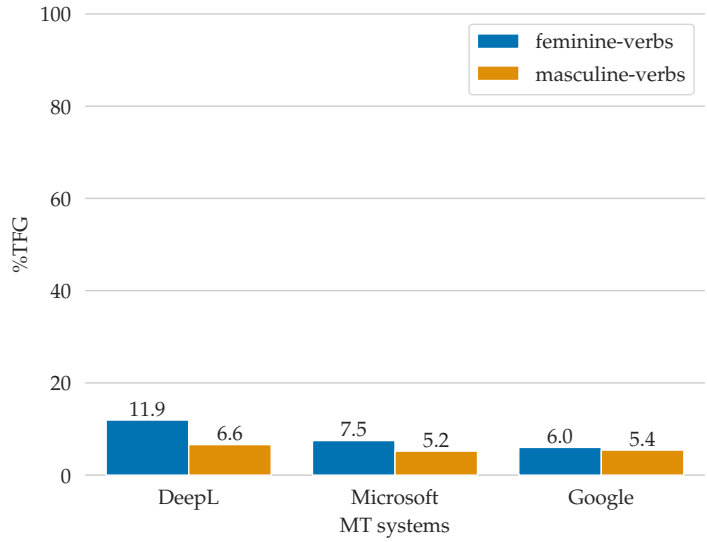


Figure 2: Percentage of translations to female gender ($\%TFG$) of occupations in verb sentences, organized by the category of verbs.

Bias sentences results in less translations to the female inflection. Instead the preceded masculine adjectives have the opposite effect. Therefore, Hypothesis $H2_b$ is rejected. Figure 3 gives an overview of the translations to the female gender ($\%TFG$) for all three NMT systems and the three categories of sentences. Table 3 lists all numbers of different translation categories for a deeper insight.

3.3 Influence of Gender-Stereotypical Verbs and Adjectives on Translations

Our findings show that gender stereotypical verbs and adjectives influence gender bias in translations of NMT systems. In the following, we discuss our results in more detail.

Hypothesis 1: In general, the results from verb sentences differ drastically from the ones of the extended WinoBias sentences, with far fewer sentences being translated to their feminine gender. The percentage of sentences translated to their feminine gender ($\%TFG$) in the verb sentences ranges from 5.2% for masculine verb sentences translated by Microsoft Translator, to 11.9% for feminine verb sentences translated by DeepL. In comparison to that, $\%TFG$ ranges from 33.2% in original WinoBias sentences without adjective translated by Google Translate, to 49.5% in extended WinoBias sentences with feminine adjectives translated by DeepL Translator.

Two reasons could be responsible for the low $\%TFG$. Firstly, the generic masculine in German: As mostly the masculine gender is used to address all genders, this must be present in the training data of all three NMT systems. Therefore, they tend to use the male inflection whenever the gender bias for a specific occupation does not outweigh the bias by the generic masculine. Secondly, the verb-sentences lack a gender pronoun like "her" or "him", which urges the NMT system to decide which gender would be correct in the translation. This probably leads to the strong bias towards male inflections in translations.

To check whether the bias induced by occupations and the generic masculine outweighed an existing bias of verbs, we analyzed the data of the verb sentences again, looking at the different groups of occupations[7]: feminine, neutral and masculine. For all three NMT systems the $\%TFG$ for verb sentences with neutral and masculine occupations was below 2% regardless of the stereotypical feminine verbs. The $\%TFG$ in sentences with female occupations was 25.1% in DeepL, 19.0% in Microsoft Translator and 17.6% in Google Translate. All differences to sentences with neutral and masculine occupations were significant with p-values below 0.001 and Chi^2 values reaching from 352 to 256.

Hypotheses 2a & 2b: As gender bias works both ways in word embeddings, meaning that words can be stereotypical feminine or stereotypical mascu-

[7]With the calculated gender score we split the list of occupations in three equally sized categories (each $n = 33$)

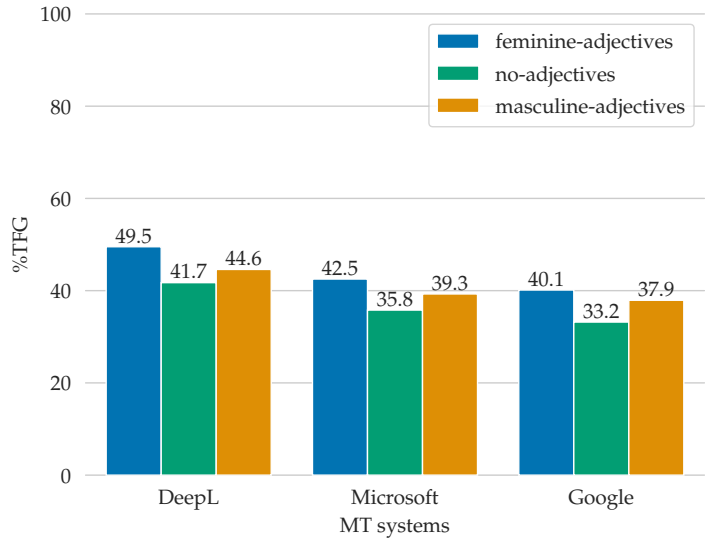


Figure 3: Percentage of translations to female gender ($\%TFG$) of occupations in all three types of EWB sentences: feminine adjective, masculine adjective, and no adjective.

line, we assumed that, depending on their gender-score derived from word embeddings, feminine adjectives would skew translations of NMT systems more often to their feminine gender and vice versa that masculine adjectives would skew translations more often to their masculine gender. This assumption was also supported by the findings of Stanovsky et al. (2019), who preceded "handsome" to occupations of sentences which would be correct, if translated to their masculine gender, and "pretty" to occupations of sentences which would be correct if translated to their feminine gender. With this measure, they could improve the accuracy of NMT systems and reduce gender bias.

Contrary to the prior assumptions, adjectives preceded to occupations led to significantly more translations with feminine gender, regardless of their gender-score derived from word embeddings. Finding that masculine adjectives also lead to more translations with feminine gender does not only result in the rejection of $H2_b$, but also weakens the acceptance of $H2_a$. Therefore, we introduce a new Hypothesis $H2_c$: "WinoBias sentences extended with a feminine adjective result in significantly more translations into the female inflection of an occupation than WinoBias sentences extended with a masculine adjective.".

The difference between the percentage of translations to the female inflection of WinoBias sentences extended with feminine and WinoBias sentences extended with masculine adjectives becomes significant for all three NMT systems: $\Delta\%TFG_{DL} = 4.9\%$, $Chi^2_{DL} = 147.6$ and $p_{DL} < 0.001$; $\Delta\%TFG_{MS} = 3.2\%$, $Chi^2_{MS} = 59.1$ and $p_{MS} < 0.001$; and $\Delta\%TFG_{GT} = 2.2\%$, $Chi^2_{GT} = 29.2$ and $p_{GT} < 0.001$. Therefore, Hypothesis $H2_c$ is accepted which strengthens $H2_a$, as stereotypical feminine adjectives preceded to occupational nouns lead to significantly more translations to the female gender than stereotypical masculine adjectives preceded to occupational nouns.

3.4 Influence of Adjectives on Correct Gender in Translations

The comparison of all three NMT systems was not one of the main research questions. Nevertheless, the extent of gender bias in the NMT systems can be of interest to users. Therefore, and because of the surprising findings considering Hypothesis 2_b a short comparison is presented in the following paragraphs.

To better understand our results regarding hypotheses $H2_a$, $H2_b$ and $H2_c$ we plotted percentage of translations with the correct gender ($\%TCG$) organized by NMT systems and the type of sentence: original WinoBias sentence (no adjective), with masculine adjective extended WinoBias sentence (M adjective), and with feminine adjective extended WinoBias sentence (F adjective). Additionally, we split each type of sentence into sentences with male pronouns (M pronouns) and sentences with female pronouns (F pronouns). Figure

Table 3: Numbers of the categorizations of the gender of translations of extended WinoBias sentences sorted by the gender-class of the adjective.

		NMT system		
adjective gender	translation category	DeepL	Microsoft	Google
feminine	true feminine	**13,341**	11,352	10,177
	false masculine	1,553	3,405	**4,593**
	true masculine	**13,833**	13,814	13,312
	false feminine	1,751	1,382	**1,826**
	neutral	**550**	440	472
	wrong	652	1,287	**1,300**
masculine	true feminine	**12,497**	8,966	8,724
	false masculine	2,378	3,405	**4,533**
	true masculine	**14,425**	11,943	12,128
	false feminine	1,028	963	**1,450**
	neutral	**483**	346	447
	wrong	869	**6,057**	4,398
no adjective	true feminine	**1,162**	916	812
	false masculine	324	503	**652**
	true masculine	**1,434**	1,365	1,313
	false feminine	97	125	**164**
	neutral	75	**77**	53
	wrong	76	**182**	174

4 shows the resulting plot.

The results are quite astonishing: while the $\%TCG$ slightly decreases for sentences with male pronouns and any adjective ($1 \leq \Delta\%TCG \leq 5$) it drastically improves for sentences with female pronouns and any adjective ($12 \leq \Delta\%TCG \leq 13$). Considering that sentences with female and male pronouns are equally prevalent in original WinoBias sentences and extended WinoBias sentences, it shows that preceding an occupational noun with any adjective likely improves the overall percentage of translations to the correct gender inflection. This reflects the findings of Stanovsky et al. (2019), who preceded "handsome" and "pretty" to occupational nouns of WinoMT sentences. With this measure, they could improve the accuracy hence $\%TCG$ of NMT systems and reduce gender bias. Our findings give more insight to this result: Stanovsky et al. (2019) very likely could have added either "handsome" or "pretty" to all sentences, regardless of their pronoun and would nonetheless have been able to record a higher accuracy.

Furthermore, Figure 4 shows that DeepL Translator performed best when it comes to $\%TCG$. Google Translate and Microsoft Translator again show more similar results, but Google Translate performed notably worse than Microsoft Translator. With the least discrepancy in $\%TCG$ between sentences with a feminine pronoun and sentences with a masculine pronoun in all three conditions (feminine adjective, masculine adjective, no adjective) DeepL Translator, therefore, inherits the lowest gender bias.

4 Conclusions and Further Work

The three neural machine translation systems evaluated with respect of their gender bias are black boxes, in so far as the architecture is not publicly available and – even more important – it is not transparent on what data these systems are trained. It is most likely that the gender bias in all three systems is inherited from the data used for training and their use of word embeddings.

To give a closer look on gender-bias of the NMT systems DeepL, Microsoft, and Google Translate, an extension of the *WinoMT* challenge set – the first challenge set designed to measure gender bias in NMT systems – has been presented. While *WinoMT* relies solely on the gender of the translation of occupations, our extended set *WiBeMT* includes gender-adjectives and gender-verbs. Thereby, a more detailed assessment of gender bias has been possible. The number of sentences in our challenge set is, with over $70,000$ sentences, nearly 20 times as large as the original WinoMT challenge set. This makes it less prone to overfitting when used to evaluate or reduce gender bias in NMT systems.

We could show that adjectives do significantly

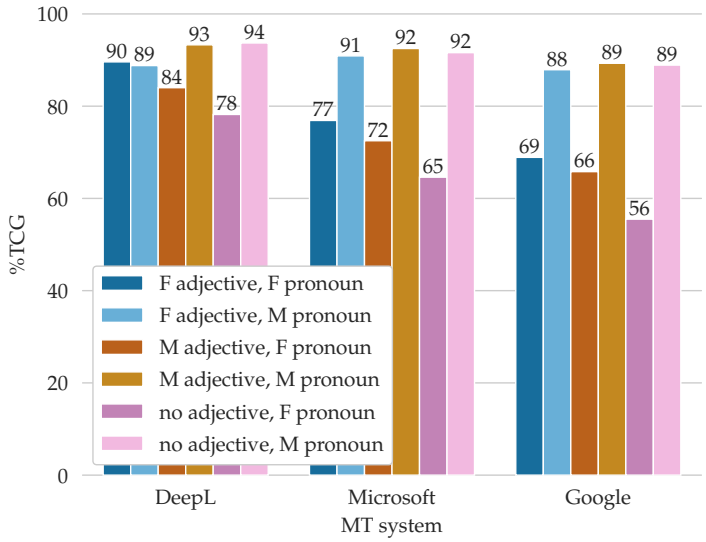


Figure 4: Bar plot of the percentage of translations into the correct gender ($\%TCG$) of original WinoBias sentences and extended WinoBias sentences.

influence the gender of translated occupations. Against the hypotheses, feminine as well as masculine adjectives skew the translations of NMT systems to more translations with the feminine gender. Nevertheless, feminine adjectives still produce significantly more translations with the feminine gender than masculine adjectives do.

All three NMT systems prefer translations to the generic masculine when no pronoun defines a correct gender for the translation. Despite this preference, 5.7% to 9.3% of all verb sentences are translated to their feminine gender by the MT systems. This can mostly be attributed to a gender bias in translating occupations, as our results regarding verb sentences and occupation categories show. The effect of gender-verbs on the gender of translations only became significant in DeepL and Microsoft Translator and was smaller than the effect of the occupations gender-categories. It can be assumed that Google Translate only performed better on the verb sentences task, as it generally has the strongest tendency to translate sentences to their masculine gender.

Surprisingly, gender-adjectives drastically improve the overall accuracy of all three NMT systems when it comes to the correct gender in translations. While the accuracy slightly drops for sentences with masculine pronouns, it drastically improves for sentences with feminine pronouns. Therefore, the discrepancy in accuracy between masculine and feminine pronoun sentences decreases, resulting in lower discrimination. One could even argue that gender-adjectives reduce gender bias in the output of NMT systems, but at the same time, it can be discriminating in itself that, as soon as you add an adjective to describe a person, the instance of this person is more likely to be translated to its feminine gender. Further research is certainly needed to find reasons for this effect and to assess the potentials of discrimination.

The sentences in our *WiBeMT* challenge set are – as the original *WinoMT* challenge set – constructed in a systematic way. While this allows for a controlled experiment environment, it might also introduce some artificial biases (Stanovsky et al., 2019). A solution could be to collect real-world examples of sentences, which are suitable for gender bias detection. Furthermore, the limitation on one source language (English) and one target language (German) does not allow for a generalization of the results.

Another limitation of our study – as most other studies – is that it does not take into account that gender should not be seen as a binary, but rather a continuous variable. Cao and Daumé (2019), for example, outline why "trans exclusionary" co-reference resolution systems can cause harm, which is probably also valid for MT systems. A further extension of the challenge set could help to shine a light on the shortcomings of the inclusion of transgender persons of NMT systems.

References

Tolga Bolukbasi, Kai Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems*, pages 4356–4364.

Yang Trista Cao and Hal Daumé. 2019. Toward Gender-Inclusive Coreference Resolution. *arXiv preprint arXiv:1910.13913*.

Sapna Cheryan, Sianna A Ziegler, Amanda K Montoya, and Lily Jiang. 2017. Why are some STEM fields more gender balanced than others? *Psychological Bulletin*, 143(1):1–35.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences of the United States of America*, 115(16):E3635–E3644.

Dirk Hovy, Federico Bianchi, and Tommaso Fornaciari. 2020. "You Sound Just Like Your Father" Commercial Machine Translation Systems Include Stylistic Biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1686–1690. Association for Computational Linguistics.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A Survey on Bias and Fairness in Machine Learning. *arXiv preprint arXiv:1908.09635*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in Pre-Training Distributed Word Representations. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 52–55.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender Bias in Coreference Resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2, pages 8–14.

Danielle Saunders and Bill Byrne. 2020. Reducing Gender Bias in Neural Machine Translation as a Domain Adaptation Problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating Gender Bias in Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684.

Harini Suresh and John V Guttag. 2019. A Framework for Understanding Unintended Consequences of Machine Learning. *arXiv preprint arXiv:1901.10002*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. *arXiv preprint arXiv:1804.06876*.

Continual Learning in Multilingual NMT via Language-Specific Embeddings

Alexandre Bérard
NAVER LABS Europe
alexandre.berard@naverlabs.com

Abstract

This paper proposes a technique for adding a new source or target language to an existing multilingual NMT model without re-training it on the initial set of languages. It consists in replacing the shared vocabulary with a small language-specific vocabulary and fine-tuning the new embeddings on the new language's parallel data. Some additional language-specific components may be trained to improve performance (e.g., Transformer layers or adapter modules). Because the parameters of the original model are not modified, its performance on the initial languages does not degrade. We show on two sets of experiments (small-scale on TED Talks, and large-scale on ParaCrawl) that this approach performs as well or better as the more costly alternatives; and that it has excellent zero-shot performance: training on English-centric data is enough to translate between the new language and any of the initial languages.

1 Introduction

Multilingual Neural Machine Translation models are trained on multilingual data to translate from and/or into multiple languages (Firat et al., 2016; Johnson et al., 2017). Multilingual NMT is a compelling approach in production, as one only needs to train, deploy and maintain one model (instead of $2 \times N$ ones, where N is the number of languages). It has also been shown to improve MT quality for low-resource languages (at the cost of a slight degradation for high-resource languages) and it can allow translation between languages that have no aligned data ("zero-shot translation").

However, such models can be costly to train, as they usually involve larger architectures and large datasets. Moreover, because they are trained jointly on all the languages, they require to know in advance the full set of languages. Adding a new language to an existing model usually means re-training the model on the full multilingual dataset. Naively fine-tuning the original model on the new language's data is not an option because of vocabulary mismatch (the shared vocabulary needs to be modified to include the new language's tokens) and catastrophic forgetting (the model will quickly forget how to translate in the other languages).

In this paper, we study the problem of multilingual NMT *incremental training* or *continual learning* and propose a novel way to efficiently add a new source or target language.

Some desirable properties of an incremental training method are:

- No degradation on the existing language pairs;
- Efficient training (e.g., no re-training on the existing language pairs);
- Minimal amount of added parameters: the approach should scale to many languages and the model fit on a single GPU;
- Minimal degradation in inference speed;
- Good zero-shot performance: when training with X-EN (or EN-X) data, where X is a new language, we would like the model to be able to translate from X to any known language Y (resp. from Y to X).

We propose a novel technique for incrementally adding a new source or target language, which consists in substituting the shared embedding matrix with a language-specific embedding matrix, which is fine-tuned on the new language's data only while freezing the other parameters of the model. In some cases (e.g., when the new language is on the target size), a small number of additional parameters (e.g., adapter modules) have to be trained to match the performance of the re-training baseline. We perform two sets of experiments, with a 20-language Transformer Base trained on TED Talks, and a 20-language Transformer Big (with deep encoder and shallow decoder) trained on ParaCrawl; and show that this approach is fast and parameter-efficient and that it performs as well or better as the more costly alternatives.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 542–565
November 10–11, 2021.

2 Related work

Some previous works study how to adapt a multilingual MT model to unseen low-resource languages, but without seeking to maintain good performance in the initial languages (Neubig and Hu, 2018; Lakew et al., 2019). Garcia et al. (2021) introduce a "vocabulary substitution" approach for adding new languages to a multilingual NMT model. They create a new shared BPE vocabulary that includes the new language and initialize the embeddings of the overlapping tokens with their previous values. Then they fine-tune the entire model on the initial model's training data combined with parallel or monolingual data in the new language. Contrary to ours, their approach assumes access to the initial model's training data and results in a small performance drop in the existing languages.

Lyu et al. (2020) and Escolano et al. (2020, 2019, 2021) propose multi-decoder / multi-encoder architectures which they show to be compatible with incremental training. To add a new target (resp. source) language, one just has to freeze the model's encoder (resp. decoder) and train a new language-specific decoder (resp. encoder). However, this results in an enormous number of parameters.

Artetxe et al. (2020); Pfeiffer et al. (2021) incrementally train language-specific embeddings for cross-lingual transfer of BERT classification models. This approach consists of four stages: 1) train a monolingual BERT on language L_1; 2) train embeddings on language L_2 using the masked LM objective while freezing the other parameters; 3) fine-tune the L_1 BERT model on the desired classification task using labeled data in language L_1; 4) substitute the L_1 embeddings with the L_2 embeddings in the classification model and use it for L_2-language classification. Artetxe et al. (2020) also combine their approach with L_2-specific adapter layers and position embeddings. While this algorithm is close to ours, it is used on encoder-only Transformers for classification tasks. Our work extends this algorithm to encoder-decoder Transformers for multilingual MT.

Also similar to our technique, Thompson et al. (2018) do domain adaptation by freezing most of the NMT parameters and only fine-tuning one component (e.g., the source embeddings). Philip et al. (2020) show that adapter modules can be used to adapt an English-centric multilingual model to unseen language pairs, but whose source and target languages are known. We wanted to go further and use adapter layers to adapt a multilingual model to unseen languages. However, we obtained the surprising result that adapting the embedding matrix is sometimes enough. In the other cases, adapter modules can be used sparingly to match baseline performance. Üstün et al. (2021) introduce "denoising adapters" which they show can be used to incrementally adapt a multilingual MT model to new languages using monolingual data only.

3 Techniques

Figures 1 and 2 illustrate our technique for a new source and a new target language respectively.

The initial model is a many-to-many model with a shared vocabulary and source-side language codes (to indicate the target language).

3.1 New source language

To add a new source language (e.g., Greek), we build a new (smaller) vocabulary for this language only and replace the source embedding matrix with a new embedding matrix corresponding to that vocabulary. Note that some tokens may appear in both vocabularies. Similarly to Pfeiffer et al. (2021); Garcia et al. (2021), we initialize the new embeddings for those tokens with the existing embedding values. We train this new embedding matrix on Greek-English parallel data while freezing all the other parameters. There is no loss in performance in the existing languages as we do not modify the original parameters. At inference, to translate from the initial set of languages, we use the initial shared vocabulary and embeddings. To translate from Greek, we use the Greek embeddings and vocab.

To better adapt to the new source language, we also try combining this language-specific embedding matrix with other language-specific components in the encoder. We either fine-tune the first encoder layer while freezing the other layers, train the full encoder, or plug in adapter modules after encoder layers (Bapna and Firat, 2019) and train these while freezing the Transformer parameters.

Data augmentation As we will show in the experiments, source lang-specific parameters tend to give poor zero-shot results, i.e., when training them on Greek-English data, the resulting model might have trouble translating into other languages than English. For this reason, we try training such models on additional data. One solution is to use a multi-aligned Greek corpus (i.e., Greek paired with all the initial languages), but this might not always

be possible. We experiment with tiny amounts of such data (e.g., 1000 line pairs per initial language); and with synthetic data: translate the English side of the Greek-English corpus into the other languages with the initial model, then use the resulting fake line pairs for training. We call this approach "back-translation" (BT) even though it is arguably closer to distillation than back-translation because the synthetic text will be on the target side.

3.2 New target language

The same incremental training techniques can be used to learn a new target language (e.g., Greek) with some modifications. The decoder has a target embedding matrix and vocabulary projection matrix, which are usually tied and shared with the source embeddings (i.e., the same parameters are used for all 3 purposes). We need to adapt both the target embeddings and output projection to the new Greek vocabulary. Like in the initial model, we tie these two parameters. Additionally, the initial model does not have a "translate into Greek" language code. We add this language code to the source embedding matrix and freeze all source embeddings but this one. It is initialized with the "to English" language code embedding of the initial model. We combine this approach with language-specific parameters (adapter modules or fine-tuned Transformer layers) in the decoder and/or encoder.

3.3 New source and target languages

To translate between two new languages (e.g., Greek to Ukrainian), we train language-specific parameters for each of these languages separately, as described previously. Then, at inference time, we combine these parameters. This is done by taking the new source Greek embedding matrix and target Ukrainian embedding matrix (and vocabulary projection). The "translate into Ukrainian" language code embedding is concatenated to the Greek embedding matrix. Similarly, the combined model includes language-specific layers and adapters from both models. When both models have adapter modules at the same layers (e.g., last encoder layer), we stack them: the target-language adapters are plugged in after the source-language adapters.

3.4 Baselines

We compare our incremental training techniques with two types of baselines: bilingual models trained from scratch with only the new language's parallel data; and re-training, i.e., training a new multilingual model that includes the new language. To save computation time, similarly to Garcia et al. (2021), we start from the initial model and substitute its vocabulary with a new vocabulary trained with the same settings and data as before plus text in the new language. This ensures a large overlap between the old and new vocabularies. Then, we initialize the embeddings of the overlapping tokens with their previous values and fine-tune the full model on the entire dataset.

Note that these baselines do not meet our criteria for a good incremental training technique. Bilingual models are parameter-inefficient and cannot do zero-shot translation (except via pivot translation, which is twice as slow). Re-training assumes access to the initial model's training data and can be very slow. It could also result in a drop in performance in the initial languages.

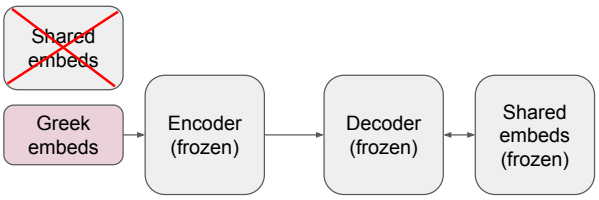


Figure 1: Adding a new source language with our incremental training technique. The source embedding matrix is replaced with the new language's embeddings and fine-tuned on the new language's data, while the other parameters are frozen.

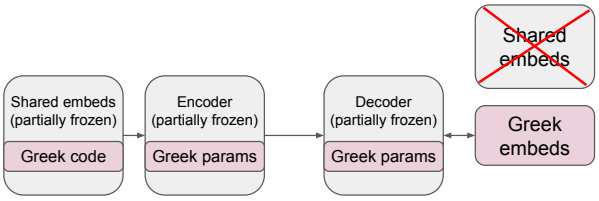


Figure 2: Adding a new target language with our incremental training technique. The tied target embedding matrix and output projection are replaced with the new language's embeddings. Some language-specific parameters can be added in the decoder or encoder, and a new language code is added in the source embedding matrix. Everything is kept frozen except for these new parameters.

4 TED Talks Experiments

We adapt a 20-language model trained on TED Talks to Greek (EL), either on the source side or target side. We pick Greek as the new language as it is from an unseen language family and uses an unseen alphabet. We also do experiments with Ukrainian

(UK), Indonesian (ID), or Swedish (SV) as the new language,[1] which are shown in Appendix.

4.1 Data and hyper-parameters

We use the TED Talks corpus (Qi et al., 2018) with the same set of 20 languages as Philip et al. (2020); Bérard et al. (2021).[2] This corpus is multi-parallel, i.e., it has training data for all 380 (20×19) language pairs. It also includes official valid and test splits for all these language pairs. Table 8 in Appendix shows the training data size per language.

The initial model is the "multi-parallel" baseline from Bérard et al. (2021), a Transformer Base (Vaswani et al., 2017) trained in two stages: English-centric training (38 directions) for 120 epochs; then multi-parallel fine-tuning (380 directions) for 10 epochs.[3] More hyper-parameters are given in Appendix (Table 23).

The shared vocabulary is created using BPE (Sennrich et al., 2016) with 64k merge operations and inline casing (Bérard et al., 2019). Both BPE and NMT training use temperature sampling with $T = 5$ (Arivazhagan et al., 2019). Single characters with a total frequency higher than 10 are added to the vocabulary. The Greek vocabulary is obtained with the same BPE settings but on Greek monolingual data with 4k merge operations. The bilingual baselines use a joint BPE model of size 8k and the same settings as in Philip et al. (2020). Our re-training baselines are obtained by creating a new shared BPE model of size 64k including all 20 initial languages plus Greek and fine-tuning the multi-parallel model for 10 more epochs with this vocabulary. Note that there is a vocabulary mismatch with the initial model (which did not have Greek). We initialize the known embeddings with their previous values and the new ones at random and reset the learning rate scheduler and optimizer. We also do a re-training baseline that includes all 4 new languages. Note that contrary to our incremental training approach, those models are trained with the new language(s) on both sides and use multi-aligned parallel data.

Finally, we train a model that follows more closely Garcia et al. (2021): we fine-tune the multi-parallel model for 10 epochs, by replacing the initial vocabulary with a vocabulary of the exact same size that includes Greek, and whose new tokens are initialized with the outdated embeddings from the old model. Like Garcia et al. (2021), we upscale the new data's sampling frequency by a factor of 5.

[1] They all use a known script (Latin or Cyrillic). Indonesian is from an unseen language family.

[2] {en, ar, he, ru, ko, it, ja, zh_cn, es, fr, pt_br, nl, tr, ro, pl, bg, vi, de, fa, hu}

[3] Note that an "epoch" when using multi-parallel data corresponds to approximately 9 English-centric epochs in terms of updates.

	Model	→EN	←EN	/ EN
1	SOTA – bilingual	32.4	**24.4**	15.0
2	SOTA – multilingual	30.9	22.3	14.8
3	English-centric	31.8	24.2	13.5
4	(**3**) + multi-parallel	32.8	24.3	16.3
5	(**4**) + EL	**33.3**	24.3	**16.6**
6	(**4**) + {EL,UK,SV,ID}	33.2	24.0	16.5

Table 1: BLEU scores (average to English, from English, and between non-English languages) of the baseline models on TED test. "SOTA" corresponds to the bilingual and multi-parallel baselines of Philip et al. (2020). (**3**) and (**4**) are from Bérard et al. (2021).

4.2 Evaluation settings

The TED Talks models are evaluated on the provided multi-parallel validation and test sets. Since those are already word-tokenized, we run SacreBLEU with the `--tok none` option.[4]

We report BLEU scores from/into English and average BLEU from/into the 19 other languages than English (which correspond to a zero-shot setting when the incremental training is done on Greek-English only data). We also report chrF scores obtained with SacreBLEU on the test and validation sets in Appendix.[5]

4.3 Results and analysis

Table 29 in Appendix details the notations used in this paper and the tables.

Baselines. Table 1 compares our initial models and re-training baselines against the state of the art on the initial set of 20 languages. In this instance, fine-tuning the initial model with more languages (**5**, **6**) does not degrade BLEU. Appendix Table 9 shows valid and test chrF on more baselines, including our implementation of the vocabulary substitution approach of Garcia et al. (2021).

New source language. Table 2 shows the test BLEU scores of several incrementally-trained models with Greek as a new source language. More re-

[4] SacreBLEU signature: `BLEU+c.mixed+#.1+s.exp+tok.none+v.1.5.1`

[5] `chrF2+numchars.6+space.false+version.1.5.1`

ID	Model	Params	EL → EN	EL → / EN
1	Bilingual baselines	35.7M	38.4	17.1
5	Re-training + EL	–	40.4	19.4
6	Re-training + {EL,UK,SV,ID}	–	40.2	19.4
7	Only embed	2.13M	38.9	18.8
8	(**7**) + random embed init	2.13M	38.8	18.7⋆
9	(**7**) + enc-norm + enc-biases	2.17M	39.5	17.7
10	(**7**) + enc-adapters-first (dim=64)	2.19M	39.3	0.6
11	(**7**) + enc-adapters-all (d=64)	2.53M	40.3	0.6
12	(**7**) + enc-adapters-all (d=512)	5.28M	41.0	0.6
13	(**7**) + enc-adapters-{1,2,3} (d=1024)	5.28M	41.0	0.6
14	(**7**) + enc-first-layer	5.28M	40.1	0.7
15	(**7**) + all-enc-layers	21.0M	40.4	0.6
16	(**12**) + EL multi-aligned	5.28M	40.5	**19.5**
17	(**12**) + EL multi-aligned (BT)	5.28M	40.2	19.0
18	(**12**) + 1k lines per lang	5.28M	40.9	18.2
19	(**12**) + 1k lines per lang (BT)	5.28M	**41.2**	17.8
20	(**14**) + 1k lines per lang (BT)	5.28M	40.3	18.9
21	(**12**) + 100 lines per lang (BT)	5.28M	40.9	17.4
22	(**7**) + {EL,UK,SV,ID}	8.30M	39.3	18.9
23	(**14**) + {EL,UK,SV,ID}	11.5M	39.9	2.9

Table 2: TED test BLEU scores of incremental training with Greek on the source side. "EL → / EN" corresponds to an average BLEU from Greek into all 19 non-English languages. "Params" gives the number of new parameters introduced by each approach. The initial model (**4**) has 80.2M parameters in total. (⋆) obtained by using the "translate into X" lang code embeddings from the initial model. The table is divided in 4 parts: baselines trained with multi-aligned data; Greek-English incremental training; incremental training with multi-aligned data (i.e., line pairs between Greek and all 20 languages); and multilingual English-centric incremental training (i.e., on 4 new source languages at once).

sults on Greek, Ukrainian, Indonesian and Swedish are given in Appendix (Tables 10, 11, 12, and 13).

Training the source embeddings only (**7**) outperforms the bilingual baselines (**1**) and comes close to the costly re-training baselines (**5**, **6**). In particular, it nearly matches the performance of the latter in the zero-shot EL → / EN directions, even though the baselines have training data for those directions. Initializing the known tokens in the new vocabulary with their old embeddings does not improve final performance (**7** vs **8**). But using language code embeddings from the initial model is necessary to be able to translate into non-English languages. Figure 4 shows that such initialization improves final performance under low-resource settings. Figure 7 in Appendix also shows that it speeds up training.

Training additional components in the encoder, like adapter modules (**11**, **12**, **13**) or the first encoder layer (**14**) helps improve EL → EN performance and outperform all baselines, though it is less useful when the new language is from a known family (see Ukrainian and Swedish scores in Appendix Tables 11 and 13). However, this results in abysmal zero-shot performance (EL → / EN). As they only encounter the "to English" language code during training, those models quickly forget how to interpret the other lang codes. This catastrophic forgetting is illustrated by Figure 3, where we see a plunge in EL → FR performance after just a few epochs of training. Only tuning the encoder layer norm parameters and biases (**9**) gives slightly higher EL → EN performance without suffering from catastrophic forgetting in the other languages. Note that language code forgetting is less pronounced when the initial model is English-centric (see Table 18 in Appendix). In this setting, adapter modules do not hurt zero-shot translation.

The third quarter of Table 2 shows how multi-aligned Greek data can be used to achieve excellent performance in both EL → EN and EL → / EN directions. The best tradeoff between EL → EN and EL → / EN performance is achieved by incrementally training with the entire Greek dataset of 2.41M line pairs. However, such data might not

ID	Model	Params	EN → EL	/ EN → EL
1	Bilingual baselines	35.7M	32.2	18.3
5	Re-training + EL	–	32.5	**21.1**
6	Re-training + {EL,UK,SV,ID}	–	32.1	**21.1**
24	Only embed	2.13M	25.7	16.7
25	(**24**) + non-tied	4.25M	27.1	17.8
26	(**24**) + dec-adapters-all (dim=64)	2.53M	29.8	19.3
27	(**24**) + adapters-all (d=64)	2.93M	32.7	19.6
28	(**24**) + enc-adapters-last (d=1024)	3.18M	32.0	20.0
29	(**26**) + enc-adapters-last (d=1024)	3.58M	33.5	20.6
30	(**24**) + dec-last-layer	6.33M	32.6	20.5
31	(**30**) + enc-adapters-last (d=1024)	7.38M	34.0	20.8
32	(**24**) + adapters-all (d=430)	7.43M	34.0	18.2
33	(**24**) + dec-adapters-all (d=690) + enc-adapters-last (d=1024)	7.43M	33.8	20.8
34	(**30**) + adapters-all (d=90)	7.36M	**34.2**	19.8
35	(**30**) + enc-adapters-all (d=170)	7.38M	34.1	19.0
36	(**31**) + EL multi-aligned	7.35M	32.9	**21.1**
37	(**31**) + {EL,UK,SV,ID}	13.5M	33.0	20.4

Table 3: TED test BLEU scores of incremental training with Greek on the target side. "/ EN → EL" corresponds to an average BLEU from the 19 non-English languages to Greek. "Params" gives the number of new parameters introduced by each approach.

always be accessible for the new language. Close performance can be reached by training with the same amounts of synthetic data instead (**17**). And more interestingly, only a tiny amount of real (**18**) or back-translated data (**19**, **20**, **21**) in the other 19 languages is needed to obtain good zero-shot results without any loss in EL → EN performance.

New target language. Table 3 shows test BLEU scores when incrementally adding Greek on the target side. Additional results on Greek, Ukrainian, Indonesian and Swedish are provided in Appendix (Tables 14, 15, 16, and 17).

With new target languages, only adapting the embedding matrix (tied with vocabulary projection) is not enough and strongly underperforms the baselines (**24** vs **1**, **5** and **6**). Training decoder-side adapter modules (**26**) gets us closer to baseline performance; and tuning the last decoder layer (**30**) bridges the gap with the baselines. However, the most effective strategy is to train some components in both the encoder and decoder (**27**, **29**, **31**, **32**, **33**, **34**, **35**). We observed that it was important for the model to have a way to modify the output of the encoder before it is read by frozen decoder components. Interestingly, only having a large adapter module after the last encoder layer (**28**) is enough to match baseline performance. Adding small adapters after each decoder layer (**29**) further improves BLEU and brings the best parameter count / performance tradeoff.

At the same parameter budget, training adapter modules after every encoder layer (**32**, **34**, **35**) gives worse / EN → EL performance than an adapter at the last encoder layer combined with decoder-side parameters (**31**, **33**), which is likely caused by the encoder overfitting to English.

In this setting, there is no clear advantage to incremental training with multi-aligned Greek data (**36**), as this hurts EN → EL performance, without any notable improvement for / EN → EL. Finally, multilingual incremental training (with 4 new target languages at once) is entirely possible (**37**) and gives competitive results to the baselines.

Table 25 in Appendix analyzes the usefulness of learning a new language code, by comparing with three other strategies: incremental training without any language code; with the "to English" language code; or with the language code of a similar language. Interestingly, the more new parameters are learned (esp. encoder-side), the less useful it is to learn a new language code. Moreover, adapting to Swedish by using a fixed English language code gives reasonable performance as the two languages are from the same family. And the proxy "to Russian" language code gives the same results as learning a new language code when adapting to

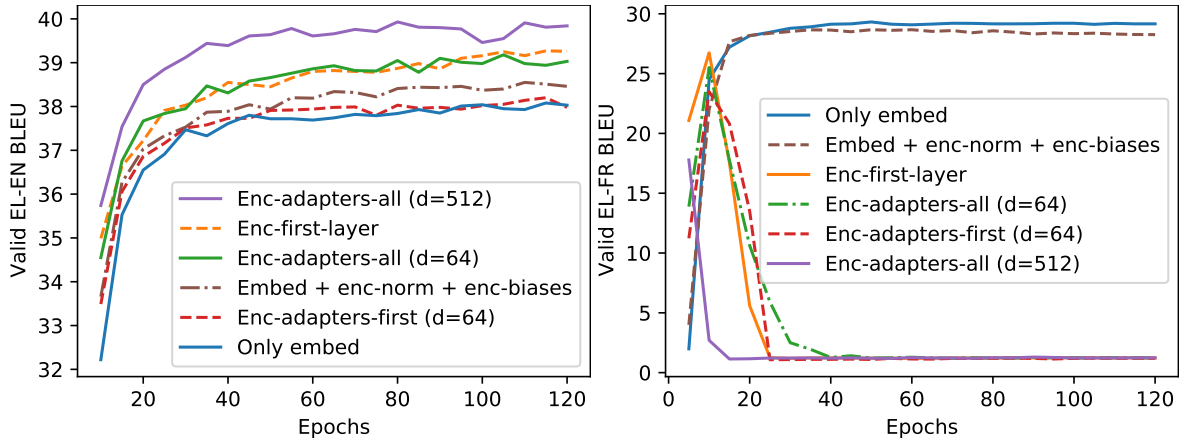


Figure 3: TED validation BLEU on EL-EN and EL-FR while incrementally training with EL-EN data only (**7**, **9**, **10**, **11**, **12**, **14**).

	Source model		Target model	BLEU
1	Bilingual			14.9
1	Bilingual (pivot through English)			18.5
6	Re-training + {EL,UK,SV,ID}			**22.0**
7	Only embed	**30**	Dec-last-layer	21.1
		31	Dec-last-layer + enc-adapters-last (d=1024)	21.0
14	Enc-first-layer	**30**	Dec-last-layer	20.7
		31	Dec-last-layer + enc-adapters-last (d=1024)	21.3
		31	Pivot through English*	21.6
20	Enc-first-layer + 1k (BT)	**30**	Dec-last-layer	21.2
		31	Dec-last-layer + enc-adapters-last (d=1024)	21.3
19	Enc-adapters-all (d=512) + 1k (BT)	**30**	Dec-last-layer	20.4
		31	Dec-last-layer + enc-adapters-last (d=1024)	20.7

Table 4: TED test BLEU scores on {EL,UK,SV,ID}→{EL,UK,SV,ID} (average over 12 directions) by combining source-language and target-language incrementally-trained parameters. ($\star$) instead of combining model parameters, translate to English with (**14**), then to the target language with (**31**).

Ukrainian because both languages are very similar.

New source and target languages Table 4 combines incrementally-trained parameters at inference time to translate between two new languages. Interestingly, combining target-language parameters with source-language parameters that had very poor zero-shot performance (**14**) gives excellent results. We hypothesize that the language code forgetting issue is less pronounced here because solely activating some language-specific parameters will make the model translate into that language.

Despite showing the best EL → EN performance, source-language encoder adapters (**19**) tend to perform more poorly when combined with target-language parameters. While better performance is obtained by pivot translation through English with two incrementally trained models (**14** and **31**), combining the parameters of these two models gives close results at a faster inference speed.

Discussion Figure 4 shows final BLEU scores of our techniques when training with smaller amounts of data. We observe that incremental training is more data-efficient than bilingual models and can achieve decent performance even with tiny amounts of training data, making it a good fit for adaptation to low-resource languages.

Figure 7 in Appendix illustrates the training speed of our approach compared to our implementation of Garcia et al. (2021). In addition to maintaining the exact same performance on the previous languages and needing only English-centric data, our approach reaches higher performance in much fewer updates than the alternatives. Note that retraining might be an efficient solution if one wants to add several languages at once and on both sides.

Finally, Tables 18 and 19 in Appendix show that

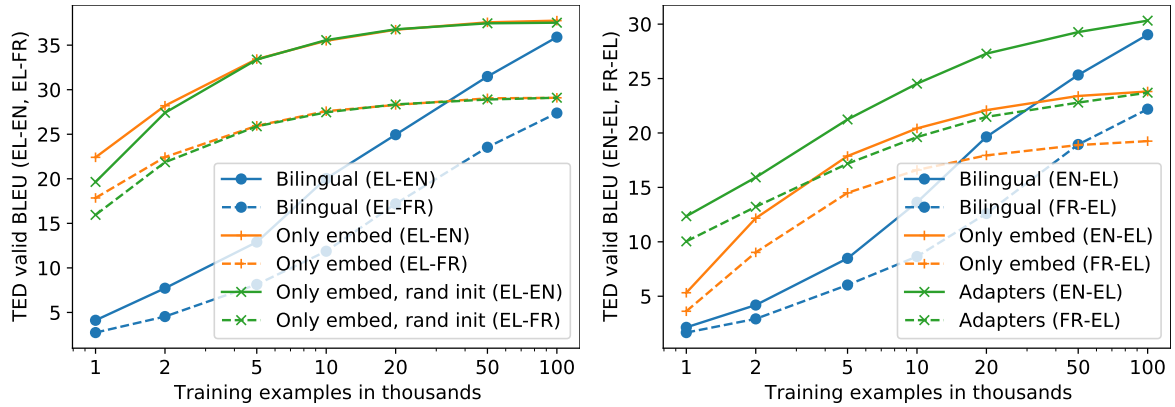


Figure 4: TED validation BLEU from Greek (left) and to Greek (right) by training corpus size, with incremental training (**7**, **8**, **24**, **27**) versus bilingual baselines (**1**).

ID	Model	→EN	←EN	/ EN
38	M2M-124	32.4	31.9	25.7
39	Big 6-6 EN-centric	38.8	36.4	18.5
40	Big 12-2 EN-centric	**39.6**	**37.1**	21.1
41	(**40**) + pivot (EN)	–	–	**27.6**
42	(**40**) + multi-para.	39.0	36.2	**27.6**
43	(**40**) + {AR,RU,ZH}	**39.6**	36.6	21.0
44	(**42**) + {AR,RU,ZH}	39.0	35.8	27.5

Table 5: FLORES devtest spBLEU scores of the ParaCrawl/UNPC baselines. Average to English (19 directions), from English (19 directions) and between non-English languages (342 directions). (**39**, **40**, **41**, **42**) are the same models as in (Bérard et al., 2021).

our incremental training approach can be applied to English-centric initial models with similar results.

5 ParaCrawl Experiments

In this section, we test our approach in a more realistic, large-scale setting: a 20-language Transformer Big initial model trained on ParaCrawl (Bañón et al., 2020). The incremental training experiments are done in three languages: Arabic (AR), Russian (RU), and Chinese (ZH). Arabic and Chinese are both from unseen language families and use unseen scripts. Russian is close to a known language (Bulgarian) and uses the same script. For training on those languages, we use UNPC (Ziemski et al., 2016).

5.1 Data and hyper-parameters

We download ParaCrawl v7.1 in the 19 highest-resource languages paired with English.[6] Then, like Freitag and Firat (2020), we build a multi-parallel corpus by aligning all pairs of languages through their English side (effectively removing any English duplicate). See Appendix Table 21 for training data statistics. We create a shared BPE vocab with 64k merge operations and inline casing, by sampling from this data with $T = 5$ and include all characters whose frequency is higher than 100.

Our initial model is the Transformer Big 12-2 (i.e., with a deep encoder and shallow decoder) multi-parallel model of Bérard et al. (2021). Like in the previous section, it was trained in two stages: English-centric training (with $T = 5$) for 1M steps; then multi-parallel fine-tuning (with $T = 2$) for 200k more steps. More hyper-parameters are given in Appendix (Table 24).

Incremental training is done for 120k steps with English-centric data from UNPC v1.0 (see Table 22 in Appendix for statistics), which we clean by removing any line pairs where either side is detected as being in the wrong language by `langid.py` (Lui and Baldwin, 2012). We use monolingual BPE models of size 8k. The Chinese data is tokenized with Jieba[7] before learning the BPE model.

The English-centric bilingual baselines are Transformer Big 6-6 models trained on UNPC for 120k steps with joint BPE vocabularies of size 16k. We do two "re-training" baselines, by fine-tuning either the English-centric or multi-parallel model on their initial ParaCrawl data plus UNPC data in all three new languages. We sample UNPC line pairs in each of the new directions with probability 0.05. The remaining 0.7 probability mass is distributed among the initial ParaCrawl directions with $T = 5$. The new BPE vocabulary is trained

[6] {fr, de, es, it, pt, nl, nb, cs, pl, sv, da, el, fi, hr, hu, bg, ro, sk, lt}

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

ID	Model	Params	AR	RU	ZH	AR	RU	ZH
			→ EN			→ / EN		
38	M2M-124 (Goyal et al., 2021)	–	25.5	27.5	20.9	19.6	24.0	18.7
45	Bilingual baselines (pivot through English)	193M	**32.1**	29.9	**23.7**	**23.2**	23.9	19.6
43	English-centric (**40**) + {AR,RU,ZH}	–	30.0	**31.7**	22.6	14.6	16.9	11.8
44	Multi-parallel (**42**) + {AR,RU,ZH}	–	31.0	31.6	23.1	19.2	23.5	15.9
50	(**44**) + pivot through English	–	–	–	–	22.2	24.5	**19.1**
46	Only embed	8.6M	24.2	30.8	20.5	16.5	23.8	15.6
47	(**46**) + enc-adapters-all (dim=512)	21.3M	31.7	31.3	23.5	1.0	1.5	1.5
48	(**46**) + enc-first-layer	21.2M	30.3	31.2	23.2	1.0	2.6	1.5
49	(**48**) + 20k lines per lang (BT)	21.2M	30.1	31.4	22.8	20.8	**24.6**	17.8

ID	Model	Params	EN →			/ EN →		
			AR	RU	ZH	AR	RU	ZH
38	M2M-124 (Goyal et al., 2021)	–	17.9	27.1	19.3	13.8	**23.0**	16.6
45	Bilingual baselines (pivot through English)	193M	**29.1**	27.5	**22.9**	**21.2**	22.5	**18.2**
43	English-centric (**40**) + {AR,RU,ZH}	–	26.5	26.8	18.4	15.7	19.3	11.3
44	Multi-parallel (**42**) + {AR,RU,ZH}	–	28.3	27.3	20.6	16.6	21.6	13.1
50	(**44**) + pivot through English	–	–	–	–	20.2	21.8	16.3
51	Only embed	8.6M	11.6	19.7	14.0	9.3	15.8	11.5
52	(**51**) + enc-adapt-last + dec-adapt-all (1024)	14.9M	27.0	26.2	20.9	18.8	20.8	16.8
53	(**51**) + dec-last-layer	25.4M	26.5	26.9	21.5	19.1	20.9	17.4
54	(**53**) + enc-adapters-last (dim=1024)	27.5M	28.2	28.0	22.5	20.0	21.8	18.0
55	(**54**) without lang ID filtering	27.5M	18.0	19.0	13.6	5.2	10.3	5.0
56	(**51**) + all-dec-layers	42.2M	28.6	**28.1**	22.4	20.1	21.9	18.1

Table 6: FLORES devtest spBLEU scores of the ParaCrawl/UNPC incrementally-trained models. The top half of each table corresponds to the baselines (SOTA, bilingual or re-training). "Params" gives the number of new parameters introduced by each approach. The incremental training is always done on one language only (i.e., one row can correspond to 3 different models). Note that the parameter counts given in this table are for Arabic (8.63M embedding parameters). Russian and Chinese embeddings have respectively 8.51M and 13.60M parameters.

with the monolingual ParaCrawl/UNPC data in all 23 languages (with $T = 5$). The new embeddings for the known tokens are initialized with their old values and the other embeddings at random.[8] Note that contrary to the TED Talks experiments, we do not have multi-aligned data for the new languages.

5.2 Evaluation settings

For validation, we use our own split from TED2020 (Reimers and Gurevych, 2020): 3000 random line pairs for each translation direction. We report chrF scores[9] computed on these valid sets in Appendix.

As test sets, we use FLORES devtest (Goyal et al., 2021). We report scores computed with their new "spBLEU" metric,[10] which runs BLEU on top of a standardized multilingual BPE tokenization.

[8] 78% of the new tokens were in the initial vocabulary, and 84% of the old tokens are in the new vocabulary.

[9] `chrF2+n.6+s.false+v.1.5.1`

[10] `BLEU+c.mixed+#.1+s.exp+tok.spm+v.1.5.1` (`https://github.com/ngoyal2707/sacrebleu`)

5.3 Results and analysis

Baselines. Table 5 compares our initial model (**42**) with other baselines. Our multi-parallel model beats the M2M-124 model of Goyal et al. (2021) in all three settings. This is not so surprising, as their model only has 615M parameters for 124 languages, compared to 255M parameters for our 20-language model. Last, we can observe that our "re-training" baselines (**43** and **44**) perform almost as well as the initial 20-language models (**40**, **42**).

New source or target language. Training only source embeddings (**46**) is a good strategy for Russian, but underperforms the baselines in the more linguistically distant Arabic and Chinese. Learning more parameters (+8% per source language) can match baseline performance in all 3 languages (**47** and **48**), but gives poor zero-shot performance. Adding small amounts of "back-translated" data (**49**) achieves close non-English performance

	Source model		Target model	spBLEU
38	M2M-124 (Goyal et al., 2021)			15.0
45	Bilingual baselines (pivot through English)			**18.3**
44	Multi-parallel (**42**) + {AR,RU,ZH}			13.0
50	(**44**) + pivot through English			16.8
46	Only embed	**53**	Dec-last-layer	13.5
		54	Dec-last-layer + enc-adapters-last (d=1024)	13.9
47	Enc-adapters-all (dim=512)	**53**	Dec-last-layer	13.0
		54	Dec-last-layer + enc-adapters-last (d=1024)	13.8
		54	Pivot through English	17.7
48	Enc-first-layer	**53**	Dec-last-layer	10.2
		54	Dec-last-layer + enc-adapters-last (d=1024)	10.6
49	Enc-first-layer + 20k (BT)	**53**	Dec-last-layer	12.4
		54	Dec-last-layer + enc-adapters-last (d=1024)	12.7

Table 7: FLORES devtest spBLEU scores of the ParaCrawl/UNPC models on {AR,RU,ZH}→{AR,RU,ZH} (average over 6 directions) by combining source-language and target-language incrementally-trained parameters.

to the pivot translation baselines without hurting English-centric scores. For new target languages, the best strategy is to train the last decoder layer with an adapter module at the last encoder layer (**54**), which matches the re-training baselines in all 3 languages and gets close performance to the parameter-inefficient bilingual baselines. Interestingly, target-side incremental training is very sensitive to training data noise. In a first iteration of our experiments, we trained with unfiltered UNPC data and observed catastrophic performance (**55**). Simple language ID filtering solved this issue.

New source and target languages. Table 7 combines source-language with target-language incrementally-trained parameters to translate between two new languages. The results are not as good as in our TED Talks experiments. The best combination in this setting (**46** with **54**) performs considerably worse than pivot translation through English with the baselines. However, it outperforms the "re-training" baseline (**44**), which has only seen English-centric data for the new languages. And pivot translation with two incrementally-trained models (**47** with **54**) gives excellent results, close to the bilingual baselines.

6 Conclusion

We propose a new technique for incrementally training multilingual NMT models on a new source or target language. It consists in creating a new monolingual BPE vocabulary for that language, substituting the shared embedding matrix with language-specific embeddings, and training those while freezing the other model parameters. At inference, translating in any of the initial languages is done by using the initial shared embeddings, and translating in the new language is done by using the newly trained embeddings. This approach does not change performance on the initial languages as the initial parameters are kept aside and not modified. For new source languages, it can achieve close performance to the more costly and less flexible bilingual and re-training baselines. For new target languages, this technique can be combined with language-specific parameters (fine-tuned Transformer layers or adapter modules) to match baseline performance at a small parameter cost. We validate this technique on two sets of experiments: small-scale on TED Talks and large-scale on ParaCrawl; and show that it is compatible with two architectures: Transformer Base 6-6 and Big 12-2. We also show that incremental training on data aligned with English is enough to learn to translate between the new language and any of the initial languages. Translation between a new source and a new target language is also possible by combining their respective parameters at inference. Finally, we provide supplementary material to facilitate reproducibility.[11]

[11] https://europe.naverlabs.com/research/natural-language-processing/efficient-multilingual-machine-translation

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver Labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy.

Alexandre Bérard, Dain Lee, Stéphane Clinchant, Kweonwoo Jung, and Vassilina Nikoulina. 2021. Efficient inference for multilingual neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic.

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 944–948, Online.

Carlos Escolano, Marta R Costa-jussà, José AR Fonollosa, and Mikel Artetxe. 2020. Training multilingual machine translation by alternately freezing language-specific encoders-decoders. *arXiv preprint arXiv:2006.01594*.

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.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

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.

Surafel Melaku Lakew, Alina Karakanta, Matteo Negri, Marcello Federico, and Marco Turchi. 2019. Adapting multilingual neural machine translation to unseen languages. In *16th International Workshop on Spoken Language Translation*.

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.

Sungwon Lyu, Bokyung Son, Kichang Yang, and Jaekyoung Bae. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 5905–5918, Online.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic.

Jerin Philip, Alexandre Bérard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online.

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, New Orleans, Louisiana.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online.

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.

Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing subnetworks to analyze domain adaptation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 124–132, Brussels, Belgium.

Ahmet Üstün, Alexandre Bérard, Laurent Besacier, and Matthias Gallé. 2021. Multilingual unsupervised neural machine translation with denoising adapters. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic.

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*.

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'16)*, pages 3530–3534, Portorož, Slovenia.

A Appendix

Language	Code	Family	Script	X-EN lines	X-* lines
English	en	Germanic	Latin	3.56M	3.56M
Arabic	ar	Semitic	Arabic	214.1k	3.43M
Hebrew	he	Semitic	Hebrew	211.8k	3.40M
Russian	ru	Slavic	Cyrillic	208.5k	3.38M
Korean	ko	Koreanic	Hangul	205.6k	3.35M
Italian	it	Romance	Latin	204.5k	3.35M
Japanese	ja	Japonic	Chinese + Kana	204.1k	3.31M
Mandarin Chinese	zh_cn	Sinitic	Chinese	199.9k	3.30M
Spanish	es	Romance	Latin	196.0k	3.23M
French	fr	Romance	Latin	192.3k	3.19M
Brazilian Portuguese	pt_br	Romance	Latin	184.8k	3.11M
Dutch	nl	Germanic	Latin	183.8k	3.05M
Turkish	tr	Turkic	Latin	182.5k	3.02M
Romanian	ro	Romance	Latin	180.5k	3.06M
Polish	pl	Slavic	Latin	176.2k	3.00M
Bulgarian	bg	Slavic	Cyrillic	174.4k	2.95M
Vietnamese	vi	Vietic	Latin	172.0k	2.81M
German	de	Germanic	Latin	167.9k	2.90M
Persian	fa	Iranian	Arabic	151.0k	2.41M
Hungarian	hu	Uralic	Latin	147.2k	2.47M
Greek	el	Hellenic	Greek	134.3k	2.41M
Ukrainian	uk	Slavic	Cyrillic	108.5k	1.81M
Indonesian	id	Malayic	Latin	87.4k	1.61M
Swedish	sv	Germanic	Latin	56.6k	978.0k
Total	all	–	–	7.11M	62.27M

Table 8: Size of the **Top 20 TED Talks** corpus. English has 253.3k unique lines.

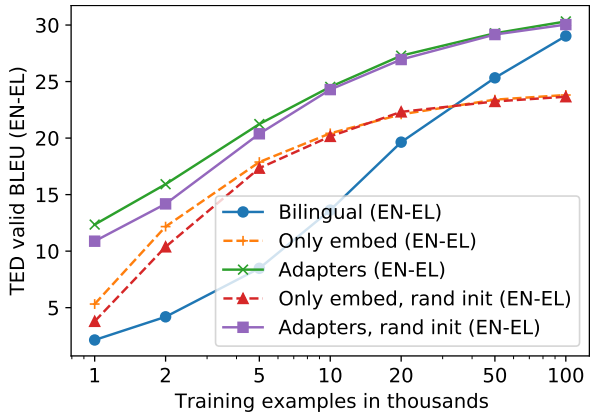


Figure 5: TED validation EN-EL BLEU by training corpus size, with incremental training (**24**, **27**) with or without known embedding initialization, versus bilingual baselines (**1**).

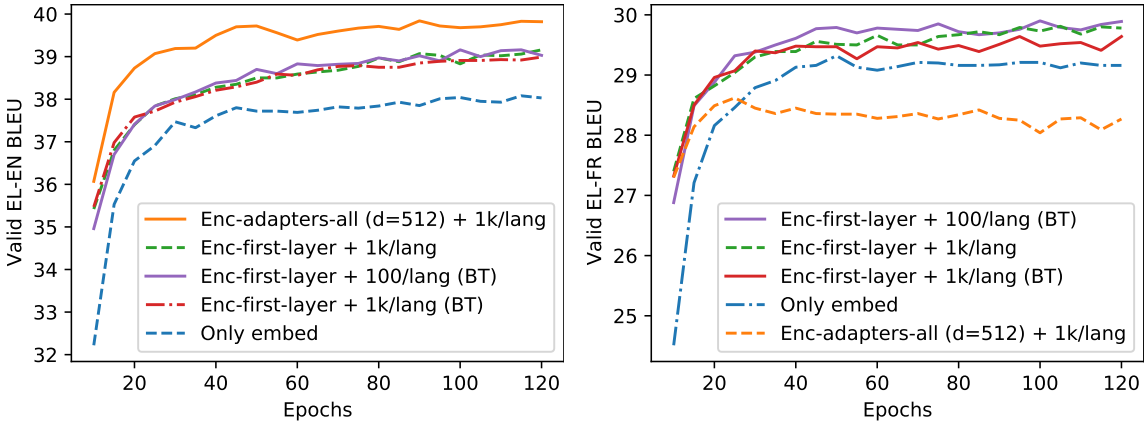


Figure 6: TED validation BLEU on EL-EN (left) and EL-FR (right) while training with EL-EN data, plus some amount of data (real or back-translated) in the 19 other languages.

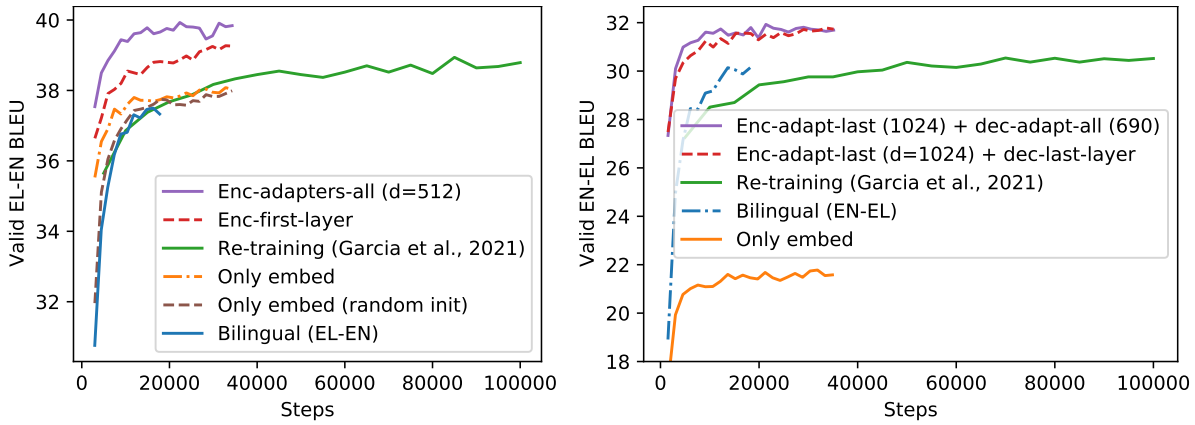


Figure 7: TED validation BLEU on EL-EN (left) and EN-EL (right) while training. Comparison of different incremental training approaches with the Garcia et al. (2021) baseline.

ID	Model	Valid chrF			Test chrF		
		→EN	←EN	/ EN	→EN	←EN	/ EN
1	Bilingual (pivot)	.542	.484	.385	.542	.484	.385
3	English-centric	.530	.487	.371	.529	.486	.370
4	(**3**) + multi-parallel training	.541	.482	.395	.540	.482	.395
57	(**3**) + EL	.528	**.488**	.372	.527	**.488**	.371
58	(**3**) + {EL, UK, ID, SV}	.526	.485	.370	.526	.485	.370
5	(**4**) + EL	**.545**	.481	**.398**	**.545**	.482	**.398**
6	(**4**) + {EL, UK, ID, SV}	**.545**	.480	**.398**	.544	.481	.397
59	(**4**) + EL (Garcia et al., 2021)	.539	.480	.394	.539	.479	.394
60	(**59**) @100k steps	.537	.479	.393	.538	.478	.392

Table 9: TED valid and test chrF scores of the baseline models. (**59**) corresponds to the best checkpoint according to validation loss (after 3 epochs, or 320k updates) and (**60**) is after just 100k updates.

ID	Model	Valid chrF		Test chrF	
		EL → EN	EL → / EN	EL → EN	EL → / EN
1	Bilingual baselines	.577	.399	.583	.400
5	Re-training + EL	.591	.426	.596	.425
6	Re-training + {EL,UK,SV,ID}	.590	.425	.594	.424
59	Re-training + EL (Garcia et al., 2021)	.587	.424	.594	.424
60	(**59**) @100k	.582	.421	.587	.420
7	Only embed	.577	.417	.581	.417
8	(**7**) + random embed init	.577	.417⋆	.580	.417⋆
9	(**7**) + enc-norm + enc-biases	.582	.407	.587	.407
10	(**7**) + enc-adapters-first (d=64)	.578	.102	.584	.100
11	(**7**) + enc-adapters-all (d=64)	.587	.102	.593	.100
12	(**7**) + enc-adapters-all (d=512)	.593	.102	.602	.100
13	(**7**) + enc-adapters-{1,2,3} (d=1024)	**.595**	.103	**.603**	.101
14	(**7**) + enc-first-layer	.590	.105	.595	.102
15	(**7**) + enc-all-layers	.590	.102	.598	.100
16	(**12**) + EL multi-aligned	.592	**.427**	.599	**.428**
18	(**12**) + 1k lines per lang	.594	.412	.601	.413
19	(**12**) + 1k lines per lang (BT)	**.595**	.411	**.603**	.411
20	(**14**) + 1k lines per lang (BT)	.589	.422	.596	.422
21	(**12**) + 100 lines per lang (BT)	**.595**	.405	.601	.406
22	(**7**) + {EL,UK,SV,ID}	.582	.419	.585	.419
61	(**12**) + {EL,UK,SV,ID}	.593	.103	.597	.100
23	(**14**) + {EL,UK,SV,ID}	.587	.158	.592	.154

Table 10: TED valid and test chrF scores of incremental training with Greek on the source side. (⋆) obtained by using the “translate into X" lang code embeddings from the initial model.

ID	Model	Valid chrF		Test chrF	
		UK → EN	UK → / EN	UK → EN	UK → / EN
1	Bilingual baselines	.484	–	.494	–
6	Re-training + {EL,UK,SV,ID}	.522	**.402**	**.534**	**.402**
7	Only embed	.516	.397	.525	.395
9	(**7**) + enc-norm + enc-biases	.518	.386	.526	.385
10	(**7**) + enc-adapters-first (d=64)	.519	.100	.525	.099
11	(**7**) + enc-adapters-all (d=64)	.520	.100	.529	.100
12	(**7**) + enc-adapters-all (d=512)	.520	.100	.529	.099
13	(**7**) + enc-adapters-{1,2,3} (d=1024)	.523	.100	.530	.100
14	(**7**) + enc-first-layer	.521	.136	.529	.134
15	(**7**) + enc-all-layers	.517	.100	.525	.099
16	(**12**) + UK multi-aligned	.522	**.402**	.530	.401
18	(**12**) + 1k lines per lang	.523	.389	.531	.387
19	(**12**) + 1k lines per lang (BT)	.520	.384	.528	.382
20	(**14**) + 1k lines per lang (BT)	.522	.397	.528	.396
21	(**12**) + 100 lines per lang (BT)	.520	.382	.527	.382
22	(**7**) + {EL,UK,SV,ID}	.518	.398	.526	.396
61	(**12**) + {EL,UK,SV,ID}	**.524**	.101	.532	.100
23	(**14**) + {EL,UK,SV,ID}	.522	.132	.527	.130

Table 11: TED valid and test chrF scores of incremental training with Ukrainian on the source side.

ID	Model	Valid chrF		Test chrF	
		ID → EN	ID → / EN	ID → EN	ID → / EN
1	Bilingual baselines	.516	–	.533	–
6	Re-training + {EL,UK,SV,ID}	.541	**.397**	.554	.404
7	Only embed	.533	.390	.548	.399
8	(**7**) + random embed init	.529	.385⋆	.547	.395⋆
9	(**7**) + enc-norm + enc-biases	.537	.379	.551	.389
10	(**7**) + enc-adapters-first (d=64)	.535	.100	.551	.101
11	(**7**) + enc-adapters-all (d=64)	.540	.101	.557	.102
12	(**7**) + enc-adapters-all (d=512)	.541	.101	.558	.102
13	(**7**) + enc-adapters-{1,2,3} (d=1024)	**.545**	.101	.558	.102
14	(**7**) + enc-first-layer	.540	.101	.554	.102
15	(**7**) + enc-all-layers	.535	.100	.552	.101
16	(**12**) + ID multi-aligned	.540	**.397**	.556	**.405**
18	(**12**) + 1k lines per lang	.541	.383	.556	.390
19	(**12**) + 1k lines per lang (BT)	.542	.382	.558	.388
20	(**14**) + 1k lines per lang (BT)	.536	.389	.553	.399
21	(**12**) + 100 lines per lang (BT)	.542	.380	.557	.389
22	(**7**) + {EL,UK,SV,ID}	.530	.388	.547	.397
61	(**12**) + {EL,UK,SV,ID}	.543	.101	**.560**	.102
23	(**14**) + {EL,UK,SV,ID}	.539	.125	.553	.126

Table 12: TED valid and test chrF scores of incremental training with Indonesian on the source side. (⋆) obtained by using the “translate into X" lang code embeddings from the initial model.

ID	Model	Valid chrF		Test chrF	
		SV → EN	SV → / EN	SV → EN	SV → / EN
1	Bilingual baselines	.577	–	.579	–
6	Re-training + {EL,UK,SV,ID}	.611	**.424**	.615	**.426**
7	Only embed	.601	.417	.607	.420
8	(**7**) + random embed init	.596	.414⋆	.605	.417⋆
9	(**7**) + enc-norm + enc-biases	.604	.413	.613	.415
10	(**7**) + enc-adapters-first (d=64)	.606	.100	.613	.102
11	(**7**) + enc-adapters-all (d=64)	.610	.100	.617	.102
12	(**7**) + enc-adapters-all (d=512)	.608	.100	.613	.102
13	(**7**) + enc-adapters-{1,2,3} (d=1024)	.612	.100	.619	.102
14	(**7**) + enc-first-layer	.608	.102	.617	.104
15	(**7**) + enc-all-layers	.601	.099	.605	.101
16	(**12**) + SV multi-aligned	.609	.421	.615	.424
18	(**12**) + 1k lines per lang	.606	.408	.613	.410
19	(**12**) + 1k lines per lang (BT)	.607	.406	.611	.409
20	(**14**) + 1k lines per lang (BT)	.605	.417	.613	.420
21	(**12**) + 100 lines per lang (BT)	.605	.402	.611	.405
22	(**7**) + {EL,UK,SV,ID}	.601	.417	.605	.418
61	(**12**) + {EL,UK,SV,ID}	**.616**	.100	**.620**	.102
23	(**14**) + {EL,UK,SV,ID}	.607	.164	.615	.172

Table 13: TED valid and test chrF scores of incremental training with Swedish on the source side. (⋆) obtained by using the “translate into X" lang code embeddings from the initial model.

ID	Model	Valid chrF		Test chrF	
		EN → EL	/ EN → EL	EN → EL	/ EN → EL
1	Bilingual baselines	.551	.421	.570	.432
5	Re-training + EL	.551	**.452**	.569	.460
6	Re-training + {EL,UK,SV,ID}	.550	.451	.568	.460
59	Re-training + EL (Garcia et al., 2021)	.553	.449	.570	.458
60	(**59**) @100k	.553	.450	.572	.459
24	Only embed	.504	.415	.518	.423
25	(**24**) + non-tied	.517	.423	.530	.432
26	(**24**) + dec-adapters-all (d=64)	.533	.435	.551	.444
27	(**24**) + adapters-all (d=64)	.556	.440	.574	.449
28	(**24**) + enc-adapters-last (d=1024)	.555	.445	.571	.455
29	(**26**) + enc-adapters-last (d=1024)	.560	.450	.580	.459
30	(**24**) + dec-last-layer	.556	.449	.576	.458
31	(**30**) + enc-adapters-last (d=1024)	.566	**.452**	.585	.461
32	(**24**) + adapters-all (d=430)	.566	.426	.585	.436
33	(**24**) + dec-ad-all (690) + enc-ad-last (1024)	.566	.451	.584	.461
34	(**30**) + adapters-all (d=90)	.565	.442	.585	.453
35	(**30**) + enc-adapters-all (d=170)	**.567**	.435	**.586**	.444
62	(**24**) + dec-all-layers	.562	.448	.577	.457
36	(**31**) + EL multi-aligned	.554	.451	.573	**.462**
37	(**31**) + {EL,UK,SV,ID}	.558	.449	.578	.458

Table 14: TED valid and test chrF scores of incremental training with Greek on the target side.

ID	Model	Valid chrF		Test chrF	
		EN → UK	/ EN → UK	EN → UK	/ EN → UK
1	Bilingual baselines	.441	–	.440	–
6	Re-training + {EL,UK,SV,ID}	.460	.401	.459	.394
24	Only embed	.446	.386	.445	.380
25	(**24**) + non-tied	.452	.390	.449	.384
26	(**24**) + dec-adapters-all (d=64)	.457	.394	.453	.387
27	(**24**) + adapters-all (d=64)	.466	.397	.465	.391
28	(**24**) + enc-adapters-last (d=1024)	.464	.400	.463	.393
29	(**26**) + enc-adapters-last (d=1024)	.469	.402	.465	.394
30	(**24**) + dec-last-layer	.468	.402	.464	.394
31	(**30**) + enc-adapters-last (d=1024)	**.471**	**.403**	.467	**.395**
32	(**24**) + adapters-all (d=430)	.470	.386	.468	.380
33	(**24**) + dec-ad-all (690) + enc-ad-last (1024)	.468	.402	.466	.394
34	(**30**) + adapters-all (d=90)	.469	.400	.467	.393
35	(**30**) + enc-adapters-all (d=170)	.470	.393	**.469**	.387
62	(**24**) + dec-all-layers	.468	.400	.463	.391
36	(**31**) + UK multi-aligned	.460	.400	.458	.393
37	(**31**) + {EL,UK,SV,ID}	.467	.402	.465	**.395**

Table 15: TED valid and test chrF scores of incremental training with Ukrainian on the target side.

ID	Model	Valid chrF		Test chrF	
		EN → ID	/ EN → ID	EN → ID	/ EN → ID
1	Bilingual baselines	.568	–	.579	–
6	Re-training + {EL,UK,SV,ID}	.579	**.498**	.591	.504
24	Only embed	.562	.483	.575	.491
25	(**24**) + non-tied	.569	.487	.582	.496
26	(**24**) + dec-adapters-all (d=64)	.579	.492	.591	.501
27	(**24**) + adapters-all (d=64)	.585	.489	.599	.498
28	(**24**) + enc-adapters-last (d=1024)	.586	.493	.600	.501
29	(**26**) + enc-adapters-last (d=1024)	**.589**	.495	.602	.504
30	(**24**) + dec-last-layer	.588	.496	.598	.503
31	(**30**) + enc-adapters-last (d=1024)	.587	.496	.601	.503
32	(**24**) + adapters-all (d=430)	**.589**	.480	.599	.489
33	(**24**) + dec-ad-all (690) + enc-ad-last (1024)	.586	.494	.599	.502
34	(**30**) + adapters-all (d=90)	.588	.493	**.603**	.502
35	(**30**) + enc-adapters-all (d=170)	**.589**	.489	.602	.496
62	(**24**) + dec-all-layers	.588	.497	.600	**.506**
36	(**31**) + ID multi-aligned	.579	.496	.589	.502
37	(**31**) + {EL,UK,SV,ID}	.584	.493	.597	.501

Table 16: TED valid and test chrF scores of incremental training with Indonesian on the target side.

ID	Model	Valid chrF		Test chrF	
		EN → SV	/ EN → SV	EN → SV	/ EN → SV
1	Bilingual baselines	.557	–	.557	–
6	Re-training + {EL,UK,SV,ID}	.568	.453	.567	.455
24	Only embed	.547	.436	.548	.438
25	(**24**) + non-tied	.552	.441	.556	.443
26	(**24**) + dec-adapters-all (d=64)	.569	.451	.572	.453
27	(**24**) + adapters-all (d=64)	.584	.444	.589	.448
28	(**24**) + enc-adapters-last (d=1024)	.583	.451	.586	.455
29	(**26**) + enc-adapters-last (d=1024)	.587	.452	.590	.457
30	(**24**) + dec-last-layer	.580	.452	.584	.458
31	(**30**) + enc-adapters-last (d=1024)	**.589**	.454	.590	.458
32	(**24**) + adapters-all (d=430)	.587	.437	.597	.443
33	(**24**) + dec-ad-all (690) + enc-ad-last (1024)	.584	.452	.587	.457
34	(**30**) + adapters-all (d=90)	.587	.449	.590	.453
35	(**30**) + enc-adapters-all (d=170)	.588	.443	**.591**	.449
62	(**24**) + dec-all-layers	.583	.452	.586	.457
36	(**31**) + SV multi-aligned	.570	.454	.569	.456
37	(**31**) + {EL,UK,SV,ID}	**.589**	**.455**	**.591**	**.459**

Table 17: TED valid and test chrF scores of incremental training with Swedish on the target side.

ID	Model	Valid chrF		Test chrF	
		EL → EN	EL → / EN	EL → EN	EL → / EN
1	Bilingual baselines (pivot)	.577	.407	.583	.407
57	Re-training + EL (pivot)	.575	**.408**	.578	**.408**
58	Re-training + {EL,UK,SV,ID} (pivot)	.572	.405	.573	.404
7	Only embed	.571	.402	.579	.403
9	(**7**) + enc-norm + enc-biases	.574	.403	.581	.403
10	(**7**) + enc-adapters-first (d=64)	.575	.404	.582	.404
11	(**7**) + enc-adapters-all (d=64)	.581	.401	.587	.401
12	(**7**) + enc-adapters-all (d=512)	**.584**	.386	.589	.387
13	(**7**) + enc-adapters-{1,2,3} (d=1024)	**.584**	.403	**.591**	.404
14	(**7**) + enc-first-layer	.581	.114	.588	.111
15	(**7**) + enc-all-layers	.579	.102	.590	.100
18	(**12**) + 1k lines per lang	.581	.394	.589	.393
19	(**12**) + 1k lines per lang (BT)	.583	.396	.588	.396
20	(**14**) + 1k lines per lang (BT)	.579	.404	.587	.404
21	(**12**) + 100 lines per lang (BT)	.583	.391	.590	.391

Table 18: TED valid and test chrF scores of incremental training with Greek on the source side when the initial model is English-centric (**3**).

ID	Model	Valid chrF		Test chrF	
		EN → EL	/ EN → EL	EN → EL	/ EN → EL
1	Bilingual baselines (pivot)	.551	.435	.570	.444
57	Re-training + EL (pivot)	.557	.430	.577	.440
58	Re-training + {EL,UK,SV,ID} (pivot)	.556	.428	.573	.438
24	Only embed	.518	.401	.534	.409
25	(**24**) + non-tied	.529	.407	.545	.416
26	(**24**) + dec-adapters-all (d=64)	.545	.415	.561	.424
27	(**24**) + adapters-all (d=64)	.564	.414	.584	.423
28	(**24**) + enc-adapters-last (d=1024)	.562	.420	.581	.429
29	(**26**) + enc-adapters-last (d=1024)	.568	.424	.586	.432
30	(**24**) + dec-last-layer	.565	.427	.585	.435
31	(**30**) + enc-adapters-last (d=1024)	.571	.426	.589	.435
32	(**24**) + adapters-all (d=430)	**.573**	.408	.591	.416
33	(**24**) + dec-ad-all (690) + enc-ad-last (1024)	.568	.425	.589	.434
34	(**30**) + adapters-all (d=90)	**.573**	.420	.592	.428
35	(**30**) + enc-adapters-all (d=170)	.572	.417	**.594**	.426
62	(**24**) + dec-all-layers	.566	.426	.583	.434
36	(**31**) + EL multi-aligned	.553	**.441**	.570	**.450**

Table 19: TED valid and test chrF scores of incremental training with Greek on the target side when the initial model is English-centric (**3**).

Source model		Target model		Valid	Test
1	Bilingual			.384	.388
1	Bilingual (pivot through English)			.428	.437
6	Re-training + {EL,UK,SV,ID}			**.461**	**.470**
7	Only embed	**30**	Dec-last-layer	.456	.465
		31	Dec-last-layer + enc-ad-last (d=1024)	.457	.466
14	Enc-first-layer	**30**	Dec-last-layer	.453	.463
		31	Dec-last-layer + enc-ad-last (d=1024)	**.461**	**.470**
		31	Pivot through English*	.460	.469
20	Enc-first-layer + 1k (BT)	**30**	Dec-last-layer	.459	.468
		31	Dec-last-layer + enc-ad-last (d=1024)	**.461**	**.470**
19	Enc-adapters-all (d=512) + 1k (BT)	**30**	Dec-last-layer	.451	.459
		31	Dec-last-layer + enc-ad-last (d=1024)	.453	.462

Table 20: TED valid and test chrF scores on {EL,UK,SV,ID}→{EL,UK,SV,ID} (average over 12 directions) by combining source-language and target-language incrementally-trained parameters. ($\star$) instead of combining model parameters, translate with (**14**) to English, then to the target language with (**31**).

Language	Code	Family	X-EN lines	X-* lines
English	en	Germanic	450.30M	450.30M
French	fr	Romance	95.43M	215.63M
German	de	Romance	76.49M	192.67M
Spanish	es	Romance	72.97M	191.71M
Italian	it	Romance	38.05M	136.11M
Portuguese	pt	Romance	29.18M	117.68M
Dutch	nl	Germanic	27.36M	104.35M
Norwegian	nb	Germanic	15.38M	65.37M
Czech	cs	Slavic	12.92M	65.55M
Polish	pl	Slavic	12.88M	69.27M
Swedish	sv	Germanic	10.97M	60.16M
Danish	da	Germanic	9.79M	61.28M
Greek*	el	Hellenic	8.92M	48.29M
Finnish	fi	Uralic	6.83M	47.62M
Croatian	hr	Slavic	6.34M	30.47M
Hungarian	hu	Uralic	6.29M	42.53M
Bulgarian*	bg	Slavic	6.10M	36.84M
Romanian	ro	Romance	5.79M	40.52M
Slovak	sk	Slavic	4.56M	36.39M
Lithuanian	lt	Baltic	4.03M	30.21M
Total	all	–	900.60M	2.043B

Table 21: Size of the **Top 20 ParaCrawl** corpus. English has 271.85M unique lines. ($\star$) all languages use the Latin script, except for Greek and Bulgarian (Cyrillic).

Language	Code	Family	X-EN lines
Russian	ru	Slavic	25.17M
Arabic	ar	Semitic	20.04M
Mandarin Chinese	zh	Sinitic	17.45M

Table 22: Size of the UNPC corpus.

Parameter name	Parameter value
share_all_embeddings	True / False[5,6]
share_decoder_input_output_embed	True
arch	transformer
lr_scheduler	inverse_sqrt
optimizer	adam
adam_betas	0.9,0.999
fp16	True
clip_norm	0.0
lr	0.0005 / 0.0001[4]
warmup_updates	4000
warmup_init_lr	1e-07
criterion	label_smoothed_cross_entropy
label_smoothing	0.1
dropout	0.3 / 0.1[2,3,4]
max_tokens	4000
max_epoch	120[1,5] / 10[2,4] / 20[3,6]
save-interval	1 / 5[5]
validate-interval	1 / 5[5]
update_freq*	4
reset_*	True
lang_temperature†	5

Table 23: fairseq v0.10.2 hyper-parameters of the **TED Talks models**. ($\star$) we normalize this value by the number of GPUs to have a constant batch size. For instance, models trained on 4 GPUs use `update_freq=1`. (1) English-centric training stage of the initial model; (2) multi-parallel training stage; (3) our re-training approach; (4) our implementation of Garcia et al. (2021); (5) English-centric incremental training; (6) multi-aligned incremental training. The bilingual baselines use the `transformer_iwslt_de_en` architecture and are trained for 25k steps with validation every 500 steps and patience 3. (†) we implement an on-the-fly data loading pipeline that builds heterogeneous batches by sampling language pair k with probability: $p_k = D_k^{1/T} / (\sum D_i^{1/T})$ where T is the temperature and D_k is the total number of line pairs for that language pair (Aharoni et al., 2019).

Parameter name	Parameter value
max_source_positions	256
max_target_positions	256
share_all_embeddings	True / False[5]
share_decoder_input_output_embed	True
arch	transformer_vaswani_wmt_en_de_big
encoder_layers	12 / 6[4]
decoder_layers	2 / 6[4]
lr_scheduler	inverse_sqrt
optimizer	adam
adam_betas	0.9,0.98
fp16	True
clip_norm	1.0
lr	0.0005
warmup_updates	4000
warmup_init_lr	1e-07
criterion	label_smoothed_cross_entropy
label_smoothing	0.1
dropout	0.1
max_tokens	8000
max_update	1000000[1] / 200000[2] / 360000[3] / 120000[4,5]
save_interval_updates	20000 / 10000[4] / 5000[5]
validate_interval_updates	20000 / 10000[4] / 5000[5]
update_freq*	32
reset_*	True
lang_temperature†	5 / 2[2,3]

Table 24: fairseq hyper-parameters of the **ParaCrawl/UNPC models**. ($\star$) (†) see Table 23. (1) English-centric training stage of the initial model; (2) multi-parallel training stage; (3) our re-training approach; (4) bilingual baselines; (5) incremental training.

ID	Model	Lang code	EN →				/ EN →			
			EL	UK	ID	SV	EL	UK	ID	SV
24	Embed only	None	-17	-18	-19	-10	-15	-18	-18	-8
		EN	-18	-19	-17	-5	-14	-16	-17	-7
		Proxy	-6	-2	-2	-13	-3	+0	-2	-6
30	(**24**) + dec-last-layer	None	-2	-4	-3	+3	-4	-6	-3	-1
		EN	+0	-2	-1	+5	-3	-5	-3	+0
		Proxy	+3	+1	-1	+4	+0	+1	-1	+2
31	(**30**) + enc-adapters-last (d=1024)	None	-1	-4	+2	-6	-2	-5	-3	-2
32	Enc-adapters-all (dim=430)	None	+2	-1	-1	+1	-8	+1	+1	-3
33	(**24**) + enc-adapters-last (d=1024) + dec-adapters-all (d=690)	None	-1	-1	1	-1	-2	-4	-1	-4

Table 25: TED valid chrF delta ($\times 1000$) of target-side incremental learning techniques with fixed language codes, compared to models with learned language codes. "None" corresponds to training and decoding without any language code. "EN" trains and decodes with the pre-trained (and frozen) "to English" language code. "Proxy" uses the closest pre-trained language code (RU for UK, BG for EL, DE for SV and VI for ID). This is an oracle, obtained by computing the Euclidean distance between trained language codes in (**6**).

ID	Model	→EN	←EN	/ EN
39	Big 6-6 English-centric	.582	.571	.400
40	Big 12-2 English-centric	**.587**	**.577**	.435
42	(**40**) + multi-parallel training	.583	.573	.486
41	(**40**) + pivot through English	–	–	**.488**
43	(**40**) + {AR,RU,ZH}	.585	.574	.433
44	(**42**) + {AR,RU,ZH}	.580	.569	.486

Table 26: TED2020-valid chrF scores of the ParaCrawl/UNPC baselines.

ID	Model	AR	RU	ZH	AR	RU	ZH
		→ EN			→ / EN		
45	Bilingual baselines (pivot through English)	.499	.460	.430	**.429**	.423	.381
43	English-centric (**40**) + {AR,RU,ZH}	.488	**.480**	.430	.372	.385	.337
44	Multi-parallel (**42**) + {AR,RU,ZH}	.494	.479	.430	.395	.418	.345
50	(**44**) + pivot through English	–	–	–	.424	**.433**	**.382**
46	Only embed	.447	.469	.416	.378	.425	.365
63	(**46**) without lang ID filtering	.447	.469	.416	.378	.425	.365
47	(**46**) + enc-adapters-all (d=512)	**.502**	.478	**.434**	.154	.157	.152
48	(**46**) + enc-first-layer	.491	.474	.428	.154	.168	.152
64	(**48**) without lang ID filtering	.488	.474	.427	.154	.158	.151
49	(**48**) + 20k lines per lang (BT)	.492	.474	.427	.417	**.433**	.376
65	(**49**) without lang ID filtering	.433	.457	.385	.148	.158	.148

ID	Model	EN →			/ EN →		
		AR	RU	ZH	AR	RU	ZH
45	Bilingual baselines (pivot through English)	.423	.437	.187	.358	.400	.156
43	English-centric (**40**) + {AR,RU,ZH}	.412	.439	.179	.295	.384	.115
44	Multi-parallel (**42**) + {AR,RU,ZH}	.423	.443	.182	.300	.402	.126
50	(**44**) + pivot through English	–	–	–	.356	.402	.153
51	Only embed	.314	.398	.158	.277	.364	.134
66	(**51**) without lang ID filtering	.282	.395	.130	.224	.301	.084
52	(**51**) + enc-adapters-last + dec-adapters-all (d=1024)	.417	.437	.187	.348	.397	.153
53	(**51**) + dec-last-layer	.412	.441	.187	.348	.401	.154
54	(**53**) + enc-adapters-last (d=1024)	**.426**	**.446**	.192	.356	**.404**	.156
55	(**54**) without lang ID filtering	.312	.363	.107	.072	.185	.037
56	(**51**) + dec-all-layers	**.426**	**.446**	**.193**	**.357**	**.404**	**.159**

Table 27: TED2020-valid chrF scores of the ParaCrawl/UNPC incrementally-trained models.

Source model		Target model		chrF
45	Bilingual baselines (pivot through English)			**.274**
44	Multi-parallel (**42**) + {AR,RU,ZH}			.237
50	(**44**) + pivot through English			.271
46	Only embed	**53**	Dec-last-layer	.248
		54	Dec-last-layer + enc-adapters-last (d=1024)	.252
47	Enc-adapters-all (d=512)	**53**	Dec-last-layer	.242
		54	Dec-last-layer + enc-adapters-last (d=1024)	.251
		54	Pivot through English*	**.274**
48	Enc-first-layer	**53**	Dec-last-layer	.223
		54	Dec-last-layer + enc-adapters-last (d=1024)	.234
49	Enc-first-layer + 20k (BT)	**53**	Dec-last-layer	.243
		54	Dec-last-layer + enc-adapters-last (d=1024)	.251

Table 28: TED2020-valid chrF scores of the ParaCrawl/UNPC models on {AR,RU,ZH}→{AR,RU,ZH} (average over 6 directions) by combining source-language and target-language incrementally-trained parameters. ($\star$) instead of combining model parameters, translate to English with (**48**), then to the target language with (**54**).

English-centric	Denotes a parallel corpus that only has alignments with English (i.e., 38 language pairs in our settings). Also denotes many-to-many models that are trained with such data. In this setting, translation between non-English language pairs is called "zero-shot."
Multi-parallel	Denotes a parallel corpus that has alignments between all possible language pairs (380 in our case), and by extension, models that are trained with such data (AKA "complete multilingual NMT", Freitag and Firat, 2020).
Enc-adapters-X ($d = N$)	Train adapter modules of bottleneck dimension N after the encoder layer X.
Enc-adapters-all ($d = N$)	Train adapter modules of bottleneck dimension N after all encoder layers.
Enc-layer-X	Fine-tune the Xth Transformer encoder layer.
Enc-norm + enc-biases	Fine-tune the layer norm parameters and all the biases in the Transformer encoder.
+ random embed init	Initialize the new language-specific embeddings at random, instead of initializing the embeddings of the known tokens with their previous values.
+ non-tied	Train separate target embeddings and output projection matrix (by default they are tied, i.e., they correspond to the same parameter).
+ EL multi-aligned	Train with multi-aligned EL data: not just paired with English, but with all of the 20 languages (2.41M lines pairs instead of 134k).
+ EL multi-aligned (BT)	Like above, but the non-EN data is obtained by translating the English side of the EL-EN corpus to the other 19 languages with (**3**). For better comparison with the above method, we back-translate the same number of lines per language as in the multi-aligned EL corpus.
+ N lines per lang	Append to the EL-EN training data N line pairs for each of the 19 non-English languages.
+ N lines per lang (BT)	Like above, but the N line pairs per language are obtained by translating the English side of the EL-EN corpus with (**3**).
/ EN	19 non-English languages in the initial model. As a column header, it means an average score over all 342 non-English translation directions.
$\rightarrow$ EN	Average score over all 19 X $\rightarrow$ EN translation directions.
$\leftarrow$ EN	Average score over all 19 EN $\rightarrow$ Y translation directions.
Re-training + $\{L_1, L_2 \ldots\}$	Fine-tune the initial model with an updated BPE vocabulary and embedding matrix that include the new languages (L_1, L_2, etc.), and on the multi-aligned data of all the 20 initial languages plus the new ones.
(K) + $\{L_1, L_2 \ldots\}$	Use the incremental training technique K, but on several languages at once (L_1, L_2, etc.) This means that a single shared BPE is trained for all these languages (whose size is multiplied by the number of languages) and the newly-trained parameters are shared between them.

Table 29: Summary of the notations used in this paper.

DELA Corpus - A Document-Level Corpus Annotated with Context-Related Issues

Sheila Castilho
ADAPT Centre - School of Computing
Dublin City University
sheila.castilho@adaptcentre.ie

João L. Cavalheiro Camargo
Western Paraná State University
joao.camargo@unioeste.br

Miguel Menezes
University of Lisbon
lmenezes@campus.ul.pt

Andy Way
ADAPT Centre - School of Computing
Dublin City University
andy.way@adaptcentre.ie

Abstract

Recently, the Machine Translation (MT) community has become more interested in document-level evaluation especially in light of reactions to claims of "human parity", since examining the quality at the level of the document rather than at the sentence level allows for the assessment of suprasentential context, providing a more reliable evaluation. This paper presents a document-level corpus annotated in English with context-aware issues that arise when translating from English into Brazilian Portuguese, namely ellipsis, gender, lexical ambiguity, number, reference, and terminology, with six different domains. The corpus can be used as a challenge test set for evaluation and as a training/testing corpus for MT as well as for deep linguistic analysis of context issues. To the best of our knowledge, this is the first corpus of its kind.

1 Introduction

Machine translation (MT) is now widely used in a variety of fields, mainly due to advancements in neural models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). As a result of these recent advances, scientists have been increasingly attempting to include discourse into neural machine translation (NMT) systems (Wang, 2019; Lopes et al., 2020). Thus, researchers started to consider a more suitable evaluation for these document-level systems as the standard MT automatic evaluation metrics have been shown to underestimate the quality of NMT systems (Shterionov et al., 2018) and the appropriateness of these metrics for document-level systems has been challenged (Smith, 2017) since they are not sensitive to their improvements (Voita et al., 2019).

Accordingly, document-level human evaluation of MT has attracted the community's attention since it allows for a more thorough examination of the output quality with context. While a few works have taken into account document-level human evaluation (Läubli et al., 2018; Toral et al., 2018; Barrault et al., 2019; Castilho, 2020, 2021), one common practice for document-level evaluation is the usage of test suites with context-aware markers (Bawden et al., 2018; Guillou et al., 2018; Müller et al., 2018; Voita et al., 2019; Cai and Xiong, 2020). However, the concept of document-level evaluation, in terms of how much text must be shown, remains uncertain (Castilho et al., 2020). While most research on document-level MT evaluation works with contrastive pairs, very few works have tried to use full documents for human evaluation (Läubli et al., 2018; Castilho, 2020, 2021) and challenge test sets (Rysová et al., 2019; Vojtěchová et al., 2019). Methodologies for assessing MT at the document-level have been looked into (Barrault et al., 2019, 2020) as well as the types of issues that come with different methodologies (Castilho, 2020, 2021).

We present a document-level corpus annotated with context-aware issues when translating from English (EN) into Brazilian-Portuguese (PT-BR). In total, 60 documents from six different domains (literary, subtitles, news, reviews, medical, and europarl) were annotated with context-aware issues, namely gender, number, ellipsis, reference, lexical ambiguity, and terminology. The corpus can be used as a challenge test set for the evaluation and as a training/testing corpus for MT and quality estimation, as well as for deep linguistic analysis of context issues. Moreover, we believe that the annotation can be also used for close-related languages such as Spanish.

2 Related Work

Document-level MT evaluation has attracted interest in the field as it allows for the evaluation of

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 566–577
November 10–11, 2021.

suprasentential content, which in turn, provides more meaningful insights on the MT output. However, the definition of what constitutes a document-level MT evaluation is still unclear (Castilho et al., 2020).

Context plays an important role as it is widely used in translation and interpreting literature (Baker, 2006), although it lacks a precise definition for practical purposes, including in everyday work of a professional translator (Melby and Foster, 2010). For Melby and Foster (2010, p 3), context in translation could be studied "either for the purpose of analysing existing translations or for the purpose of improving the production of new translations". For the authors, context can be categorised into *non-text* (non-linguistic variables) and *text* (linguistic aspects), where the latter is divided into four aspects of context: relating to the source text: *co-text* (the version of the document itself) and *chron-text* (past and future versions); and relating to other text: *rel-text* (monolingual related texts) and *bi-tex* (bilingual related texts). In this work, we adopt Melby and Foster's view of context that is important to the analysis of translations, and focus (i) on the co-text, i.e. the boundaries within the document translated, and (ii) in the non-text, where the name of the authors, speakers, and products have an effect on the translation.

In a survey with native speakers, Castilho et al. (2020) tested the context span for the translation of 300 sentences in three different domains, namely reviews, subtitles, and literature. The results showed that over 33% of the sentences tested were found to require more context than the sentence itself to be translated or evaluated, and from those, 23% required more than two previous sentences to be properly evaluated. The authors found that ambiguity, terminology, and gender agreement were the most common issues to hinder translation. Moreover, differences in issues and context span were found between domains. Their recommendations include to show whole documents when possible, include information on text type, topic, product, hotel and movie names in case of reviews, and include visual context whenever possible (non-text). This shows that document-level evaluation enables the assessment of textual cohesion and coherence types of errors which are impossible at times to recognise at sentence level.

Regarding overall MT evaluation, a few attempts have been made to perform human evaluation with document-level set-ups. Läubli et al. (2018) compared sentence-level evaluation versus document-level evaluation with pairwise rankings of fluency and adequacy to evaluate the quality of MT against human translation (HT) with professional translators. Their results show that document-level raters clearly preferred HT over MT, especially in terms of fluency. The authors argue that document-level evaluation enables the identification of certain types of errors, such as ambiguous words, or errors related to textual cohesion and coherence.

The Conference for Machine Translation (WMT), which has been running since 2006 and only evaluated sentences, attempted document-level human evaluation for the news domain for the first time in 2019 (Barrault et al., 2019). Their direct assessment (DA) (Graham et al., 2016) required crowdworkers to assign a score (0-100) to each sentence. They asked raters to evaluate (i) whole texts, (ii) single consecutive segments in their original order, and (iii) single random phrases. In the following year, WMT20 changed the approach and expanded the context span to include full papers, requiring raters to evaluate specific segments while seeing the complete document, as well as to assess the content's translation (Barrault et al., 2020).

Castilho (2020, 2021) tested for the differences in inter-annotator agreement (IAA) between single sentence and document-level set-ups. In Castilho (2020), the author asked translators to evaluate the MT output of freely available online systems in terms of fluency, adequacy (Likert scale), ranking and error annotation in two different set-ups: (i) translators give one score per single isolated sentence, and (ii) translators give one score per document. The results showed that IAA scores for the document-level set-up reached negative levels, and the level of satisfaction of translators with that methodology was also very low. Nonetheless, it avoided cases of misevaluation that happen in isolated single sentences. Following on from that work, Castilho (2021) modifies the document-level set-up and re-runs the experiment with more translators, where she compares the IAA in evaluation of (i) random single sentences, (ii) evaluation of individual sentences where translators have access to the full source and MT output, and (iii) evaluation of full documents. Results showed that a methodology where translators assess individual sentences within the context of a document yields a good level

of IAA compared to the random single-sentence methodology, while a methodology where translators give one score per document shows a very low level of IAA. The author demonstrates that the methodology of assigning one score per sentence in context avoids misevaluation cases which are extremely common in the random sentences-based evaluation set-ups. Moreover, the author posits that the higher IAA agreement in the random single sentence set-up is because "raters tend to accept the translation when adequacy is ambiguous but the translation is correct, especially if it is fluent" (Castilho, 2021, p 42), and asserts that the single random sentence evaluation method should be avoided as the misevaluation issue is especially problematic when assessing the quality of NMT systems as they have an improved fluency level.

One current way of evaluating document-level issues is the use of test suites designed to better evaluate translation of the addressed discourse-level phenomena. Commonly, these test suites are contrastive, that is, each sample sentence in the test has both correct and wrong translations for a given phenomena (Bawden et al., 2018; Guillou et al., 2018; Müller et al., 2018; Voita et al., 2019; Cai and Xiong, 2020). The higher the accuracy of the model in rating correct translations over incorrect ones, the better the quality is deemed to be. Test suites with document-level boundaries are still scarce, e.g. Vojtěchová et al. (2019) present a test suite designed to evaluate coherence when testing MT models trained for the news domain on audit reports, and Rysová et al. (2019) designed a document-level test suite to assess three document-level discourse phenomena, namely information structure, discourse connectives, and alternative lexicalisation of connectives.

Given the above, the need to move toward document-level methodologies in MT is indisputable. Moreover, with the lack of resources for the topic of document-level MT, the document-level corpus annotated with context-aware issues presented here can be used as a challenge test set for evaluation and as a training/testing corpus for MT as well as for deep linguistic analysis of context issues.

3 Corpus Compilation

The corpus was collected from a variety of freely available sources. Following a pre-determined list of context issues found in Castilho et al. (2020) that hindered the translation of single sentences and sentence pairs, the annotators searched for challenging English texts for the MT systems when translating into PT-BR. In total, 60 full documents (57217 tokens) were collected from six different domains: literary, subtitles, news, reviews, medical, and legislation (europarl). Table 1 shows the statistics of the corpus.

Domains	#Docs	#Sent.	Av. Sent. Lgth
Subtitles	9	1074	18.69
Literary	4	756	9.76
News	15	634	17.17
Reviews	28	608	13.42
Medical	3	339	13.02
Legislation	1	272	23.70
TOTAL	60	3683	15.57

Table 1: Full corpus statistics, where *average sentence length* is calculated as words per sentence.

Each domain has their plain text and .xls versions of the documents segmented into sentences with sentence id and document boundary tags, and all documents contain the source (url or corpus) where the documents were retrieved from. What follows is a detailed description of each domain is provided.[1]

3.1 Subtitles

To compile the corpus for the subtitle domain, nine full TED Talks were selected from the Opus Corpus (Tiedemann, 2012) from a variety of different topics and speakers, where: doc1: education, doc2: climate change, doc3: astronomy, doc4: computers, doc5: creativity, doc6: science, doc7: technology, doc8: anthropology, and doc9: psychology. We chose these talks specifically in order to obtain a blend of different topics and speakers' genders.

	#Sent.	#Tokens	Av. Sent. Lgth
doc1	105	1671	15.91
doc2	98	1309	13.35
doc3	40	650	16.66
doc4	71	1213	17.08
doc5	176	3654	20.88
doc6	130	2485	19.26
doc7	77	1384	18.45
doc8	167	4213	25.53
doc9	210	3346	16.00
TOTAL	1074	19925	18.55

Table 2: Corpus statistics for each document in the subtitle domain.

[1] Although some of the documents were already segmented by sentence (i.e. Opus and WMT), the full corpus was manually checked for sentence segmentation.

3.2 Literary

	#Sent.	#Tokens	Av. Sent. Lgth
doc1	205	2921	14.24
doc2	122	2002	16.40
doc3	76	689	9.06
doc4	353	1767	5.00
TOTAL	756	7379	9.76

Table 3: Corpus statistics for each document in the Literary domain.

To compile the corpus for the literature domain, four documents[2] were selected:
doc1: one chapter from a fan-fiction story.[3]
doc2: one excerpt from "The Road to Oz" book.[4]
doc3: a short story generated with the PlotGenerator website.[5]
doc4: a short play generated with the PlotGenerator website.

Note that a blend of contemporary and classic excerpts, combining descriptive and fast moving styles, were gathered. Note too that the synthetic stories (doc3 and doc4) were generated as they allowed the researchers to add a good number of possible issues, including lexical ambiguities cases that can only be solved with a larger context than two consecutive sentences which is rather difficult to find in "natural" texts. Nonetheless, English native speakers then revised both stories for fluency and readability. Table 3 shows the statistics for each document in the literary domain.[6]

3.3 News

The news domain was compiled with 15 documents gathered from different sources. Table 4 shows the statistics of the corpus.[7]

Five documents were gathered from the WMT series (four documents from WMT19[8] and one from WMT20[9]), and their size varied from 13 to 32 sentences. Ten documents were gathered from several news websites,[10] and they varied from 23-35 sentences.

	#Sent.	#Tokens	Av. Sent. Lgth
docs 1-5	112	2293	20.47
docs 6-15	521	8578	16.46
TOTAL	633	10871	17.17

Table 4: Corpus statistics for each document in the news domain.

3.4 Reviews

The reviews domain was compiled with 28 documents gathered from reviews available on Amazon[11] and TripAdvisor[12] websites. Table 5 shows the statistics of the corpus.[13]

Reviews gathered from Amazon consist of users' reviews about a variety of products and movies, totalling 25 reviews, and vary from 6 to 84 sentences. The reviews were sought by searching products that could generate lexical ambiguities, such as "plant", "ship", etc. Reviews gathered from TripAdvisor consist of 3 reviews about places, and vary from 23-35 sentences.

	#Sent.	#Tokens	Av. Sent. Lgth
docs 1-25	520	6901	13.27
docs 26-28	88	1261	14.32
TOTAL	608	8162	13.42

Table 5: Corpus statistics for each document in the review domain. Documents 1-25 are product reviews gathered on the Amazon website, and documents 26-28 are location reviews gathered on the TripAdvisor website.

3.5 Medical

The medical domain corpus was compiled with three full documents, where two of them were collected from Autopsy reports available on the Medical Transcriptions website,[14] and one document was collected from the leaflets available on the

[2] Excerpts of two copyrighted books are in the process of being granted permission, and if so, they will be added to the corpus.

[3] "Harmonic Resonances" (based on the Carrie film) fan fiction (archiveofourown.org/works/26524723/chapters/64650841), last accessed 01 June 2021.

[4] Chapter 3 "Queer Village" (www.gutenberg.org/files/485/485-h/485-h.htm#chap03), last accessed 21 June 2021.

[5] https://www.plot-generator.org.uk/ last accessed 21 June 2021.

[6] Note that the Av. Sent. Lgth for the literature domain is skewed because of doc4, which – due to its play format where the names of each character is given in a single line before they speak – contains a great number of very short sentences. The Av. Sent. Lgth for literature when doc4 is left out is 13.9.

[7] Note that for the news domain, we grouped documents due to space constraints.

[8] http://www.statmt.org/wmt19/

[9] http://www.statmt.org/wmt20/

[10] mercurynews.com, zdnet.com, usmagazine.com, machinedesign.com, nytimes.com, thejournal.ie, thesun.ie, theconversation.com, goodhousekeeping.com, allthatsinteresting.com, last accessed 01 June 2021.

[11] amazon.com

[12] tripadvisor.com

[13] Note that for the review domain, we grouped documents due to space constraints.

[14] mtsamples.com

Royal College of Obstetricians and Gynaecologists (RCOG).[15]

	#Sent.	#Tokens	Av. Sent. Lgth
docs 1-2	243	2912	11.98
doc 3	96	1503	15.65
TOTAL	339	4415	13.02

Table 6: Corpus statistics for documents in the medical domain. Documents 1-2 were compiled from autopsy reports, while document 3 was compiled from medical leaflets

3.6 Legislation

For the legislation domain, we chose an excerpt of Europarl (Koehn, 2005)[16] taken from the Opus Corpus (Tiedemann, 2012).

	#Sent.	#Tokens	Av. Sent. Lgth
doc 1	272	6465	23.7

Table 7: Corpus statistics for documents in the legislation domain extracted from the Europarl corpus.

4 Methodology for Annotation

Following literature on document-level test suites (see Section 2), together with issues found when trying to define how much context span is needed to translate and evaluate MT (Castilho et al., 2020), we compiled a list of context-aware issues that are challenging for MT when translating from EN into PT-BR to be annotated:

1- Gender	2- Number
3- Ellipsis	4- Reference
5- Lexical Ambiguity	6- Terminology

Three annotators tagged those issues that might occur in a translation from EN into PT-BR when no context information is given. For example, in the following single sentence given to a translator to translate:

"And thanks for the case."

The translator will not be able to translate this sentence with absolute certainty because:
i) it is not possible to know the gender of the person who is saying 'thanks' as Portuguese differentiates between masculine and feminine genders.
ii) it is not possible to know what the word "case" is as this word has a few different meanings that would fit this sentence, i.e it could be some type of protective box (a case for my phone, a case for my glasses), a woman's bag, a pencil case, a folder, a suitcase, or a police case to be investigated, each one with a different translation in Portuguese. Consequently, the translation of "for" will have a different gender depending on the meaning of the word "case".

When evaluating the translation of the source sentence given by 3 different MT systems (Google Translate[17] (GG), Microsoft Bing[18] (MS) and DeepL[19] (DPL) the translator has to evaluate all three systems' outputs as correct:
GG: "E obrigado pelo caso." (masculine, police case)
MS: "E obrigado pelo estojo." (masculine, pencil case)
DPL: "E obrigado pela caixa." (masculine, box)

That is because without a wider context, it is impossible to know the correct translation or the sentence, which should be:

HT: *"E obrigada pela capa."* (feminine, phone case)

Therefore, the issues tagged in the corpus are issues that might arise in the translation of sentences when the full context is not given. Annotators used different MT systems to help check for issues that would go unnoticed when only looking at the source text.

Moreover, a few modifications to the source text were performed in order to add those issues and make the translation more challenging for MT, such as modifying the gender, substituting the name of a product for 'it', splitting a sentence into two, etc. These modifications are explained in the spreadsheet file for each line modified, so researchers can decide if they can use or not documents that had the source modified.

4.1 Annotation of Context-Related Issues

As previously mentioned, six context-related issues were tagged in the corpus when they could not be solved within the sentence they appeared. A detailed guideline was developed as the annotators gathered the corpus and discussed how the annotation would be better performed. Figure 1 shows the decision tree that guides the annotation of the context-related issues.

[15] Copyright permission was granted by both websites.
[16] http://www.statmt.org/europarl/
[17] https://translate.google.com/
[18] https://www.bing.com/translator
[19] https://www.deepl.com/en/translator

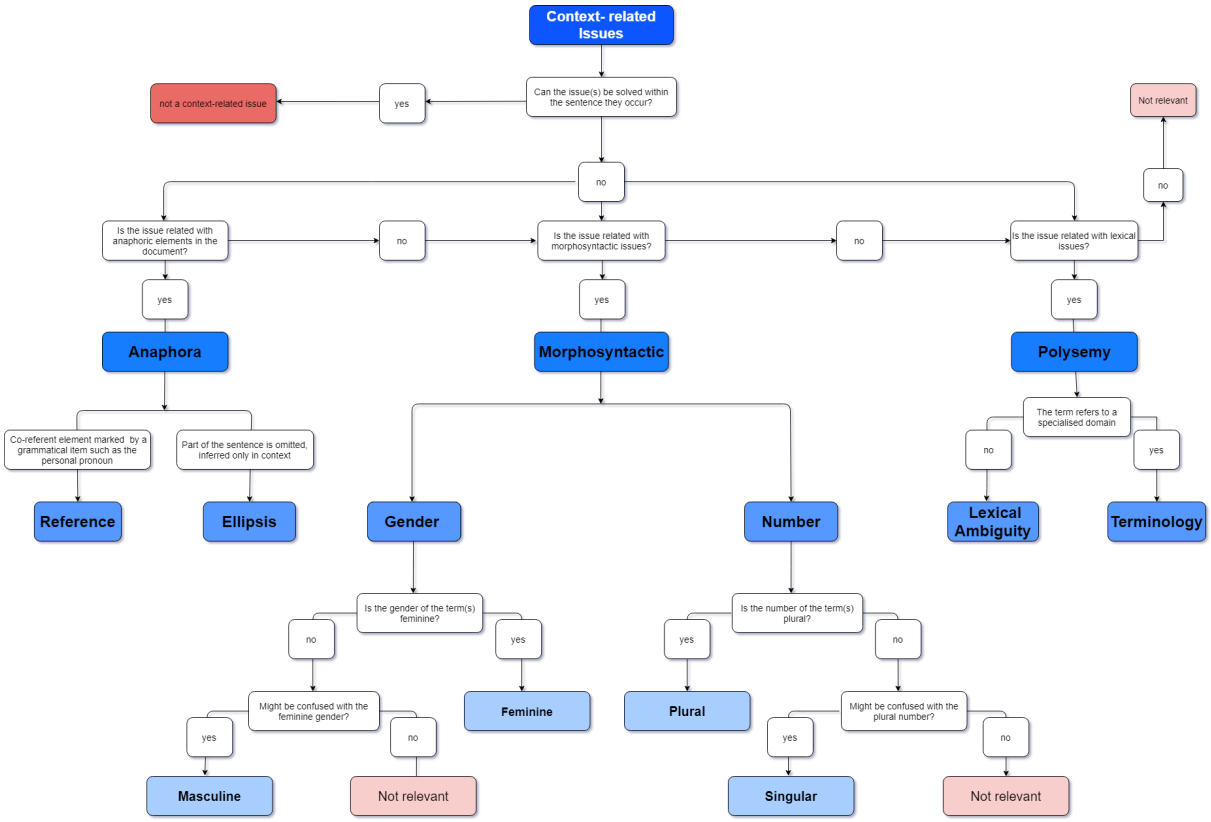


Figure 1: Decision tree used to guide the annotation of context-related issues.

4.1.1 Reference

Reference is associated with the notion that "anaphoric references and noun phrase organizers may serve as cohesive ties linking separate sentences into unified paragraphs [aiding] the reader's memory structure" (Fishman, 1978, p 159). Differently from ellipsis which is generally dependent on the previous clause, reference can reach back a long way in the text and extend over a long passage (Halliday and Matthiessen, 2013), thus being of significance for the present work.

In the annotation guide, we annotated the reference whenever we faced a disruption or ambiguity in the referential chain, e.g., we only annotated dependent referential units. Moreover, and similar to all annotated categories, the disagreement had to be expressed at the document level, e.g. the issue could not be solved only by looking at the sentence. In example **A)**, we annotate the second individual *it* as being a referential issue because there is not enough lexical material in the sentence to properly establish the referent, thus affecting translation correctness and final text readability.

A) *It is understandable though since* ***it*** *was shipped from China.*
reference –> it = the ship
it = o navio.

In example **B)**, we annotated *they* as being a referential unit issue, due to the fact that there is not enough lexical material in the sentence to determine its referent. Moreover, we also tagged this issue as a gender problem since there is no information in the source sentence that allows one to determine that the referential unit should be translated into PT-BR as a plural feminine pronoun.

B) *They actually hurt*
reference –> they = the choices
gender –> they = feminine
They actually hurt = Elas / As escolhas realmente machucam.

4.1.2 Ellipsis

Ellipsis is a form of anaphoric cohesion where there is an omission from a clause, and, so, the reader must "presuppose something by means of what is left out" (Halliday and Matthiessen, 2013, p 635). Ellipsis differs from reference as the relationship it entails is lexicogrammatical rather than semantic

(Halliday and Matthiessen, 2013).

In the annotation guide, we annotate ellipsis exclusively when the omission of information affects the translation of that specific single sentence which needs a broader context to be understood. For example, in **C)**, ellipsis is tagged because the omission with the explicit indication of "do" causes lexical ambiguity that cannot be solved within the sentence.[20] Therefore, the tags and the solution for the issues are both ellipsis and lexical ambiguity, with the translation of the ellipsis also containing an explanation of the lexical ambiguity caused by it.

C) *In my laughter, I bellied out a "YES, I do!!"*
ellipsis –> do = think
lexical ambiguity –> do = make (incorrect) *vs* think (correct)
Sim, eu faço! (Yes, I make, incorrect) *vs*
Sim, eu **acho!** (Yes, I "think", correct)

In example **D)**, ellipsis is tagged because the omission causes gender issues that cannot be solved within the sentence. Therefore, the tags and the solution for the issues are both ellipsis and gender, with the translation of the ellipsis also containing an explanation of the lexical ambiguity caused by the ellipsis:

D) *Several more are planned for the rest of the year, including The Angry Beavers Seasons 1 & 2, Danny Phantom Season 1, Aaahh!! Real Monsters Season 1, Catdog Season 1 Part 1 and The Wild Thornberrys Season 2 Part 1.*

ellipsis –> several more = releases
gender –> several more are planned = feminine
Several more are planned... = Vári**as** outr**as** estão planejad**as**..."

Sentence **E)** is an example of ellipsis with the auxiliary *do* that has not been tagged in the corpus because the omission is solved within the sentence:

E) *Also, not once did I **feel a blast of hot** air like I **do** when taking things out of the oven.*
do = feel a blast of hot air

[20]It is only with the previous sentence "*He came back in the house and said 'So you think this is funny?!' up the stairway at me and I LOST IT*" that we can solve "I do" as being "I think so".

4.1.3 Gender

As Portuguese is a language in which grammatical gender (feminine and masculine) plays a very significant role, the definition of gender used is from a grammatical standpoint, where word classes like adjectives, articles or pronouns are bound to respect and reflect a word's gender (Grosjean et al., 1994).

In the annotation guide, we annotated gender whenever facing a gender issue e.g., gender disagreement, unsolvable within the sentence itself and requiring broader context information.[21] For example, in example **F)**, gender (feminine) is tagged because the issue is not possible to be solved within the sentence. Since the default of the PT-BR is to have everything in the masculine, translations (both HT and MT) follow the same pattern. Therefore, we tag the word that needs to be in a different gender and the solution for its gender marker, with the translation containing an explanation:

F) *I'm surprised to see you back so early.*
gender –> surprised = feminine
surprised = surpresa

In example **G)**, we note that not only the pronoun "they" needs to be tagged with the feminine gender tag, but also the expression "staying at", as it is translated with an adjective in Portuguese:

G) *She waited for a few minutes longer, but nothing happened, no one followed her, so she made her way back to the motel they were staying at.*
gender –> they = feminine
gender –> staying at = feminine
they were staying at = elas estavam hospedadas

Gender was also tagged even when the most used translation for the the given term was a neutral one, because the adjective could still be translated with one of its synonyms. For instance, in example **H)**, the adjective "glad" has its most common translation as "feliz" which is used for both masculine and feminine gender. If a

[21]Note that gender was most exclusively tagged as *feminine* when a problem with the agreement was obvious. As the MT systems typically tend to translate the gender into the masculine form (when no specific gender markers are given in the single sentence) for PT-BR, the masculine gender was only tagged when there was an ambiguity in the sentence.

translator chooses to translate the text as "I'm glad = Estou feliz", no gender marking is needed. However, synonyms of that translation would need to be translated into feminine (satisfeita, grata, encantada, animada), and so, gender is tagged for that case:

H) *I'm so glad that it comes with the extender, so I have more levels to use to continue to get smaller.*
gender –> glad = feminine
reference –> it = waist cincher

4.1.4 Number

Number agreement is one of the basic coherence devices within a text, and it is "part of the code, used to signal that linguistic constituents carrying the same number are linked regardless of whether they appear together or apart in an utterance"(Bock et al., 1999, p 330), in the entirety of the text, and thus is significant for the present work.

In the annotation guide, we annotated number whenever we faced a number disagreement within the referential chain, e.g. (i) noun or pronoun, (ii) verb and noun/pronoun, (iii) adjective, caused by lack of enough contextual information within the sentence.[22] In example **I)**, the number category was applied to the word *you* because it is not possible to identify within this single sentence whether we are facing a pronoun in the plural or singular.

I) *I was praying for you.*
number –> you = plural
you –> vocês

Example **J)** depicts a mistranslated number agreement chain into PT-BR which originated from the absence of contextual evidence in the sentence that allowed us to determine whether *you* should be translated in the plural rather than in the singular. Furthermore, as a consequence of this initial mistranslation, the adjective *agreeable* was affected, being translated in the singular rather than the plural.

J) *You should be more agreeable.*
number –> you = plural
number –> agreeable = plural
number –> should be = plural
gender –> agreeable = feminine
You should be more agreeable. –> Você**s** dever**iam** ser mais simpátic**as/** agradáv**eis**.

4.1.5 Lexical Ambiguity

Lexical ambiguity refers to the fact that "a single word form can refer to more than one different concept" (Rodd, 2018, p 2). Lexical ambiguity can be divided into two categories: (i) one takes into account a word's morphological aspect (verbs, adjectives) referred to as syntactic ambiguity, e.g. the word "play" can be either the act of taking part in a sport or the conducting of a sporting match; and (ii) the second focuses on the fact that a word can assume different meanings according to context, e.g. the word "ball" as in *They danced till dawn at the ball* versus *This dog can be entertained all day with a ball* (Small et al., 2013, p 4), which is referred as semantic ambiguity.

In the annotation guide, we annotated lexical ambiguity, the more generic term, whenever we faced one of the two cases above ((i) and (ii)) and whenever they appeared to be detrimental to the translation and understandable only within the broader context, rather than at sentence level. In example **K)**, lexical ambiguity is tagged because the clause *I lost it*, without context, can be interpreted either as someone losing something or someone losing control:

K) *He came back in the house and said "So you think this is funny?!" up the stairway at me and I LOST IT.*

lexical ambiguity –> lose something *vs* to lose control
I lost it –> Eu o/a perdi *vs* Eu perdi o controle

In example **L)**, lexical ambiguity is tagged because the word *Period* is polysemic, meaning simultaneously menstruation, a portion of time, and a punctuation mark, and by the fact that there is not enough lexical information at a sentence level to disambiguate the complete meaning.

L) *Period.*
lexical ambiguity –> period = era/menstruation *vs* full stop
Período *vs* Ponto final

[22] Note that number was most exclusively tagged as *plural* for the pronoun "you" (and its referential chain (verb/adjectives)) when a problem with the agreement was obvious. As the MT systems typically tend to translate "you" in the singular (when no specific plural markers are given in the single sentence) for PT-BR, the pronoun was only tagged for singular when there was an ambiguity in the sentence.

4.1.6 Terminology

Terminology, according to Sager and Nkwenti-Azeh (1990) (as cited in (Kast-Aigner, 2018)), can have three different definitions: (i) the theory behind the relationship between a concept and a term; (ii) terminology curatorship activities, e.g. collection, description, processing and presentation of terms; and (iii) the vocabulary of a specific field. In the present work, we perceived terminology as (iii) i.e. the lexical expression of a domain-specific area.

In the annotation guide, we annotated terminology whenever we faced a wrongly domain-specific word translation caused by contextual poor sentences. In the following example **M)**, the category terminology was applied to the word 'farm' because its meaning shifts from "a piece of land used for crops and cattle raising", its more generalised conceptualisation, into a more domain-specific concept, "an area of land with a group of energy-producing windmills or wind turbines".

M) *The center will also conduct testing (power curve, mechanical loads, noise, and power quality) at its own experimental wind farm*
terminology –> generalised lexic (farm) *vs* domain-specific lexic (park)
wind farm –> parque eólico

4.2 Format

The annotation was performed for each sentence, which are tagged one per line, in the order they appear in the sentence, followed by their explanation/solution, along with modifications performed in the source (if any) and translations of some cases and notes. Sentences with no context-related issues are followed by two Xs for the issue and the solution. For Reference and Ellipsis, the term that contains the issue is stated along with an equals sign (=) and the explanation of what it refers to. For Gender and Number, the issue is tagged along with an equals sign (=) and the solution (feminine/masculine or singular/plural) is given. For Lexical Ambiguity and Terminology, the term (or terms) is stated along with an equals sign (=) and a contrasting solution is given, the wrong meaning(s) compared to (vs) the correct one. Table 8 illustrates how the annotation format is performed for each issue.

The corpus will be made freely available in two formats. One is a spreadsheet (.xls) containing the tagged corpus in all domains and full information. This .xls format will allow for filtering specific issues or sentences and enable users/researchers to see the rationale of the annotation. The corpus will be also available in plain text (.txt) format, containing the segment id, sentence, issue and explanation all in one line.[23] This format will allow for an automatic use of the corpus, for training or as a test suite. Figure 2 shows the .xls and .txt formats.

4.3 Agreement

As previously mentioned, three annotators compiled and annotated the corpus. Their backgrounds include linguistics, translation and computational linguistics. Throughout the process of compilation and annotation, the annotators worked closely together to discuss the corpus compilation and also what issues should be tagged. Disagreements were discussed and resolved, and then the annotation process would resume. This process helped to refine the list of issues as well as to develop and finalise the guidelines. The corpus annotation carried out by the three first annotators was corrected at the final stage in order to ensure that it follows the established version of the guidelines.

In order to reveal some possible weaknesses of the annotation guidelines and the decision tree, another expert annotator was involved at the final stage. The fourth annotator worked with 9% of the documents from the original collection, where at least one document of each domain was selected randomly. The annotation was done according to the guidelines and the decision tree used by the first three annotators (see Figure 1). During the annotation process, the annotator was given the guidelines, decision tree and was explained what the goal of the annotation was, but was not allowed to communicate with the other annotators. We then calculated inter-annotator agreement using Cohen's Kappa (Cohen, 1960) treating the first annotation (performed by the three annotators) as the gold standard.

Results show that the overall Kappa score was 0.61 meaning that, by using the guidelines and the decision tree on a first try, we could reach a substantial agreement (Landis and Koch, 1977). We note that the majority of disagreement cases are related to agreeing whether or not a sentence contains an issue to be annotated, while our gold

[23] Modifications and translation are not provided in this format.

Issue	Explanation (solution)	Translation & notes
Reference	it = support group	o grupo de suporte
Ellipsis	I do = I think	Eu acho
Gender	it = feminine	Ela
Number	surrender = plural	Entreguem-se
Lexical ambiguity	paper = news (wrong) vs research article (right)	O jornal vs O artigo
Terminology	wind farm = farm (wrong) vs park (right)	Fazenda eólica vs Parque eólico

Table 8: Annotation format for every context-related issue.

Example .xls format

ID	text = source	issue	explanation	modification	extra explanations and notes
<seg id="1">	Willa G. Tsarones	X	X	subsititution = William >> Willia changed gender to add gender issues to the translation	
<seg id="2">	Reviewed in the United States on November 24, 2017	X	X		
<seg id="3">	I really liked this waffle maker after receiving it.	X	X		
<seg id="4">	I felt it was designed well and is highly functional and easy to use.	reference gender gender	it = waffle maker it = feminine well designed = feminine		aqui o "designed" vira "desenhado", então deve-se aplicar no feminino para "máquina"
<seg id="5">	I make 2 waffles twice every weekend with this Cusinart WAF-F20 and have been doing so for the last 10 months with great success.	X	X		
<seg id="6">	I really liked it.	reference	it = waffle maker		
<seg id="7">	Then, one weekend after cleanup, I looked at the rear support post and stared for awhile and asked myself "What an I looking at?"	X	X		
<seg id="8">	I was dumbfound.	gender	dumbfound = femine		

Example .txt format

```
<seg id="1">    Willa G. Tsarones   X   X
<seg id="2">    Reviewed in the United States on November 24, 2017  X   X
<seg id="3">    I really liked this waffle maker after receiving it.    X   X
<seg id="4">    I felt it was designed well and is highly functional and easy to use.   reference   it= waffle maker  gender    it=feminine gender  well designed=feminine
<seg id="5">    I make 2 waffles twice every weekend with this Cusinart WAF-F20 and have been doing so for the last 10 months with great success.   X   X
<seg id="6">    I really liked it.  reference   it=waffle maker
<seg id="7">    Then, one weekend after cleanup, I looked at the rear support post and stared for awhile and asked myself "What an I looking at?"   X   X
<seg id="8">    I was dumbfound.    gender  dumbfound=femine
```

Figure 2: Example of one excerpt of the corpus in the .xls and .txt format

standard has 170 issues annotated in this portion of the corpus, the fourth annotator found 106 issues. After this IAA was calculated, we discussed the annotation produced by annotator 4 and revised the corpus.

5 Conclusion

We have presented a document-level corpus annotated with context-aware issues when translating from EN into PT-BR, namely gender, number, ellipsis, reference, lexical ambiguity, and terminology. This first version of the corpus contains 60 documents, with 3680 sentences, in six different domains: subtitles, literary, news, reviews, medical and legislation. To the best of our knowledge, this is the first corpus of its kind.[24]

With the rise in NMT quality and the claims of human parity, the need to move towards a more fine-grained evaluation involving the whole document is beyond question. Moreover, with the lack of resources for the document-level MT area, this document-level corpus can be used as a challenge test set for evaluation and as a training/testing corpus for MT as well as for deep linguistic analysis of context issues. We believe that the annotation can be also used for closely-related languages such as Spanish.

We intend to increase the corpus, adding more documents, domains and more context-aware issues. The full translation into PT-BR is ongoing, and we want to annotate it for other languages, starting with the Romance language family.

[24]The corpus and guidelines will be freely available at https://github.com/SheilaCastilho/DELA-Project

Acknowledgements

We would like to than Helena Moniz and Vera Cabarrão for the fruitful discussions and invaluable help. This project was funded by the Irish Research Council (GOIPD/2020/69). ADAPT, the Science Foundation Ireland Research Centre for AI-Driven Digital Content Technology at Dublin City University, is funded by the Science Foundation Ireland through the SFI Research Centres Programme (Grant 13/RC/2106_P2).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly

Learning to Align and Translate. In *Proceedings of ICLR*, San Diego, CA.

Mona Baker. 2006. Contextualization in translator-and interpreter-mediated events. *Journal of pragmatics*, 38(3):321–337.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (WMT 19)*, pages 1–61, Florence, Italy.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Kathryn Bock, Janet Nicol, and J Cooper Cutting. 1999. The ties that bind: Creating number agreement in speech. *Journal of Memory and Language*, 40(3):330–346.

Xinyi Cai and Deyi Xiong. 2020. A test suite for evaluating discourse phenomena in document-level neural machine translation. In *Proceedings of the Second International Workshop of Discourse Processing*, pages 13–17, Suzhou, China. Association for Computational Linguistics.

Sheila Castilho. 2020. On the same page? comparing inter-annotator agreement in sentence and document level human machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1150–1159, Online. Association for Computational Linguistics.

Sheila Castilho. 2021. Towards document-level human MT evaluation: On the issues of annotator agreement, effort and misevaluation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 34–45, Online. Association for Computational Linguistics.

Sheila Castilho, Maja Popović, and Andy Way. 2020. On Context Span Needed for Machine Translation Evaluation. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC'20)*, Marseille, France.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Anne Stevens Fishman. 1978. The effect of anaphoric references and noun phrase organizers on paragraph comprehension. *Journal of Reading Behavior*, 10(2):159–170.

Yvette Graham, Timothy Baldwin, Meghan Dowling, Maria Eskevich, Teresa Lynn, and Lamia Tounsi. 2016. Is all that glitters in machine translation quality estimation really gold? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3124–3134, Osaka, Japan. The COLING 2016 Organizing Committee.

François Grosjean, Jean-Yves Dommergues, Etienne Cornu, Delphine Guillelmon, and Carole Besson. 1994. The gender-marking effect in spoken word recognition. *Perception & Psychophysics*, 56(5):590–598.

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A pronoun test suite evaluation of the English–German MT systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 570–577, Belgium, Brussels. Association for Computational Linguistics.

M.A.K. Halliday and C.M.I.M. Matthiessen. 2013. *Halliday's Introduction to Functional Grammar*. LSE International Studies. Taylor & Francis.

Judith Kast-Aigner. 2018. *A Corpus-Based Analysis of the Terminology of the European Union's Development Cooperation Policy: with the African, Caribbean and Pacific Group of States*. Peter Lang International Academic Publishers.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

António V Lopes, M Amin Farajian, Rachel Bawden, Michael Zhang, and André F T Martins. 2020. Document-level Neural MT: A Systematic Comparison. In *22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has Machine Translation Achieved Human Parity?

A Case for Document-level Evaluation. In *Proceedings of EMNLP*, pages 4791–4796, Brussels, Belgium.

Alan Melby and Christopher Foster. 2010. Context in translation: Definition, access and teamwork. *The International Journal for Translation & Interpreting Research*, 2.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

RCOG. Managing premenstrual syndrome (pms) - patient information leaflet. https://www.rcog.org.uk/en/patients/patient-leaflets/managing-premenstrual-syndrome-pms/. London: March 2018, used with the permission of the Royal College of Obstetricians and Gynaecologists. Last acessed August 2021.

Jennifer Rodd. 2018. Lexical ambiguity. *Oxford handbook of psycholinguistics*, pages 120–144.

Kateřina Rysová, Magdaléna Rysová, Tomáš Musil, Lucie Poláková, and Ondřej Bojar. 2019. A test suite and manual evaluation of document-level NMT at WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 455–463, Florence, Italy. Association for Computational Linguistics.

J.C. Sager and B. Nkwenti-Azeh. 1990. *A Practical Course in Terminology Processing*. J. Benjamins Publishing Company.

Dimitar Shterionov, Riccardo Superbo, Pat Nagle, Laura Casanellas, Tony O'dowd, and Andy Way. 2018. Human versus Automatic Quality Evaluation of NMT and PBSMT. *Machine Translation*, 32(3):217–235.

Steven L Small, Garrison W Cottrell, and Michael K Tanenhaus. 2013. *Lexical Ambiguity Resolution: Perspective from Psycholinguistics, Neuropsychology and Artificial Intelligence*. Elsevier.

Karin Sim Smith. 2017. On Integrating Discourse in Machine Translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 110–121.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS*, pages 3104–3112, Montreal, Canada.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of WMT*, pages 113–123, Brussels, Belgium.

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 Conference on Neural Information Processing Systems (NeurIPS 2017)*, pages 5998–6008, Long Beach, CA.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. SAO WMT19 test suite: Machine translation of audit reports. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 481–493, Florence, Italy. Association for Computational Linguistics.

Longyue Wang. 2019. *Discourse-aware neural machine translation*. Ph.D. thesis, Ph. D. thesis, Dublin City University, Dublin, Ireland.

Multilingual Domain Adaptation for NMT: Decoupling Language and Domain Information with Adapters

Asa Cooper Stickland* University of Edinburgh a.cooper.stickland@ed.ac.uk
Alexandre Bérard NAVER LABS Europe first.last@naverlabs.com
Vassilina Nikoulina NAVER LABS Europe first.last@naverlabs.com

Abstract

Adapter layers are lightweight, learnable units inserted between transformer layers. Recent work explores using such layers for neural machine translation (NMT), to adapt pre-trained models to new domains or language pairs, training only a small set of parameters for each new setting (language pair or domain). In this work we study the compositionality of language and domain adapters in the context of Machine Translation. We aim to study, 1) parameter-efficient adaptation to multiple domains and languages simultaneously (full-resource scenario) and 2) cross-lingual transfer in domains where parallel data is unavailable for certain language pairs (partial-resource scenario). We find that in the partial resource scenario a naive combination of domain-specific and language-specific adapters often results in 'catastrophic forgetting' of the missing languages. We study other ways to combine the adapters to alleviate this issue and maximize cross-lingual transfer. With our best adapter combinations, we obtain improvements of 3-4 BLEU on average for source languages that do not have in-domain data. For target languages without in-domain data, we achieve a similar improvement by combining adapters with back-translation. Supplementary material is available at https://tinyurl.com/r66stbxj.

1 Introduction

Multilingual Neural Machine Translation (NMT) has made a lot of progress recently (Johnson et al., 2017; Bapna and Firat, 2019; Aharoni et al., 2019; Zhang et al., 2020; Fan et al., 2020a) and is now widely adopted by the community and MT service providers. Multilingual NMT models handle multiple language directions at once and allow for knowledge transfer to low-resource languages. Machine translation systems often need to be adapted to specific domains like legal or medical text. However, when adapting multilingual systems, in-domain data for most language pairs might not exist. We would like to be able to leverage data in a subset of language pairs to transfer domain knowledge to other languages.

Straightforward methods of domain adaptation include fine-tuning (Freitag and Al-Onaizan, 2016) or usage of domain tags (Kobus et al., 2017; Britz et al., 2017) for different domains. For these methods each new domain request would require retraining the whole model, which is a costly procedure. And naive training on a subset of languages typically reduces performance on all other languages (Garcia et al., 2021), a phenomenon known as 'catastrophic forgetting' (McCloskey and Cohen, 1989).

An alternative technique for adapting such models to new language-pairs or domains are 'adapter layers' (Bapna and Firat, 2019), lightweight, learnable units inserted between transformer layers. A previously trained large multilingual model can be adapted to each language-pair by learning only these small units, and keeping the rest of the model frozen. This procedure also allows for the incremental adding of new language pairs and/or domains to the pre-trained model, reducing the cost of adaptation. Previous studies have shown it is possible to combine language-specific (as opposed to language-pair specific) adapters (Philip et al., 2020), or language and task adapters (Pfeiffer et al., 2020) trained independently, enabling zero-shot compositions of adapters. Our ultimate goal is, for ease of deployment and storage, a single model that can handle all languages and domains. In this work we analyse how to combine *language adapters* with *domain adapters* in multilingual NMT, and study to what extent the domain knowledge can be transferred across languages.

First, we show it is hard to decouple language

*Work done during an internship at NAVER LABS Europe.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 578–598
November 10–11, 2021.

knowledge from domain knowledge when fine-tuning multilingual MT systems on new domains. In Section 5.2 we demonstrate that adapters learnt on a subset of language pairs fail to generate into languages not in that subset. Such generation into the wrong language is referred to as 'off-target' translation. We additionally find combinations of domain and language adapters not seen at training time lead to bad performance. We examine how adapter placement and other techniques can improve the compositionality of language and domain adapters when dealing with source or target languages that do not have in-domain data (which we refer to throughout this work as "**out-of-domain languages**"). Our key contributions are:

- We examine domain adaptation capacity in the multi-lingual, multi-domain setting. We find that encoder-only adapters can be just as effective as default adapters added in every layer, and that composing domain adapters with language adapters outperforms language adapters alone, although fine-tuning with domain tags performs better for most domains.
- We improve the cross-lingual transfer of domain knowledge for adapters. We analyse different language and domain adapter combinations that improve performance and reduce off-target translations. Our best results for translation into out-of-domain languages use decoder-only domain adapters, regularisation with domain adapter dropout, and data augmentation with English-centric back-translation.

2 Related Work

Cross-lingual transfer Many works have demonstrated that large pre-trained multilingual models (Devlin et al., 2019; Conneau et al., 2020; Liu et al., 2020) fine-tuned on high-resource languages (or language pairs) can transfer to lower-resource languages in various tasks: Natural Language Inference (Conneau et al., 2018), Question Answering (Clark et al., 2020), Named Entity Recognition (Pires et al., 2019; K et al., 2020), Neural Machine Translation (Liu et al., 2020) and others (Hu et al., 2020).

Domain adaptation in NMT Domain adaptation has been discussed extensively for bilingual NMT models. A typical approach is to fine-tune a model trained on a large corpus of 'generic' data on a smaller in-domain corpus (Luong and Manning, 2015; Neubig and Hu, 2018). A common technique to make use of monolingual in-domain data is to do back-translation (Sennrich et al., 2016a; Berard et al., 2019a; Jin et al., 2020). Although effective, it is expensive to create back-translated data, especially when one needs to cover multiple language pairs. Multi-domain models can be trained with domain tags (Kobus et al., 2017; Britz et al., 2017; Berard et al., 2019a; Stergiadis et al., 2021) that can encode domain-specific information. However, domain tags do not allow *incrementally* adding new domains to a model: each new domain adaptation requires retraining the full model (as opposed to adapter layers that can be trained independently for each language/domain). There are a number of works (Jiang et al., 2020; Britz et al., 2017; Dabre et al., 2020) trying to explicitly decouple domain-specific representations from domain independent representations in bilingual settings. In our work we try to decouple language and domain specific representations through adapter layers.

Adapter layers Bapna and Firat (2019) introduce adapter layers for NMT as a lightweight alternative to fine-tuning. They study both adding language-pair specific adapters to multilingual NMT models to match the performance of a bilingual version, and domain-specific adapters for parameter-efficient domain adaptation. Philip et al. (2020) train adapters for each *language* instead of *language-pair* and show that composing such adapters improves zero-shot translation in English-centric settings, and can adapt a model to all language directions in a scalable way. Pfeiffer et al. (2020, 2021) study adapter layers for pre-trained language models evaluated on NLU tasks. They show it is possible to compose language and task adapters. Combining language adapters trained with a masked language modelling objective for language x and task adapters trained on a classification task in language y can transfer to classification in language x. We have a similar objective to Pfeiffer et al. (2020), but for NMT where in addition to encoding sentences we need to generate text for new language and domain combinations.

To the best of our knowledge none of the works above study composing language and domain adapters for generation tasks (such as translation) which is the goal of this work.

3 Composing Adapter Modules

Adapter modules (Rebuffi et al., 2017; Houlsby et al., 2019) are randomly initialised modules inserted between the layers of a pre-trained network and fine-tuned on new data. An adapter layer is typically a down projection to a bottleneck dimension followed by an up projection to the initial dimension, which we write as $\text{FFN}(\mathbf{h}) = W_{\text{up}} f(W_{\text{down}}\mathbf{h})$, with $f(\cdot)$ a non-linearity. The bottleneck controls the parameter count of the module; typically NMT requires slightly larger parameter counts than classification to match fine-tuning (Bapna and Firat, 2019; Cooper Stickland et al., 2021). With a residual connection and a near-identity initialization the original model is (approximately) retained at the beginning of optimization, keeping at least the performance of the parent model.

3.1 Stacking Domain and Language Adapters

In this work we study 'stacking' adapter modules, i.e. each language and domain has a unique adapter module associated with it. When passing a batch with source language x, target language y, and domain z, we only 'activate' the adapters for $\{x, y, z\}$. The encoder adapters for x and decoder adapters for y are activated.

We mostly follow the architecture of Bapna and Firat (2019). Language adapters LA are defined as:

$$\text{LA}(\mathbf{h}_l) = \text{FFN}_{\text{lg}}(\text{LN}_{\text{lg}}(\mathbf{h}_l)) + \mathbf{h}_l \quad (1)$$

where $\mathbf{h}_l$ is the Transformer hidden state at layer l and LN_{lg} is a newly initialised layer-norm. Let $\mathbf{z} = \text{LA}(\mathbf{h}_l)$; when stacking domain and language adapters, the layer output $\mathbf{h}_{l,\text{out}}$ is given by:

$$\mathbf{h}_{l,\text{out}} = \text{FFN}_{\text{dom}}(\text{LN}_{\text{dom}}(\mathbf{z})) + \mathbf{z} \quad (2)$$

For all models without any stacking we obtain layer output as in Eq. 2 but replace $\text{LA}(\cdot)$ with the identity operation.

Pfeiffer et al. (2020) use a different formulation that empirically performed well for them, but that in initial experiments produced worse results in our setting. We list the corresponding equations and results in Appendix B and Appendix D.

3.2 Improving the Compositionality of Adapters

In our initial experiments (Section 5.2) we found that (unlike Pfeiffer et al., 2020) naive stacking of language and domain adapters does not work very well for unseen combinations of language and domains, and often results in off-target translation (i.e. translations into the wrong language). Therefore, we study several strategies to improve the compositionality of adapters in the context of NMT:

1) Using **decoder-only** domain adapters when translating from an out-of-domain source language into an in-domain[1] target language, and **encoder-only** domain adapters when translating from an in-domain source language into an out-of-domain target language. This means we never stack together a combination of language and domain adapter that was not seen at training time. We also find empirically that decoder-only adapters work well with back-translation, perhaps because they can 'ignore' the noisy synthetic source-side data.

2) **Domain adapter dropout** (DADrop). Similar to layer-drop (Fan et al., 2020b) but specialised to adapter layers, or AdapterDrop (Rücklé et al., 2020) but without targeting specific layers, we randomly 'drop' (i.e. skip) the domain adapter[2] and only pass the hidden state through the language adapter. This means the adapter stack in the layer above can more easily adapt to unfamiliar input, and encourages domain and language adapters to be more independent of each other.

3) **Data augmentation**. We often have access to monolingual data in a domain even when no parallel data is available. In this work we leverage English-centric back-translation (BT), i.e. translating monolingual data in some languages into English (thus avoiding the more expensive step of translating from each language into every other language). We examine the ability of such data to help cross-lingual transfer to unseen combinations of source and target language (BT means we have artificial data for every language in combination with English). We briefly explore '*denoising auto-encoder*' style objectives as in unsupervised MT (Lample et al., 2018) or sequence-to-sequence pre-training (Lewis et al., 2020).

[1] Reminder we refer to the subset of languages we have parallel data for in a particular domain as 'in-domain', and all other languages as 'out-of-domain'.

[2] We could additionally drop the language adapter, but since this was frozen in many experiments we limit ourselves to domain adapters for simplicity

4 Experimental Settings

4.1 Data

For studying the domain transfer across languages we select four diverse domains that have data available in most language directions: translations of the Koran (**Koran**); medical text from the European Medicines Agency (**Medical**); translation of TED Talks transcriptions (**TED**); various technical IT text, e.g. the Ubuntu manual (**IT**). All data was obtained from the OPUS repository (Tiedemann, 2012). We create validation and test sets of around 2000 sentences each, and avoid overlap with training data (including parallel sentences in any language) with a procedure described in Appendix A. Note that Medical, Koran and IT are from the same source as those of Aharoni and Goldberg (2020), although the train/test splits are different due to expanding the number of languages and wanting a consistent pipeline for obtaining the data.

Domain	Langs.	Avg size (lines)
ParaCrawl	12	125M
Koran	10†	52k
Medical	11‡	500k
IT	12	196k
TED	12	138k

Table 1: Basic statistics for the datasets we use; number of languages covered, and average number of training examples across all language directions. †: missing nb & da, ‡: missing nb.

4.2 Models

In **multilingual settings** we concentrate on 12 high-resource European languages[3] due to the availability of domain-specific parallel data for most language pairs. Our **baseline model** is a Transformer Base (Vaswani et al., 2017) trained on English-centric ParaCrawl v7.1 data (Bañón et al., 2020) with all 12 languages (803M line pairs in total). It is trained with fairseq (Ott et al., 2019) for 800k updates, with a batch size of maximum 4000 tokens and accumulated gradients over 64 steps (Ott et al., 2018).[4] The source/target embeddings are shared and tied with the output layer. We tokenize the data with a shared BPE model of size 64k with inline casing (Berard et al., 2019b) Both the multilingual models and BPE model are trained with temperature-based sampling with $T = 5$ (Arivazhagan et al., 2019). We calculate all BLEU scores with Sacrebleu[5] (Post, 2018). On the recommendation of Marie et al. (2021) we additionally report chrF (Popović, 2015) calculated using Sacrebleu[6] for most models in the Appendix. We use adapter bottleneck size of 1024 unless stated otherwise, and when using DADrop (Section 3.2) use a 20% chance of skipping the domain adapter.

We additionally train monolingual language adapters (Philip et al., 2020) for all 12 languages on multi-parallel ParaCrawl data, which we obtain by aligning all languages through their English side, like Freitag and Firat (2020). The adapters are trained for another 1M steps, without accumulated gradients. We report the results of models fine-tuned on both all the domains simultaneously, or each domain separately, with access to in-domain data available for all the languages. Both serve as a potential upper bound for cross-lingual transfer.

We train the same model (i.e. with access to all languages) with domain tags: one special token per domain prepended to each source sequence (Kobus et al., 2017). We also measure the cross-lingual transfer ability of domain tags, by training a model with domain tags on all 4 domains but with in-domain data in only 4 languages (fr, de, cs and en). Because the latter model exhibits catastrophic forgetting issues in the other languages, we also train the same model with ParaCrawl data in all language directions (with a "paracrawl" domain tag). ParaCrawl line pairs are sampled with probability 0.5. More training hyper-parameters are given in Appendix A.

4.3 Our model pipelines

We perform two series of experiments.

Multilingual multi-domain models. Firstly, we experiment with different ways of multi-domain adaptation of multilingual models. We adapt the English-centric ParaCrawl pre-trained model to four domains (Koran, Medical, IT and TED) and every language direction simultaneously. We test models with language adapters, language + domain adapters, and domain tags. There is no cross-lingual domain transfer needed[7] since all language

[3] {cs, da, de, en, es, fr, it, nb, nl, pl, pt, sv}

[4] This corresponds to an effective batch size of ≈207k tokens and training length of 7 epochs.

[5] Signature: BLEU+case.mixed+lang.m2m-en+numrefs.1+smooth.exp+tok.13a+version.1.5.0.

[6] Signature: chrF2+numchars.6+space.false+version.1.5.1

[7] There is obviously cross-lingual domain transfer that may take place when all the domains are trained jointly, but we do

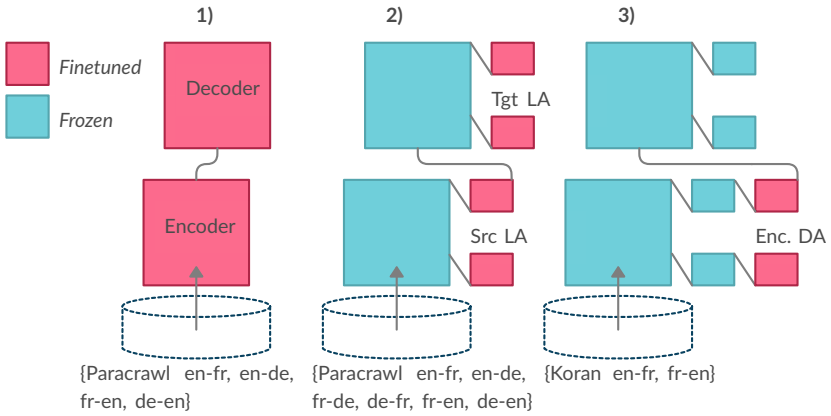


Figure 1: Toy diagram showing one of our proposed pipelines for training language and domain adapters, on a example subset of languages: {en,fr,de}, with 'domain-agnostic' data from ParaCrawl and specialised data from the Koran. Red indicates a fine-tuned model component, blue indicates a frozen component. LA = language adapter, DA = domain adapter. From left to right we show: 1) Training an encoder-decoder model with English-centric ParaCrawl. 2) Training mononlingual language adapters with multiparallel Paracrawl data. 3) Training domain adapters stacked on language adapters in the encoder, on a subset (here {en, fr}) of languages for the domain of interest (e.g. Koran). Here we show domain adapters added only to the encoder, but we consider various other configurations in this work.

directions are included in the training data. Results for this scenario are reported in Section 5.1.

Cross-lingual domain transfer. In the second experiment we try to decouple the notion of domain from language via analysing the zero-shot composition of domain and language adapters. This is described in a toy diagram in Figure 1. We first extend the baseline multilingual English-centric model with 12 (one for each language) monolingual language adapters (Philip et al., 2020) trained on multi-parallel ParaCrawl. We then test the cross-lingual domain transfer ability of our proposed combinations of adapters by training on data in a particular domain with a subset of four languages (referred to as '**in-domain**'; in Figure 1 *en* and *fr* would be in-domain). We test our model on all language directions from the set of all twelve languages. This will include cases where we don't have in-domain data for either the source or target language, which we refer to as '**out-of-domain**' (in Figure 1 *de* would be out-of-domain).

Finally, we extend the above mentioned scenario with back-translated (BT) data from *out-of-domain* languages into English. To create the BT data, we use the model with language adapters trained on ParaCrawl **(11)** (which has not seen any in-domain data) on the English-aligned training data for each language and domain, and use beam search with a beam size of 5. Results for this scenario are reported in Section 5.2.

To train language and domain adapters, we freeze all model parameters except for adapter parameters, and use a fixed learning rate schedule with learning rate 5×10^{-5}. Following Philip et al. (2020), when training language adapters without domain adapters we build homogeneous batches (i.e. only containing sentences for one language direction) and activate only the corresponding adapters. When training language *and* domain adapters together, we build homogeneous batches that only contain sentences for the same combination of language direction and domain.

5 Results and Discussion

First, in Section 5.1 we discuss the results of experiments testing the domain adaptation capacity of various models, assuming access to data for all language pairs. In Section 5.2 we analyse domain transfer across languages with adapters and other methods. We first demonstrate problems with cross-lingual generalisation during domain adaption for 'naive' methods, and then propose potential solutions. Note we concentrate on the medical domain and a particular language subset for convenience. Appendix D has results in

not explicitly study this in the first experiment.

ID	Model	IT	Koran	Medical	TED	Params (M)
(1)	Base (En-centric)	23.2	7.0	25.7	19.0	N/A
(2)	Finetuned	40.8	16.0	42.7	26.6	79
(3)	Finetuned + domain tags	**43.6**	**20.3**	**46.0**	27.2	79
(4)	Single adapter per layer ($d = 1024$)	39.6	14.7	41.8	26.2	12.6
(5)	LA ($d = 1365$)	42.0	17.6	43.7	26.8	202
(6)	LA ($d = 2048$)	42.2	18.1	43.8	26.9	303
(7)	LA + dec. DA ($d = 1024$)	42.1	18.5	43.6	27.1	177
(8)	LA + enc. DA ($d = 1024$)	42.3	19.3	43.8	27.5	177
(9)	LA + enc & dec. DA ($d = 1024$)	42.7	20.1	44.0	**27.7**	202

Table 2: BLEU scores averaged across all the language-directions for various multilingual multi-domain adaptation strategies, i.e. training on all language directions from the 12 languages and all domains. LA = language adapters, DA = domain adapters. 'Params (M)' refers to the number of trainable parameters in millions. Note that unlike in Table 3 the LA here are not pre-trained on ParaCrawl; they are trained jointly with domain adapters.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	44.3	43.9	44.8	44.2	44.2
(3)	FT (all langs & domains) + dom. tags	46.0	45.3	46.3	45.9	46.0
	Baselines					
(1)	Base (En-centric)	25.7	27.0	27.2	25.9	24.3
(11)	**(1)** + ParaCrawl LA	30.2	29.6	30.8	30.0	30.0
	Straightforward Methods					
(12)	**(1)** + Domain adapters only	23.0	44.7	37.8	13.4 (7%)	13.4 (11%)
(13)	Freeze LA + enc. & dec. DA	26.9	44.0	36.7	20.1 (71%)	19.9 (76%)
(14)	Freeze LA + enc. DA	29.6	42.6	34.0	27.0 (89%)	24.6 (88%)
(15)	Freeze LA + dec. DA	29.0	41.7	**40.7**	22.5 (77%)	22.0 (77%)
(16)	FT (all domains) + dom. tags	15.6	**46.8**	13.2 (55%)	12.0 (1%)	10.7 (2%)
	Improving Off-target Translation					
(17)	**(16)** + ParaCrawl	34.7	42.2	39.6	32.4	31.0
(18)	**(13)** + BT	33.9	43.2	36.8	35.9	28.0 (85%)
(19)	**(14)** + BT	32.5	41.8	35.0	34.7	26.8 (83%)
(20)	**(15)** + BT	**36.9**	40.9	38.2	36.4	**35.1**
(21)	**(13)** + DADrop	28.0	42.6	36.7	22.9 (82%)	21.5 (82%)
(22)	**(13)** + BT + DADrop	34.8	42.2	37.0	36.5	30.2
(23)	Unfreeze LA + dec. DA	14.3	45.8	36.6	0.0 (1%)	0.0 (2%)
(24)	**(23)** + DADrop	31.1	45.5	36.9	23.9 (82%)	27.8
(25)	**(23)** + DADrop + BT	35.2	44.5	33.4	**38.2**	31.8

Table 3: BLEU score of various models trained on the {en, fr, de, cs} subset of the Medical domain, except 'Oracle' models trained on all language pairs. LA = language adapters, DA = domain adapters. 'Out→in' is the average score when translating from an out-of-domain source language into {en, fr, de, cs}. 'In→out' corresponds to when the out-of-domain language is the target language. 'In→in' refers to average score when source and target are in the set {en, fr, de, cs}. 'Out→out' is the average score when both the source and target language are unseen during domain adaptation. We note percentage of **on-target** (correct language) translations in brackets, when it is less than 90% only.

other domains and language subsets, and also chrF (Popović, 2015) scores; we find similar trends to those reported in Section 5.2.

5.1 Multilingual multi-domain models

Table 2 reports the results from the challenging task of adapting a multilingual NMT model to multiple domains and language directions simultaneously. In this scenario, we assume access to in-domain data in all the language directions, and so we are testing the capacity of various models for domain adaptation, rather than cross-lingual transfer. Models are compared against a baseline **(1)** not trained on in-domain data.

We report the results for naive fine-tuning on the concatenation of in-domain parallel datasets for all the languages and all the domains **(2)**. On all domains we improve on these results by fine-tuning with domain tags **(3)** (a similar result to Jiang et al. (2020) in the bilingual setting). Fine-tuning with domain tags **(3)** outperforms the model with stacked adapters **(9)**. A fine-grained comparison of these models is in Figure 5 in the Appendix. For the IT and Medical domains the model with tags **(3)** is clearly better for all language directions. For the lowest resource domains, Koran and TED, most of the differences are not statistically significant, except for English-centric language pairs for TED, where the adapter model **(9)** is better. Exploring the combination of domain tags and adapters could be an interesting future research direction.

Stacking domain and language adapters **(9)** results in better performance than a model with the same parameter budget devoted to language adapters only **(5)**. We believe this is because it allows the model to (partially) decouple domain from language-specific information, and better exploit the allocated parameter budget. Even a higher capacity language adapter model **(6)** does not perform as well.

We also note that usage of encoder-only domain adapters **(8)** outperforms the decoder-only domain adapter model **(7)**. This is perhaps because the encoder representations influence the whole model (it is directly connected to the decoder at all layers with encoder-decoder attention) as opposed to the the decoder adapters that only impact decoder representations. We find a similar trend in bilingual domain adaptation, see Appendix C.

The strong performance of encoder-only adapters has interesting implications for inference speed. With an auto-regressive decoder, the computational bottleneck is on the decoder side. The encoder output is computed all at once, while computing the decoder output requires L steps, where L is the output length. This implies devoting more capacity to encoder adapters would achieve similar performance and faster inference (more details in Appendix C).

5.2 Cross-lingual Domain Transfer

To study the capacity of our models to transfer domain knowledge across languages, we perform domain adaptation using parallel datasets for a *subset* of language pairs, and evaluate on the test sets available for *all* language pairs. In this section we report the results for adaptation to the *medical* domain using the subset of all the language directions including {en, fr, de, cs} languages (Table 3). We refer to these languages as *in-domain* languages, and *out-of-domain* languages would include all the other languages, {de, nl, sv, es, it, pt, pl } (referred to as *In* and *Out* respectively in Table 3).

We report BLEU scores averaged across test sets of different categories of language-directions depending on whether the source/target language was observed during the domain adaptation training: In→in for language pairs observed during DA, Out→out for fully zero-shot DA performance, and In→out, Out→in for translation directions combining *in-domain* and *out-of-domain* languages.

First, we report the results for *Oracle* models providing an upper bound for the scores models could achieve with access to in-domain data for *all* the languages: model **(10)** was fine-tuned on *medical* data for all the language directions[8], and a model with domain tags **(3)** discussed in section 5.1.

Baseline models include the default multilingual English-centric model **(1)**, as well as model **(11)** with language adapters trained on multi-parallel ParaCrawl data. Comparing against this baseline shows us improvements from domain-specific (rather than language-specific) information.

Straightforward Methods We train several 'straightforward' adapter models for the subset of *in-domain* languages on the top of the baseline model, one with no language adapters, model **(12)**, and model **(13)** with domain *and* language adapters (where language adapters are frozen), stacking them in the encoder and decoder.

[8]This is different from the model **(3)** which was fine-tuned on *all* the domains and all the language directions.

Both of these models achieve good scores when translating into *in-domain* languages (the In→in and Out→in categories), on par or better than *Oracle* scores and much higher than the baselines. On the other hand they suffer from significant drops in performance when translating into *out-of-domain* languages (the In→out and Out→out categories).

The model **(16)** trained with tags on a subset of *in-domain* languages suffers from the same low performance translating into *out-of-domain* languages and additionally has low performance with out-of-domain *source* languages.

Looking closer at the translations of the above models, we see that many translations are either generated in English, copy the source language, or mix words between English and the true target language; see Table 4 in the Appendix for illustrative examples. We refer to this phenomenon as "off-target" translation. We report the percentage of translations generated in the correct target language in Table 3 when it is lower than 90% [9].

We believe this phenomenon is partly due to decoder domain adapters having never been exposed to *out-of-domain* language generation. Encoder domain adapters seem to be less sensitive to composition with new language adapters (as observed by Pfeiffer et al. (2020) for NLU tasks, and Table 3 in the Out→In column).

To investigate this, we train models **(14)** and **(15)** with encoder-only and decoder-only adapters. Figure 2 compares the performances of these models as well as model **(13)** trained with encoder and decoder domain adapters, **(14)**, **(15)** against the baseline model **(11)**. The decoder-only model **(15)** can better translate *from* out-of-domain languages and the encoder-only model **(14)** slightly improves for translations *into* out-of-domain languages. However the problem of off-target translation persists for both models and neither improves over ParaCrawl LA **(11)**. Therefore, we conclude that a straightforward combination of domain and language adapters leads to catastrophic forgetting both in the encoder and the decoder, but the encoder is less important for this effect.

Effect of data augmentation We train models **(17)**,**(18)** **(19)**, **(20)** with additional data (either a portion of ParaCrawl data, or back-translation of in-domain data) to alleviate potential forgetting of representations for *out-of-domain* languages. All of these models improve the translation quality into *out-of-domain* languages. The model with tags **(17)** reaches competitive results and can be considered as a strong baseline.

For models with back-translation data, the decoder-only adapter **(20)** model outperforms the encoder-only adapter **(19)** model on out-of-domain target languages (as opposed to the case without BT) and has the strongest results overall on translating into out-of-domain languages. While the BT models are trained on exactly the same data, this effect is possibly due the encoder adapters being more influenced by potentially noisy synthetic source-side data, whereas decoder adapters are more influenced by clean reference translations. The decoder-only BT model **(20)** improves over the baseline for all the language directions except for translation into English; see Figure 3.

We report results for the other data augmentation methods (see Section 3.2) in Appendix D; these only improve over the ParaCrawl LA baseline in limited settings.

Domain adapter dropout Models **(21)** and **(22)** trained with dropping domain adapters (DADrop; see Section 3.2) also allow to reduce catastrophic forgetting, although only combining DADrop with Data Augmentation (model **(22)**) allows to solve the problem of off-target translation. We also note slight decreases in *in-domain* performance for those models, perhaps due to underfitting.

Increasing adaptation capacity When naively increasing model capacity by unfreezing LA stacked with decoder DA **(23)**, the model seems to mostly devote this capacity to In→in category, and suffers on other language pair groups. This trend seems to be similar to the un-augmented model with tags **(16)**. However, once regularized with DADrop **(24)**, and augmented with back-translation **(25)** it reaches very competitive results.

Figure 4 shows fine-grained results for different models with DADrop, back-translation and unfrozen LA. Back-translation improves performance on the Out→out and In→out groups, but decreases performance on the Out→in group. Finally, unfreezing language adapters decreases the performance on Out→in but improves on the Out→out group.

[9] This percentage is computed against the reference translations that were correctly tagged by 'langdetect', a Python language identifier (`https://pypi.org/project/langdetect/`). This is to exclude very short and numerical examples which can be quite frequent in some domains.

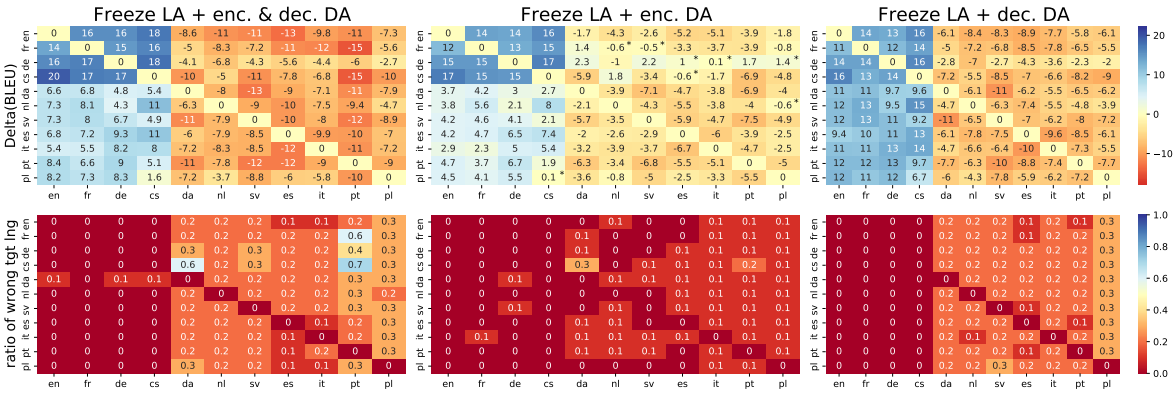


Figure 2: Comparing models with encoder-decoder adapters, encoder-only adapters and decoder-only adapters. x-axis shows the target language and y-axis shows the source language. Languages are grouped so the in-domain languages are in the top left corner. Top: Difference in BLEU compared to the baseline **(11)** (negative scores indicate a decrease w.r.t. the baseline, "*" indicates *not* statistically significant). Bottom: proportion translating into the wrong target language. Best viewed in .pdf form.

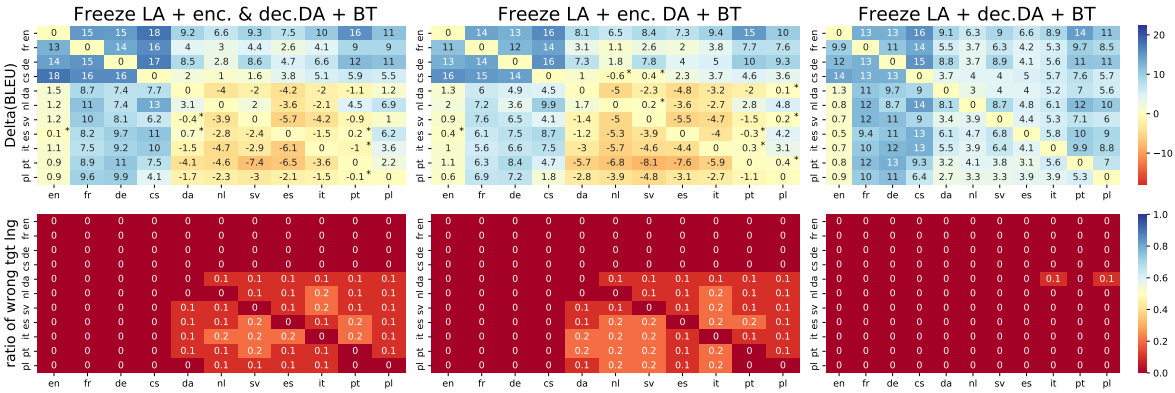


Figure 3: Comparing adapter models trained with back-translation. Top: Difference in BLEU compared to the baseline **(11)** ("*" indicates *not* statistically significant). Bottom: proportion translating into the wrong target language. See Figure 2 for more details.

Adapters vs. tags As mentioned previously, model **(17)** with tags augmenteed with ParaCrawl reaches competitive scores overall. Note that this model was trained on a concatenation of *all* the domains, unlike the models with adapters which were trained *only* on the medical domain. Therefore it has been exposed to more data overall. On the other hand, several of our models fine-tune only a single adapter per-layer and use frozen LA. Thus, encoder-only or decoder-only models only require 6.3 million tunable parameters, compared to 79 million for tag-based models. Additionally adapter models can easily be 'mixed-and-matched' by activating a particular adapter for a particular language pair. For example we could activate model **(15)** on 'Out→in' (out-of-domain source, in-domain target) data, model **(18)** on in-domain data and model **(20)** otherwise. Such models could easily be extended to new domains by training more adapters, in contrast to tag-based models which update all parameters for each domain adaptation request.

6 Conclusion

In this work we studied multilingual domain adaptation both in the full resource setting where in-domain parallel data is available for all the language pairs, as well as the partial resource setting, where in-domain data is only available for a small set of languages.

In particular, we study how to better compose language and domain adapter modules in the context of NMT. We find that while adapters for encoder architectures like BERT can be safely composed, this is not true for NMT adapters: domain adapters learnt in the partial resource scenario struggle to generate into languages they were not trained

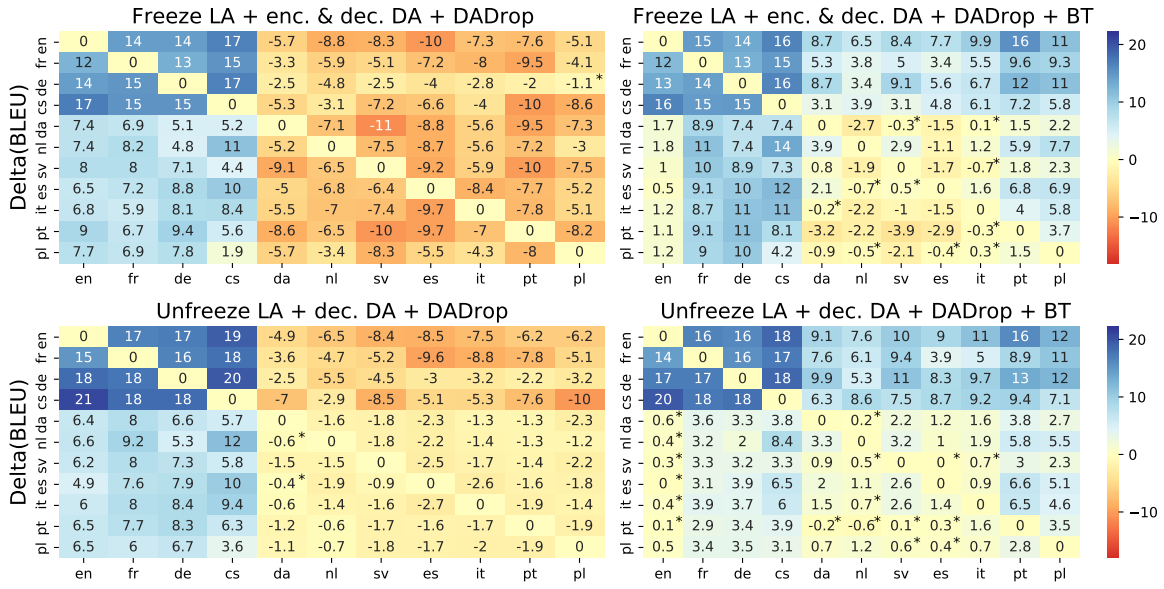


Figure 4: Comparing models with DADrop, back-translation and unfrozen language adapters. Difference in BLEU compared to the baseline **(11)** ("*" indicates *not* statistically significant). See Figure 2 for more details.

on, even though the original model they are inserted in was trained on those languages. We found that randomly dropping the domain adapter and back-translation can regularize the training and lead to less catastrophic forgetting for when generating into out-of-domain languages, although they do not fully solve the problem of off-target translation.

We experimented with different adapter placement and found that devoting additional capacity to encoder adapters can lead to better results compared to when the same capacity is shared between the encoder and the decoder. Similarly, in the partial resource scenario, models with encoder-only domain adapters suffer less from catastrophic forgetting when translating into out-of-domain languages. In contrast, decoder-only domain adapters perform well when translating *from* out-of-domain into in-domain languages, and combine well with back-translation, perhaps due to their ability to ignore noisy synthetic source data.

Finally we note that a model fine-tuned with domain tags serves as a very competitive baseline for multilingual domain adaptation. On the other hand, domain adaptation with adapters offers modularity, and allows incrementally adapting to new domains without retraining the full model. Future research directions could explore multi-task training combining parallel and monolingual in-domain data in other ways to alleviate the need for back-translation.

Our work is the first attempt to combine domain adapters and language adapters for a generation task (NMT). Although such combinations have shown to be successful for NLU tasks, obtaining good representations for generating unseen target languages proves to be a difficult problem. We believe a fine-grained study of where to use language or domain-specific capacity could lead to better cross-lingual domain transfer in future. Finally, we provide supplementary material to facilitate reproducibility.[10]

Acknowledgements

We would like to thank Laurent Besacier, Hady Elsahar, Matthias Gallé, Germán Kruszewski, Ahmet Ustun and all of the NAVER Labs Europe team for useful discussions. Asa Cooper Stickland was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

[10] https://tinyurl.com/r66stbxj

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. 2019a. Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 168–176, Hong Kong. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019b. Naver Labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. In *Transactions of the Association of Computational Linguistics*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP 2018*, pages 2475–2485.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2021. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3440–3453, Online. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation. *arXiv preprint arXiv:2001.01115*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020a. Beyond english-centric multilingual machine translation.

Angela Fan, Edouard Grave, and Armand Joulin. 2020b. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*,

pages 550–560, Online. Association for Computational Linguistics.

Xavier Garcia, Noah Constant, Ankur P. Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. *CoRR*, abs/2103.06799.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, volume 97, pages 2790–2799, Long Beach, California, USA.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1834, Online. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. A simple baseline to semi-supervised domain adaptation for machine translation.

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.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *CoRR*, abs/2106.15195.

M. McCloskey and N. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. 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, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, Long Beach, California, USA.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. *CoRR*, abs/2010.11918.

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.

Emmanouil Stergiadis, Satendra Kumar, Fedor Kovalev, and Pavel Levin. 2021. Multi-domain adaptation in neural machine translation through multidimensional tagging.

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 *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, California, USA.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

A Data and Hyper-parameters

For **bilingual** domain adaptation we use a Transformer Base (Vaswani et al., 2017) model trained for 12 epochs on German to English WMT20 data (47M parallel lines), with a joint BPE (Sennrich et al., 2016b) vocabulary of size 24k with inline casing (Berard et al., 2019b) (i.e. wordpieces are put in lowercase with a special token indicating their case.). For bilingual domain adaption we use the same datasets as Aharoni and Goldberg (2020), namely parallel text in German and English from five diverse domains: Koran, Medical, IT, Law and Subtitles.

For multilingual settings we use the following hyper-parameters. We share embeddings between encoder and decoder. We use the Adam optimizer (Kingma and Ba, 2014) with an inverse square root learning rate schedule for pre-training, and a fixed learning rate schedule for training adapters. We speed up training with 16 bit floating point arithmetic. We use label smoothing 0.1 and dropout 0.1. We train for either 20 epochs or 1 million updates, whichever corresponds to the smallest number of training updates. We use early stopping, checking performance after each epoch or every 100,000 training steps, and use average validation negative-log-likelihood on all of the training data (but not out-of-domain language data) as our criteria for choosing the best model. We otherwise use default Fairseq (Ott et al., 2019) parameters. We train all models on a single Nvidia V100 GPU, and training takes between 8 and 36 hours depending on dataset size.

In order to create validation and test splits that had no overlap with training data in any language, we first set aside a number of English sentences. Then we aligned all language pairs to these sentences, i.e. the German to French test set is composed of German and French sentences that share the same English sentence. Finally we remove all sentences in any language from the train splits of all parallel data if those sentences are aligned with any English sentences in the subset we set aside for validation/test splits. Both validation sets and test sets contain around 2000 examples for every domain and language-pair.

B MAD-X Style Stacking

Pfeiffer et al. (2020) use the following stacking formulation,

$$\mathrm{LA}(\mathbf{h}_l, \mathbf{r}_l) = \mathrm{FFN}_{\mathrm{lg}}(\mathbf{h}_l) + \mathbf{r}_l. \quad (3)$$

The residual connection $\mathbf{r}_l$ is the output of the Transformer's feed-forward layer whereas $\mathbf{h}_l$ is the output of the subsequent layer normalisation. When stacking domain and language adapters the layer output is given by applying the model's pre-trained layer norm $\mathrm{LN}_{\mathrm{pre}}$,

$$\mathbf{h}_{l,\mathrm{out}} = \mathrm{LN}_{\mathrm{pre}}(\mathrm{FFN}_{\mathrm{dom}}(\mathrm{LA}(\mathbf{h}_l, \mathbf{r}_l)) + \mathbf{r}_l) \quad (4)$$

and using the output of the Transformer's feed-forward layer as a residual instead of the language adapter output. We refer to this as '**MAD-X**' style after Pfeiffer et al. (2020). This leaves the layer output 'closer' to the pre-trained model, with the same layer-norm and residual connection, contrary to Eq. 2 which has a newly initialised layer-norm and a residual connection. For all models without any stacking we obtain layer output as in Eq. 4 but replace $\mathrm{LA}(\cdot)$ with the identity operation.

C Additional Results for Bilingual Domain Adaptation

Before studying multilingual domain adaptation, we validate some of our ideas on a simpler, *bilingual* German → English domain adaptation setting. Table 11 reports the results of this experiment. First, we note that *encoder-only* adapters perform similarly to *encoder & decoder* adapters, while *decoder-only* adapters perform worse.

Moreover, adding adapters to only the last three layers of the encoder almost matches the performance of adapting every layer, while adding adapters to the first three layers decreases performance. We believe this is because the last encoder layer directly influences every layer of the decoder through cross-attention.

Table 12 presents results of bilingual domain adaption with smaller adapter bottleneck dimension. The same trends emerge: encoder-only adapters perform better, and the last three layers of the encoder are better than the first three. The last three encoder layers also perform better than the first three for a multilingual model, see Table 7 models **(38)** and **(39)**. Interestingly the multilingual last three encoder layer DA model is roughly halfway between encoder-only and decoder-only on Out→in and In→out performance, suggesting it might be a useful compromise between the two.

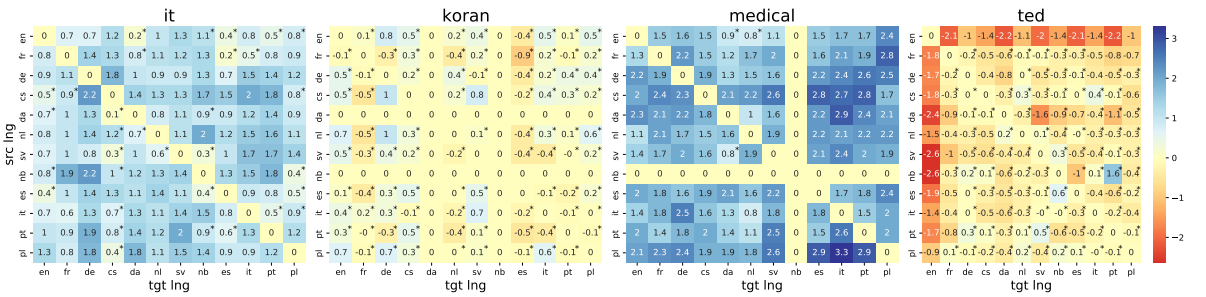


Figure 5: Difference in BLEU score for each domain between the model trained with adapters **(9)** and model trained with domain tags **(3)**, for the multilingual multi-domain models. A positive number corresponds to the case where model **(3)** has higher score than the model **(9)**. The "*" indicates the cases when the difference is *not* statistically significant.

D Additional Results for Cross-lingual Transfer

Does language diversity increase transfer? Table 5 compares models trained on a mix of language families (fr, de, cs, en) and mostly romance languages (fr, it, es, en) to test whether diversity of languages in our in-domain training set improved transfer. Positive numbers in this table indicate diversity of training languages improves performance. Diversity helps for translating out-of-domain languages into in-domain. We have unclear results for when both source and target are out-of-domain; it seems when using back-translation (BT), i.e. when all languages have been seen (albeit with artificial English parallel data) diversity helps, but without BT it mostly hurts performance. We speculate that training on mostly romance languages means the domain adapter encodes less 'language information', but leave further exploration to future work.

Additional results and metrics We present additional results for the setting discussed in Section 5.2 of the main paper in Table 9 (Koran domain), Table 10 (Koran results for the romance language subset), and Table 7 (additional Medical results). We use the chrF metric as discussed in the main paper, and find the conclusions based on BLEU score are unchanged. For the Koran domain, we see similar trends with decoder-only domain adapters (DA) performing best on out-of-domain source to in-domain target languages, and vice versa for encoder-only DA. Additionally we see as before that combining BT with decoder-only DA works the best, and achieves the highest overall performance. We report on-target (correct language) percentage for all medical domain models in Table 8.

We briefly experiment with denoising objectives, where we simply copy target data in out-of-domain languages to the source side (and optionally add 'noise' to the source side, e.g. swap tokens or mask tokens (Lewis et al., 2020)). Although we got reasonable improvements (models **(42)** and **(41)**) for out-of-domain target languages, we were mostly unable to improve over the pre-trained ParaCrawl LA, and so concentrate on back-translation.

We experiment with a setting where we jointly train on all language directions for IT, Koran and TED Talks domains and a subset of languages for Medical, and similarly with only a subset of Koran (models **(43)**, **(44)** etc.). These models stack language and domain adapters. Such models don't require any pre-trained LA, and improve out-of-domain performance and decrease off-target translation compared to freezing ParaCrawl LA and training DA. However these scores are still worse than simply using pre-trained 'domain-agnostic' ParaCrawl LA **(11)**.

source (fr)	ref (pt)	**(12)**	**(13)**
La durée du traitement dépend de la nature et de la sévérité de l' infection et de la réponse observée.	A duração do tratamento depende da natureza e da gravidade da infecção e da resposta verificada.	The duration of treatment depends on the nature and severity of the infection and on the response observed.	A duração do tratamento depende da nature e severidade da infecção e da resposta observed.
Insuman Comb 50 40 UI/ ml suspension injectable en flacon	Insuman Comb 50 40 UI/ ml, Suspensão injectável num frasco para injectáveis	Insuman Comb 50 40 IU/ ml suspension injectable en flacon	Insuman Comb 50 40 IU/ ml suspension for injection in vial
A quoi ressemble TAXOTERE et contenu de l' emballage extérieur TAXOTERE 80 mg, solution à diluer pour perfusion est une solution visqueuse, limpide, jaune à jaune marron.	Qual o aspecto de TAXOTERE e contéudo da embalagem TAXOTERE 80 - mg concentrado para solução para perfusão é uma solução viscosa transparente amarela ou amarela- acastanhada.	What TAXOTERE looks like and contents of the pack TAXOTERE 80 mg concentrate para solution for infusion is a solution visqueuse, limpida, de jaune à marron.	TAXOTERE 80 mg, Diluted for Solution for Infusion é uma solution viscous, limpa, yellow to marrom.

Table 4: Some examples of translations generated by straight-forward adapter training settings, in this case from a known source language, *fr* into a target language unseen during domain adaptation, *pt*, and for the medical domain.

Model	Out→ {en,fr}	Out→Out
Koran		
LA + Dec. DA	0.6	-0.9
LA + Dec. DA	0.3	-0.3
Unfr. LA + Dec. DA	0.1	-0.4
LA + Enc & Dec. DA	1.3	-1.3
Koran + BT		
LA + Dec. DA	0.4	0.2
LA + Dec. DA	1.9	0.9
Unfr. LA + Dec. DA	0.3	0
LA + Enc & Dec. DA	0.7	0.3
Medical		
LA + Dec. DA	0.8	-2.4
LA + Dec. DA	3.2	0.2
Unfr. LA + Dec. DA	0.2	0.1
LA + Enc & Dec. DA	3.2	-1.1
Medical + BT		
LA + Dec. DA	0.4	2.9
LA + Dec. DA	1.3	1.6
Unfr. LA + Dec. DA	0.5	3.1
LA + Enc & Dec. DA	0.5	1.5

Table 5: Difference in average BLEU score between models trained on a diverse subset of languages and models trained on mostly romance languages. Data source is noted in bold. Refer to the main paper for model definitions. Out→ {en,fr} corresponds to translation from an out-of-domain source language into {en,fr}. 'Out→Out' is the average score when both the source and target language are unseen during domain adaptation (choosing languages unseen by either subset).

ID	Model	IT	Koran	Medical	TED	Params (M)
(1)	Base (En-centric)	.456	.300	.488	.456	N/A
(2)	Finetuned	.623	.394	.625	.513	79
(3)	Finetuned + domain tags	**.645**	**.433**	**.646**	.517	79
(4)	Single adapter per layer ($d = 1024$)	.612	.382	.619	.512	12.6
(5)	LA ($d = 1365$)	.632	.408	.630	.517	202
(6)	LA ($d = 2048$)	.634	.411	.631	.517	303
(7)	LA + dec. DA ($d = 1024$)	.633	.412	.629	.518	177
(8)	LA + enc. DA ($d = 1024$)	.634	.426	.631	.522	177
(9)	LA + enc & dec. DA ($d = 1024$)	.636	.429	.632	**.523**	202

Table 6: chrF scores of various multilingual multi-domain adaptation strategies, i.e. training on all language directions from the 12 languages and all domains.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	.635	.631	.638	.635	.635
(3)	FT (all langs & domains) + domain tags	.646	.641	.648	.646	.646
	Baselines					
(1)	Base (En-centric)	.488	.500	.497	.493	.475
(11)	**(1)** + ParaCrawl LA	.537	.532	.540	.538	.535
	Straightforward Methods					
(12)	**(1)** + Domain adapters only	.400	.635	.575	.295	.287
(13)	Freeze LA + enc. & dec. DA	.468	.631	.566	.401	.400
(16)	FT (all domains) + dom. tags	.277	**.650**	.236	.283	.193
	Improving Off-target Translation					
(17)	**(16)** + ParaCrawl	.570	.622	.601	.556	.544
(26)	Unfreeze LA	.551	.639	.574	.514	.535
(27)	**(34)** + BT	.571	.636	.556	.594	.548
(15)	Freeze LA + dec. DA	.492	.613	**.605**	.432	.421
(20)	**(15)** + BT	**.586**	.608	.590	.584	**.577**
(28)	**(15)** + BT + DADrop	.584	.604	.587	.583	.576
(14)	Freeze LA + enc. DA	.518	.623	.548	.506	.477
(19)	**(14)** + BT	.548	.620	.567	.574	.498
(29)	**(14)** + BT + DADrop	.561	.614	.570	.579	.527
(30)	Freeze LA + enc. first 3 layers DA	.440	.618	.525	.379	.373
(31)	Freeze LA + enc. last 3 layers DA	.512	.622	.576	.472	.465
(21)	**(13)** + DADrop	.490	.621	.567	.447	.429
(32)	**(13)** + BT	.559	.626	.581	.582	.511
(22)	**(13)** + BT + DADrop	.569	.619	**.583**	.587	.534
(33)	**(13)** + BT + MAD-X style	.540	.625	.575	.561	.477
(23)	Unfreeze LA + dec. DA	.221	.641	.573	.010	.007
(24)	**(23)** + DADrop	.528	.639	.577	.452	.514
(25)	**(23)** + DADrop + BT	.573	.636	.559	.595	.550

Table 7: chrF score of various models trained on the {en, fr, de, cs} subset of the Medical domain. Some models are also included in the main paper.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	95%	95%	95%	95%	95%
(3)	FT (all langs & domains) + domain tags	95%	95%	96%	95%	95%
	Baselines					
(1)	Base (En-centric)	91%	93%	92%	91%	89%
(11)	**(1)** + ParaCrawl LA	95%	96%	96%	95%	95%
	Straightforward Methods					
(12)	**(1)** + Domain adapters only	40%	95%	93%	7%	11%
(13)	Freeze LA + enc. & dec. DA	81%	95%	93%	71%	76%
(16)	FT (all domains) + dom. tags	25%	96%	55%	1%	2%
	Improving Off-target Translation					
(17)	**(16)** + ParaCrawl	93%	95%	93%	93%	92%
(34)	Unfreeze LA	94%	95%	95%	93%	95%
(35)	**(34)** + BT	94%	96%	95%	95%	94%
(15)	Freeze LA + dec. DA	84%	95%	95%	77%	77%
(20)	**(15)** + BT	94%	95%	95%	94%	94%
(36)	**(15)** + BT + DADrop	94%	95%	95%	94%	94%
(14)	Freeze LA + enc. DA	90%	95%	93%	89%	88%
(19)	**(14)** + BT	90%	96%	94%	94%	83%
(37)	**(14)** + BT + DADrop	93%	95%	95%	94%	91%
(38)	Freeze LA + enc. first 3 layers DA	59%	95%	93%	35%	43%
(39)	Freeze LA + enc. last 3 layers DA	88%	95%	94%	83%	86%
(21)	**(13)** + DADrop	86%	95%	93%	82%	82%
(18)	**(13)** + BT	91%	95%	95%	94%	85%
(22)	**(13)** + BT + DADrop	93%	95%	95%	94%	91%
(40)	**(13)** + BT + MAD-X style	89%	95%	95%	92%	80%
(23)	Unfreeze LA + dec. DA	36%	96%	95%	1%	2%
(24)	**(23)** + DADrop	90%	95%	95%	82%	92%
(25)	**(23)** + DADrop + BT	94%	95%	95%	94%	94%

Table 8: On-target translation percentages of various models trained on the {en, fr, de, cs} subset of the Medical domain.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	.461	.437	.427	.487	.477
(3)	FT (all langs & domains) + domain tags	.433	.423	.409	.455	.438
	Baselines					
(1)	Base (En-centric)	.300	.307	.299	.306	.294
(11)	**(1)** + ParaCrawl LA	.334	.330	.328	.340	.335
	Straightforward Methods					
(12)	**(1)** + Domain adapters only	.246	.451	.349	.163	.150
(13)	Freeze LA + enc. & dec. DA	.165	.449	.137	.144	.089
	Improving Off-target Translation					
(16)	FT (all dom.) + dom. tags	.166	.436	.162	.143	.081
(17)	FT (all dom. + ParaCrawl) + dom. tags	.359	.410	.375	.351	.332
(34)	Unfreeze LA	.352	.454	.351	.322	.335
(15)	Freeze LA + dec. DA	.304	.404	**.385**	.249	.244
(41)	**(15)** + Mono data	.355	.390	.373	.342	.336
(20)	**(15)** + BT	.381	.399	.371	.387	.375
(36)	**(15)** + BT + DADrop	**.382**	.402	.373	.388	**.376**
(14)	Freeze LA + enc. DA	.319	.438	.328	.315	.266
(42)	**(14)** + Mono data	.347	.410	.338	.353	.324
(19)	**(14)** + BT	.365	.432	.353	.385	.330
(37)	**(14)** + BT + DADrop	.368	.425	.354	**.394**	.336
(18)	**(13)** + BT	.374	.434	.366	**.394**	.341
(22)	**(13)** + BT + DADrop	.381	.436	.369	.406	.349
(23)	Unfreeze LA + dec. DA	.224	.457	.351	.088	.138
(24)	**(23)** + DADrop	.339	**.458**	.354	.288	.320
(43)	Multi-domain dec. DA	.326	.403	.360	.304	.285
(44)	Multi-domain enc. DA	.337	.412	.360	.327	.297
(45)	Multi-domain enc. & dec. DA	.327	.417	.369	.302	.279

Table 9: chrF score of various models trained on the {en, fr, de, cs} subset of the Koran domain. LA = language adapters, DA = domain adapters. 'Out→in' is the average score when translating from an out-of-domain source language into {en, fr, de, cs}. 'In→out' corresponds to when the out-of-domain language is the target language. 'In→in' refers to average score when source and target are in the set {en, fr, de, cs}. 'Out→Out' is the average score when both the source and target language are unseen during domain adaptation. 'Mono data' refers to adding copied monolingual data for out-of-domain languages, and additionally multiparallel ParaCrawl data in small amounts.

ID	Model	All	In→in	Out→in	In→out	Out→out
(13)	Freeze LA + enc. & dec. DA	.309	.491	.362	.267	.229
(21)	**(13)** + DADrop	.311	.490	.360	.270	.233
(34)	Unfreeze LA	.357	.515	.395	.303	.307
(35)	**(34)** + BT	.372	.529	.364	.370	.319
(15)	Freeze LA + dec. DA	.332	.463	**.418**	.268	.262
(20)	**(15)** + BT	**.382**	.460	.408	.362	**.345**
(14)	Freeze LA + enc. DA	.320	.491	.345	.296	.249
(19)	**(14)** + BT	.359	.492	.374	.354	.298
(23)	Unfreeze LA + dec. DA	.322	.519	.399	.216	.267
(24)	**(23)** + DADrop	.353	.515	.399	.291	.303
(25)	**(23)** + DADrop + BT	.377	**.522**	.372	**.373**	.327
(18)	**(13)** + BT	.368	.490	.388	.363	.308
(22)	**(18)** + DADrop	.373	.494	.389	.369	.314

Table 10: chrF score of various models trained on the mostly romance language {en, fr, it, es} subset of the Koran domain. LA = language adapters, DA = domain adapters. 'Out→in' is the average score when translating from an out-of-domain source language into {en, fr, it, es}. 'In→out' corresponds to when the out-of-domain language is the target language. 'In→in' refers to average score when source and target are in the set {en, fr, it, es}. 'Out→Out' is the average score when both the source and target language are unseen during domain adaptation.

ID	Model	IT	Koran	Medical	Subtitles	Law
(46)	No fine-tuning	35.3	14.8	38.1	26.8	42.4
(47)	Fine-tuned	43.8	22.7	53	30.9	57.9
(48)	Enc. + dec. adapters ($d = 1024$)	42.9	21.8	51.7	30.5	56
(49)	**(48)** + MAD-X style	40.6	19.3	48.8	29.8	54.3
(50)	Dec. adapters ($d = 2048$)	42.1	19.8	50.5	29.7	55.1
(51)	Enc. adapters ($d = 2048$)	42.4	21.5	51.9	30.1	56.1
(52)	Last 3 encoder layers only ($d = 4096$)	42.9	21.1	52.1	30.1	56
(53)	First 3 encoder layers only ($d = 4096$)	42.2	20	50.1	28.5	54.9

Table 11: BLEU scores of various domain adaptation strategies for a German → English bilingual model. ($d = N$) refers to adapters with a bottleneck dimension of size N.

ID	Model	IT	Medical	Koran	Subtitles	Law
(54)	No fine-tuning	35.3	14.8	38.1	26.8	42.4
(55)	Finetuned	43.8	22.7	53	30.9	57.9
(56)	Enc. + dec. adapters (d=64)	40	18.7	47.3	29.4	51.5
(57)	Dec. adapters (d=128)	39	17.5	46	28.8	50.6
(58)	Enc. adapters (d=128)	40	18.9	47.3	29.2	51.5
(59)	Last 3 encoder layers only (d=256)	40	19	47.3	29	51.1
(60)	First 3 encoder layers only (d=256)	39.5	18	46	28.8	49.5

Table 12: BLEU scores of various domain adaptation strategies for a German → English bilingual model. ($d = N$) refers to adapters with a bottleneck dimension of size N.

Translation Transformers Rediscover Inherent Data Domains

Maksym Del*, **Elizaveta Korotkova***, **Mark Fishel**
Institute of Computer Science
University of Tartu, Estonia
{maksim,lisa_k,mark}@tartunlp.ai

Abstract

Many works proposed methods to improve the performance of Neural Machine Translation (NMT) models in a domain/multi-domain adaptation scenario. However, an understanding of how NMT baselines represent text domain information internally is still lacking. Here we analyze the sentence representations learned by NMT Transformers and show that these explicitly include the information on text domains, even after only seeing the input sentences without domains labels. Furthermore, we show that this internal information is enough to cluster sentences by their underlying domains without supervision. We show that NMT models produce clusters better aligned to the actual domains compared to pre-trained language models (LMs). Notably, when computed on document-level, NMT cluster-to-domain correspondence nears 100%. We use these findings together with an approach to NMT domain adaptation using automatically extracted domains. Whereas previous work relied on external LMs for text clustering, we propose re-using the NMT model as a source of unsupervised clusters. We perform an extensive experimental study comparing two approaches across two data scenarios, three language pairs, and both sentence-level and document-level clustering, showing equal or significantly superior performance compared to LMs.

1 Introduction

Neural machine translation (NMT, Bahdanau et al., 2015; Vaswani et al., 2017b) heavily depends on training data and the text domains covered in it. Full-scale NMT Transfomer models (Vaswani et al., 2017b) are usually trained on a mix of corpora from several domains (Barrault et al., 2020). However, the field lacks an understanding of how these NMT models represent the training data domains in their inner vector spaces.

This paper consists of two main parts. First, we analyze domain representations learned by the NMT Transformer. We consider sentence-level as well as document-level representations via mean pooling of token contextual embeddings. Our analysis shows that NMT models explicitly learn to include the domain information in their representational spaces across layers. Furthermore, we show that text representations preserve enough domain-specific information to reveal the underlying domains with Principal Component Analysis and k-means clustering without supervision. In the case of document-level clustering, the result of k-means matches the original corpora almost perfectly. In the case of sentence-level clustering, we observe some deviation between automatic clusters and the original corpora that the sentences belong to, showing corpus heterogeneity on the sentence level.

Aharoni and Goldberg (2020) previously revealed that a similar property exists in pre-trained language models (LMs). We compare LMs with NMT Transformers in how well we can extract unsupervised domain clusters from them and show the superiority of NMT models.

In the second part of the paper, we show how to effectively utilize our analysis to improve an existing approach to NMT domain adaptation which uses automatically extracted domains (Tars and Fishel, 2018; Currey et al., 2020). This method targets the case when training domain labels are not precise (e.g. Currey et al., 2020) or missing overall, as in case of heterogeneous corpora (e.g. Paracrawl, Esplà et al., 2019). This framework has so far been used with external models for clustering, which automatically makes us rely on clusters not necessarily aligned with the objectives of translation or target data domains.

We propose exploiting clusters extracted from the NMT baseline (already trained as a part of the framework) to improve translation quality without relying on external language models. We test our

*Equal contribution

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 599–613
November 10–11, 2021.

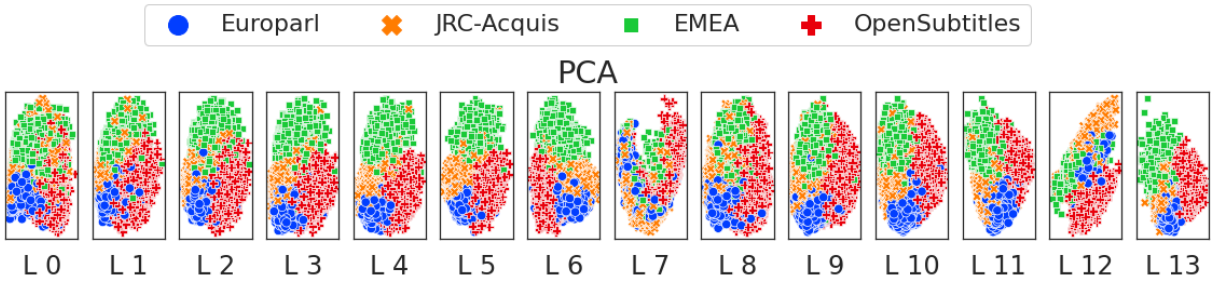


Figure 1: PCA plots of sentence representations extracted from all layers of the 60th checkpoint of the trained baseline NMT model. Representations are computed with English sentences. The dots, denoting sentences, are colored according to the domain the corresponding sentences come from. The model learns to distinguish between domains in its hidden space, despite not being explicitly provided with any information about domains. L0 corresponds to fixed encoder embeddings, L1–L6 are encoder layers' representations, L7 shows fixed decoder embeddings and L8–L13 – the decoder layers' representations. The figure shows that representations from the same domain cluster together.

proposal empirically, covering three language pairs and two data settings: a mix of corpora with known domain labels and a heterogeneous corpus without such labels. We show that fine-tuning the NMT models to the automatically discovered clusters on average matches or surpasses tuning to the original corpus labels (when available) and deep LM-based clusters.

Our contributions are thus two-fold:[1]

- we analyze the NMT encoder's representations, showing their ability to automatically discover inherent text domains and cluster unlabelled corpora, testing both sentence-level and document-level representations (Section 3);
- we utilize findings from our analysis to improve an existing Automatic Domains for NMT approach (Section 4) and perform an extensive experimental study, showing the superiority of our method (Section 5);

2 Related Work

Aharoni and Goldberg (2020) found that BERT (Devlin et al., 2019) produces meaningful unsupervised domain clusters and used this finding for NMT data selection. In this work we analyse (sentence-level and document-level) hidden representations produced by a baseline NMT model and find that it learns superior unsupervised clusters by itself.

In NMT, domain-specific information on the word level was recently analyzed by Jiang et al. (2020) in the context of domain mixing in a joint modular multi-domain NMT system. They found that representations contain domain-specific information related to the multiple domains in different proportions on the word level. We analyze representation on the sentence and document level, revealing that domain-specific information in representations converges to the one specific domain with a broader context.

Currey et al. (2020) used contextual embeddings and mean-pooled representation clustering for domain adaptation. We compare our approach to Currey et al. (2020), however in their case the representations were extracted from multilingual BERT (mBERT). We cluster based on the NMT encoder's representations directly and also experiment with document-level representations in addition to sentence-level ones.

Before Currey et al. (2020), the automatic domains framework has been used in NMT only with external models for clustering as well. Tars and Fishel (2018) used fixed embeddings from FastText (Bojanowski et al., 2017) for clustering mean-pooled sentence representations and then either tuning NMT systems to these clusters or supplying the cluster identity to the NMT system as additional input for multi-domain translation.

3 Analysis

In this section, we perform an analysis of inherent domain representations in translation transformers. We reveal how well the domain-specific information in text representations is preserved in NMT models. We focus on "out-of-the-box" NMT systems without any changes and explore the extent

[1]We release our code at https://github.com/TartuNLP/inherent-domains-wmt21

to which we can use their internal representations to match the original text domains using Principal Component Analysis (PCA) and k-means clustering. We also measure the effect of using broader document-level representations.

Additionally, we compare NMT representations to the ones extracted from a pre-trained language model, for which Aharoni and Goldberg (2020) revealed a high degree of domain-specific information.

3.1 Models and Data

In our analysis, we start by following Currey et al. (2020) and similarly to them use a multilingual LM (XLM-R, Conneau et al., 2020) to obtain clusters. XLM-R is a multilingual masked language modeling transformer covering 100 languages.

We then train *Transformer-base* (Vaswani et al., 2017a) NMT models, which have ∼97M parameters each. We train the models on parallel data covering four corpora/text domains: parliament speeches (Europarl, Koehn, 2005), medical (EMEA, Tiedemann, 2012), subtitles (OpenSubtitles, Lison and Tiedemann, 2016) and legal (JRC-Acquis, Steinberger et al., 2006). We sub-sampled the larger corpora in order to balance the size of training data across domains. The NMT models were trained for 60 epochs. A detailed description of the setup, models, and data is provided in Appendix B.

We focus on sentence-level and document-level representations, and two language pairs: English→Estonian (EN-ET) and German→English (DE-EN).

3.2 Dimensionality Reduction

We start by unsupervised dimensionality reduction using PCA to visualize domain placement. We take the development set data, extract token embeddings from each model's layer, and average them to obtain sentence representations. Then we apply cosine-based PCA and t-SNE dimensionality reduction to the representations to visualize the data in a 2D space, and post factum color each data point (sentence) according to its corresponding domain. We show the resulting visualizations in Figure 1 (best viewed in color) for ET-EN (and in Figure 4 for t-SNE in the Appendix A, which mirrors the PCA result).

Figure 1 shows that NMT partitions the domains quite well at all encoder hidden layers and deep decoder layers. Encoder layer 0 corresponds to the fixed embeddings, and the latent space is not well partitioned there yet; however, as we go deeper into the network, the separation increases. Layer 7 is the decoder's embedding layer, and there the same logic applies. While the encoder learns to partition the hidden space based on domains from scratch, the decoder has access to the encoder hidden states via encoder-decoder attention, which might simplify its task.

In summary, Figure 1 is our initial evidence that the NMT encoder places the domains separably.

3.3 Clustering

Our primary method, however, is unsupervised k-means clustering. We consider four data clustering setups: sentence-level XLM-R clusters, sentence-level NMT clusters, document-level XLM-R clusters, and document-level NMT clusters. The first one is the baseline clustering approach investigated by Aharoni and Goldberg (2020) while the remaining three are our original contributions.

3.3.1 Per-layer Clustering Purity

Metric In our analysis, we estimate how well the NMT model preserves domain-specific information in its internal text representations. To do that, we measure the goodness-of-fit between unsupervised clusters and oracle domains. Specifically, we follow Aharoni and Goldberg (2020) and use the *clustering purity* metric. To compute clustering purity, we align domains and clusters by the highest overlap in numbers of sentences. The number of overlapping data points for each cluster-domain pair gives us the number of 'correctly predicted' examples. Then, the sum of all 'correctly predicted' examples divided by the total number of examples will be the clustering purity score.

Embedding and Clustering We first take the concatenation of a small subset of sentences (3k) from each of the four domains and try to partition them into four clusters based on the representations from each layer of XLM-R and NMT Transformer. We only use source sentences since we do not have targets at runtime in NMT. Specifically, we follow the steps below for each layer of each of the two models:

1. For each sentence in the dataset, we extract contextualized token embeddings from a layer of the model.
2. We use the average of contextualized token embeddings as sentence representations.

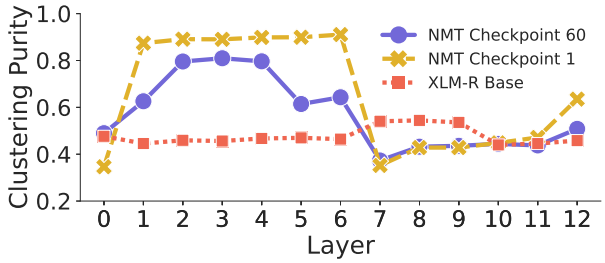


Figure 2: Sentence-level clustering purity between clusters obtained with k-means over 3k of EN-ET development set sentences and actual data domains. Representations extracted from XLM-R, NMT Baseline after epoch 1, and NMT Baseline after training has finished (epoch 60). While XLM-R is relatively poor in its ability to rediscover original domains, representations extracted from the trained NMT model largely outperform it at layers 1-6. Layers 1-6 are the encoder and 7-12 the decoder. The results are for the best clustering (with least variance) over 10 k-means runs.

3. We apply k-means clustering to sentence representations to assign a cluster label to each sentence.

4. We compute clustering purity for predicted labels and oracle domains.

We perform ten random restarts of k-means clustering, selecting the iteration with the smallest within-cluster variance.

Results Figure 2 shows per-layer clustering purity computed for sentence representations for XLM-R and two NMT Baseline checkpoints (after the first epoch and after the 60th epoch of training). Figure 2 shows that NMT surpasses the language model in its ability to rediscover domains. About 3.5x higher performance at the encoder layers shows that the encoder is the part that learned to be very aware of the input domains (in an unsupervised way). Figure 2 also shows that the checkpoint saved after the 1st training epoch rediscovers clusters slightly better then 60th checkpoint. However, this does not suggest that an NMT model should be trained for one epoch since the translation quality is suboptimal early on. Instead, we assume that the model quickly learns domain-specific information (perhaps due to the common lexical statistics) and then slightly "moves away" towards a higher level of abstraction as training progresses. This abstraction is necessary to successfully learn a task as complex as NMT.

	EN-ET			DE-EN		
	train	dev	test	train	dev	test
sentence						
XLM-R	53.47	52.9	50.07	44.04	49.2	48.6
NMT	67.21	72.56	70.7	66.32	70.02	72.28
document						
XLM-R	85.77	72.89	70.14	97.64	91.74	95.23
NMT	99.61	100.0	99.1	99.21	97.58	99.78

Table 1: Clustering purity. We trained the NMT model on about 2m EN-ET or DE-EN sentences from multiple corpora and used pre-trained XLM-R Base model. Based on the Figure 2, we used the 4th layer to extract source representations from the NMT model and 8th layer for XLM-R. The results are for the best clustering (with least variance) over 10 k-means runs. Both NMT and XLM-R rediscover inherent data domains when document level representations are used, and seem to produce more customized separations when clustered based on sentence-level. NMT tends to be better at rediscovery.

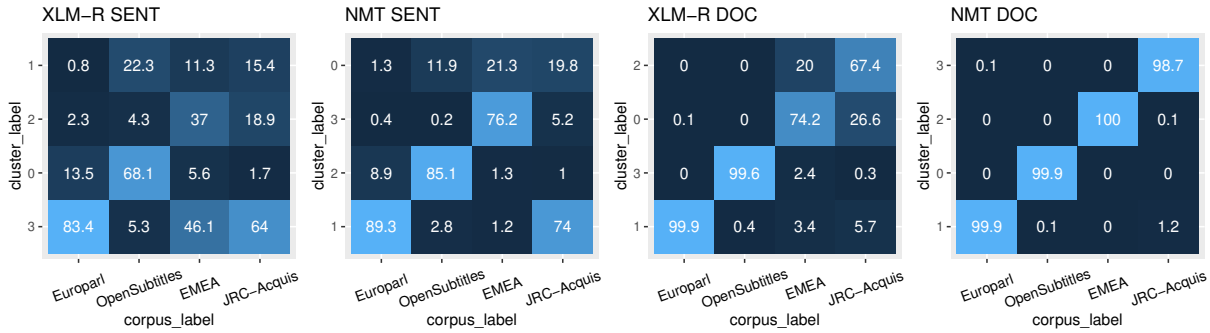
3.3.2 Large-scale Clustering

Next, we repeat the same steps for the entire training dataset and include a second language pair. Specifically, we pick one of the best performing layers (4th for NMT and 7th for XLM-R) based on the experiment above and use it to cluster the training part of the multi-domain machine translation training set (about 500k examples per domain, 2M in total). We then predict cluster labels for the training examples and use the same model to cluster unseen examples from the development and test set.

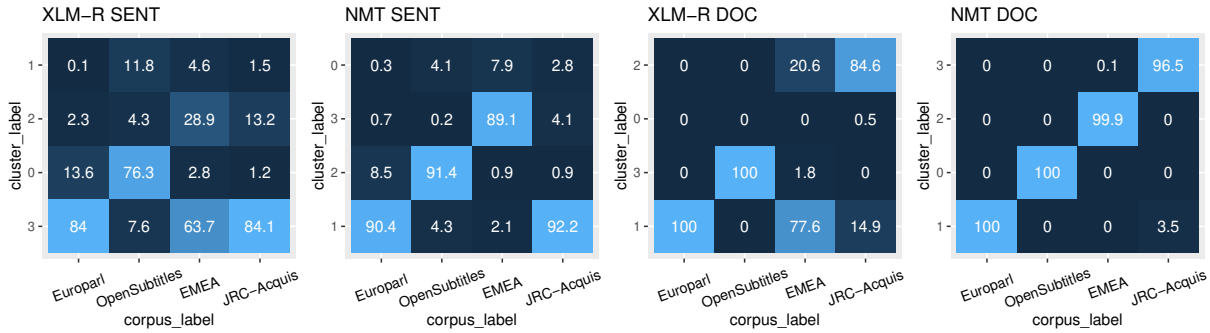
We also extend our analysis to the document-level scenario. Specifically, we average over sentence representations to get document embeddings and cluster-based on them. Then, we assign the predicted label for each document to each sentence in that document.[2]

Results We present large-scale clustering confusion matrices in Figure 3 and clustering purity in Table 1. These show that sentence-level NMT is generally better than sentence-level XLM-R at rediscovering domains. However, they both show quite modest results for both language pairs. At the same time, document-level clusters are much better at rediscovering original domains.

[2]Sentence pairs coming from the same XML file were considered to belong to the same document. The training, development and test sets in all experiments were constructed in such a way that a document is always included in one set in its entirety.



(a) EN-ET Train



(b) EN-ET Test

Figure 3: Corpus-cluster confusion tables for about 2M sentences for EN-ET for (a) the training and (b) test sets. The numbers are percentages for each domain (column). NMT document clusters almost perfectly match original data domains. On the sentence level, however, both NMT and XLMR learn a more customized notion of clusters, with NMT being more aligned with original domains.

The reason for that might be that sentence-level clustering largely relies on the more shallow information in the text. For example, we observed that both sentence-level NMT and XLM-R produced a cluster responsible for extremely short sentences (the average sentence length is about four tokens for these clusters). On the other hand, document-level representations factor out these shallow stylistic features by averaging over sentence representations. Therefore, the models are inclined to cluster by topics. An alternative explanation is that domain-specific lexical statistics, which not all sentences might preserve, get more robust as we average sentence embeddings to get a document embedding.

Even though sentence-level clustering maintains a general idea about oracle domains, they split sentences into clusters quite freely. For example, JRC-Acquis consistently gets mixed with Europarl, which both belong to legal domains. We can see it from *NMT SENT* for both language pairs.

For documents, the rediscovery trend is common and pronounced for both language pairs, and separation is generally consistent between train and test. However, for *EN-ET XLM-R DOC* we can observe that EMEA and JRC-Acquis got split between two clusters in the training set. Considering that we perform ten random k-means restarts and choose the best iteration, this suggests that XLM-R may become inconsistent (as a source of sentence representations on the document level) in some cases.

Figure 5 in Appendix A shows similar heatmaps for DE-EN. DE-EN is consistent with what we observe for EN-ET apart from XLM-R DOC, where the DE-EN diagonal is cleaner.

4 Practical Application

Our analysis in Section 3 revealed that NMT models represent domains in their embedding space separately, similarly to what pre-trained language models do (Aharoni and Goldberg, 2020). We demonstrated that simple clustering on NMT representations allows recovering original data domains to a large degree.

This section proposes to utilize this finding to improve an existing framework of automatic domain generation for NMT. In this framework, related work first clusters the training data using representations from an external encoder, and then the base-

line NMT model is adapted (fine-tuned) on each cluster separately. We propose to re-use the NMT baseline itself as the encoder in this framework.

Representations extracted from translation Transformers are specific to the task of translation. We hypothesise that it might result in clusters most suitable for downstream translation tasks like fine-tuning to specific domains/clusters.

Moreover, an advantage of our scenario is that we cluster the same data (with our NMT model) that we use for NMT model training. It is a frequent multi-domain NMT setup, where multiple target domains are available in training. In the pre-trained language model setup, the data will be more out-of-domain, despite the model's generality.

4.1 Existing Framework (Background)

In this subsection we describe an existing framework which uses automatic domains (clusters) to perform NMT domain adaptation (Tars and Fishel, 2018). Recent work (Currey et al., 2020) employs large pretrained language models as part of the framework. It consists of several steps.

In step 1.1, we begin with a single heterogeneous dataset ("Original Dataset") and train a baseline NMT model on it. At the same time (step 1.2), we pass this dataset through the external pre-trained XLM-R model to extract hidden sentence/document representations for the whole dataset. In step 2, we use the extracted sentence/document representations to train a k-means clustering model. In step 3, we use this k-means model to separate the original dataset into sub-datasets corresponding to the clusters. Lastly, we use the cluster-specific datasets to fine-tune the baseline NMT model from step 1.1 on each dataset separately, resulting in a set of specialized models. We use the k-means model at runtime to determine which NMT model to use to translate a new sentence/document. If we only use sentence clusters, the approach is equivalent to the one proposed by Currey et al. (2020). Refer to Figure 6 from Appendix A for the illustration of the steps described above.

4.2 Improved Framework (Ours)

In this subsection we describe our modification to the existing automatic domains pipeline presented in Section 4.1.

We propose reusing an NMT baseline to produce sentence representations for the clustering step instead of using an external encoder. Specifically, in step 1, we train a baseline NMT model just like in the existing framework. However, we found we can omit using the XLM-R model (step 1.2). Instead, to extract sentence/document representations for step 2, we reuse the trained NMT baseline. The rest of the pipeline remains the same. Figure 7 (Appendix A) illustrates the updated framework.

Moreover, to produce clusters in both frameworks, we additionally study text representations on the level of documents.

5 Experiments

In this section we perform an extensive experimental study comparing performance of the existing automatic domains framework (Section 4.1) with our proposed version (Section 4.2). We experiment with both sentence-level and document-level representations as a basis for k-means algorithm on three language pairs and two data scenarios.

We first train baseline Transformer NMT models on concatenated data from all domains (same baseline as in Section 3) and then cluster the training, development, and test data using either this same baseline or XLM-R. Next, we fine-tune[3] our baseline models to the different obtained data partitions (clusters) and compare the translation quality of resulting fine-tuned (adapted) models.

5.1 Setup

We explore two data scenarios. First, we perform experiments on a mixture of distinct corpora. For these experiments, we reuse the data and *concat* baseline NMT model (Transformer-base) described in Section 3 (EN-ET and DE-EN). In this setting, we can compare the performance of models fine-tuned to automatically discovered domains to that of oracle models (fine-tuned using known domains/datasets). We also randomly partition the data (into equal parts) and fine-tune the baseline models to them to get our lower bound estimates.

Second, we explore a scenario with a single corpus, which is highly heterogenous, and thus may contain multiple domains which are unknown. In this setting, we use the ParaCrawl (Esplà et al., 2019) parallel corpus[4], which consists of diverse documents crawled from the web. We use three language pairs: English→Estonian (EN-ET), German→English (DE-EN), and English→Czech

[3] We use the terms "fine-tuning" and "NMT domain adaptation" interchangeably.

[4] https://paracrawl.eu/

(EN-CS). We use ∼3M sentence pairs for all languages for training, and ∼3,000 sentence-pairs for development and testing. The exact experimental setup with data sizes, training and preprocessing details can be found in in Appendix B.

For our *concat* baselines we follow the setup from Section 3.1 (described in more detail in Appendix B). In the mixture of corpora experiments, baseline fine-tuning is performed for 50 epochs, and in the single heterogenous corpus experiments for 25 epochs (fine-tuning hyperparameters can be found in Appendix B). For comparison, we also continue training the baseline models for longer as suggested by Gururangan et al. (2020) (*concat-cont*). We continue training for the same number of epochs fine-tuning is done for in the corresponding experiment.

For each of the models, we evaluate the checkpoint which shows the highest BLEU score on the particular model's development set, and translate the test sets with beam size set to 5. We use the BLEU score (Papineni et al., 2002), specifically, the sacreBLEU implementation (Post, 2018) to assess the models' translation performance. To test for statistical significance, we use paired bootstrap resampling (Koehn, 2004).

5.2 Labelled Domain Mix Experiments

In this section we consider a scenario which can be practically interesting in cases where the data consists of several distinct domains with the labels unavailable or corrupted as in Currey et al. (2020). Moreover, it serves as an oracle experiment showing how well automatic domains perform compared to the golden labels. This way we have a better idea what to expect when applying them to unlabeled data as in Section 5.3.

Table 2 shows the results for DE→EN. We see that, for all corpora except Europarl, at least one model of the two that are based on document-level clustering always manages to surpass the oracle performance obtained by fine-tuning to known domains, and on Europarl the document-level models perform comparably to oracle. In most cases, document-level models show significantly better translation quality than XLM-R sentence-level models, which have been used in previous work, while NMT sentence-level models closely match the performance of XLM-R sentence ones. When scores are averaged over all four domains, document clustering obtained from the NMT encoder is the overall winner.

	EP	OS	JRC	EMEA	avg
concat	37.2	21.7	52.3	73.8	46.25
concat-cont	37.2	22.3	52.4	73.7	46.40
oracle	**37.4**	22.6	53.4	74.7	47.03
sentence					
XLM-R	36.6	22.4	52.8	73.9	46.43
NMT	36.6	22.3	52.8	74.0	46.43
document					
XLM-R	37.3**	**22.9***	53.0	75.0**	47.05
NMT	37.3**	22.5	**53.7****	**75.4****	**47.23**
random	36.8	22.4	51.7	73.3	46.05

Table 2: BLEU scores of the DE-EN baseline models, models fine-tuned to known corpora (oracle), to the proposed automatic domains, and to a random partitioning of the data. EP, JRC, EMEA and OS stand for Europarl, JRC-Acquis, EMEA and OpenSubtitles test sets, respectively. Statistically significant improvements of our proposed methods over sentence-level XLM-R clustering are marked with * ($p \leq 0.05$) or ** ($p \leq 0.01$). Document-level clustering matches and slightly surpasses the performance of fine-tuning on oracle domains.

Table 3 shows results for the EN→ET language pair. While fine-tuning on oracle domains yields an average improvement of 0.8 BLEU points over the baseline, fine-tuning on unsupervised document clusters obtained from the NMT encoder allows us to match that performance. However, for the EMEA test set XLM-R sentence clusters turn out to be the most successful approach, showing significantly higher BLEU scores than all other automatic partitions and outperforming the oracle by 1.2 BLEU points, while document-level NMT clustering also manages to surpass the oracle performance, albeit slightly. For OpenSubtitles and JRC-Acquis, oracle shows the highest overall scores, with document-level NMT clustering a close second, outperforming XLM-R sentence clustering by a noticeable margin. For OpenSubtitles, however, none of the automatic domain approaches manage to improve the baseline performance (and neither does continued training of the baseline), and even the oracle partition does not manage to do so by a statistically significant degree. For Europarl, all automatic domain approaches yield comparable BLEU scores, with none being significantly better or worse than XLM-R sentence clusters.

Document-level XLM-R automatic domains have a low average score due to underperforming on the EMEA test set. We see from Figure 3 that this is a case of train-test mismatch: the EMEA

	EP	OS	JRC	EMEA	avg
concat	28.7	19.1	47.3	47.8	35.725
concat-cont	28.8	18.5	48.4	48.5	36.050
oracle	28.7	**19.2**	**50.0**	48.2	36.525
sentence					
XLM-R	29.0	18.6	48.9	**49.4**	36.475
NMT	29.1	18.7	49.0	48.1††	36.225
document					
XLM-R	**29.2**	18.6	47.9††	39.2††	33.725
NMT	29.1	19.0*	49.8**	48.4††	**36.575**
random	28.5	18.5	47.1	47.0	35.275

Table 3: BLEU scores of the EN-ET baseline models, models fine-tuned to known corpora (oracle), to the proposed automatic domains, and to a random partitioning of the data. EP, JRC, EMEA and OS stand for Europarl, JRC-Acquis, EMEA and OpenSubtitles test sets, respectively. Statistically significant improvements of our proposed methods over sentence-level XLM-R clustering are marked with * ($p \leq 0.05$) or ** ($p \leq 0.01$), daggers mark results which are significantly lower than for sentence-level clustering based on XLM-R († and †† denote $p \leq 0.05$ and $p \leq 0.01$, respectively). Document-level clustering as well as XLM-R based sentence-level clustering match the performance of the fine-tuning on oracle domains.

test set is mostly translated by the model fine-tuned on cluster 1, whose training set predominantly consists of Europarl data. Cluster 0, which sees the most EMEA examples during fine-tuning, is not used to translate the test set at all, as we see from Figure 3.

5.3 Heterogeneous Corpus Experiments

In this subsection we present results for our method applied to the Paracrawl dataset, which constitutes a heterogeneous corpus of data crawled from the web with no training-time domain information known.

EN-ET We first experiment on the EN-ET language pair. While in the multi-corpus setup we chose the number of clusters to match the number of different corpora in our training set, in the ParaCrawl experiments we do not have a predefined number of domains. Therefore, we experiment with separating the dataset into 3, 4, 5, and 8 clusters.

The resulting BLEU scores for EN-ET are shown in Table 4. Fine-tuning based on NMT and XLM-R clustering of the data outperforms a strong *concat-cont* baseline by 0.2-1.6 BLEU points depending on the choice of embedding model and clustering

N of clusters	3	4	5	8
concat	46.1	46.1	46.1	46.1
concat-cont	46.6	46.6	46.6	46.6
sentence				
XLM-R	**47.0**	46.8	47.1	47.0
NMT	46.9	47.1	**47.6****	47.4*
document				
XLM-R	46.8	**47.2***	47.3	47.6**
NMT	46.8	47.0	47.2	**48.2****
random	46.1	45.9	45.5	45.3

Table 4: BLEU scores of models trained on EN-ET ParaCrawl and fine-tuned to different numbers of automatic clusters and to a random partitioning of the data. Statistically significant improvements of our proposed methods over sentence-level XLM-R clustering are marked with * ($p \leq 0.05$) or ** ($p \leq 0.01$). For different numbers of clusters different approaches score best, but the best result overall is obtained with document-level NMT and 8 clusters.

level. The best result overall is achieved by our document-level NMT clustering, which also outperforms all other approaches with 8 clusters by at least 0.6 BLEU. Both document-level approaches improve their performance with a growing number of clusters. With 3 clusters, all clustering methods show comparable results, with none being significantly better or worse than sentence-level XLM-R. Document-level XLM-R and sentence-level NMT significantly outperform sentence-level XLM-R with 4 and 5 clusters, respectively.

EN-CS & DE-EN As separating the data into 8 clusters yields the highest BLEU score among all fine-tuning scenarios for EN→ET, we choose this number of clusters for experiments on other language pairs. Table 5 shows the BLEU scores for EN→ET, EN→CS, and DE→EN models fine-tuned to automatic domains.

For EN-CS, only the NMT sentence-level clustering manages to outperform the baseline, noticeably surpassing all other automatic domain extraction methods as well.

For DE-EN, none of the approaches outperform the baseline model by a considerable margin. Sentence-level clustering based on XLM-R performs comparably to the baseline. Document-level NMT clustering shows a slightly lower score, but the difference is not statistically significant. At the same time, document XLM-R and sentence NMT perform worse than sentence XLM-R.

	EN-ET	EN-CS	DE-EN
concat	46.1	44.4	48.2
concat-cont	46.6	44.3	48.1
sentence			
XLM-R	47.0	44.2	**48.3**
NMT	47.4*	**44.9****	48.0†
document			
XLM-R	47.6**	44.2	47.9††
NMT	**48.2****	44.2	48.0
random	45.3	43.6	47.5

Table 5: BLEU scores of models trained on ParaCrawl and fine-tuned to automatic clusters and to a random partitioning of the data on three language pairs. For each language pair we use ~3M training examples and split the data into 8 clusters. Statistically significant improvements of our proposed methods over sentence-level XLM-R clustering are marked with * ($p \leq 0.05$) or ** ($p \leq 0.01$), Daggers mark results which are significantly lower than for sentence-level clustering based on XLM-R († and †† denote $p \leq 0.05$ and $p \leq 0.01$, respectively).

N of clusters	4	12
concat	50.6	50.6
concat-cont	50.9	50.9
sentence		
XLM-R	50.9	50.6
NMT	51.0	**50.9***
document		
XLM-R	**51.1**	50.8
NMT	**51.1**	50.8
random	50.2	49.8

Table 6: BLEU scores of models trained on 10M sentence pairs from DE-EN ParaCrawl and fine-tuned to 4 and 12 automatic clusters. The data size is increased compared to the previous experiments, the NMT model size remains the same. We see an improvement in baseline performance, but no improvement in the performance of fine-tuned models. Statistically significant improvements of our proposed methods over sentence-level XLM-R clustering are marked with * ($p \leq 0.05$).

5.4 Additional Exploration

While automatic domains demonstrate reasonable performance for EN-ET and EN-CS language pairs, DE-EN does not seem to benefit from either XLM-R or NMT-based clustering. In this section we perform additional experiments with DE-EN data to see whether there are conditions under which automatic domains could be beneficial in this case.

Data Size and Number of Clusters First, we increase the training data size and vary the number of clusters. Specifically, we use 10M parallel sentence pairs for training instead of 3M, and partition the dataset into 4 and 12 clusters instead of 8.

The resulting BLEU scores for DE-EN are shown in Table 6. We do not observe any significant improvement over the *concat-cont* baseline for any of the methods. With the data separated into 12 clusters, sentence-level NMT clustering significantly outperforms sentence-level XLM-R, but still does not beat continued training of the baseline.

Model Size It is also possible that NMT needs different model capacity for handling different language pairs, so we experiment with decreasing the model size. We use the same number of layers, but decrease the width of the model (4 attention heads, embeddings of size 160, dimension of the feed-forward layer 320) so that the total number of parameters decreases five-fold. We compute NMT clusters based on the new, smaller baseline model. Our motivation for this is to understand whether automatic domains are not useful for DE-EN ParaCrawl at all, or could aid a weaker baseline.

The results are shown in Table 7. The smaller baseline does benefit from adaptation to automatic domains (clusters). While NMT clusters are generated by a model which is 5 times as small, XLM-R and NMT show equivalent performance.

6 Discussion

Our analysis is implicit inductive evidence for the high degrees of domain-specific information in sentence and document NMT representations. However, it is still open to what kind of information is preserved (topical/stylistic/lexical).

For example, our approach could result in clusters by domain/dataset due to standard lexical statistics and not sentence semantics. However, on the practical side, we show that adapting NMT to these types of clusters is just as good or better as to other possible types of clusters since it benefits the baseline performance. Moreover, previous work that uses pre-trained language models to obtain the clusters is likely to suffer from the same issue.

Moreover, while XLM-R is a general-purpose encoder, NMT models are only that helpful for domains we train them on. However, the data constitutes all domains of interest by definition for a multi-domain NMT (the task we tackle). Thus, NMT models are a perfect fit that simplifies and outperforms an existing approach.

Model size	Base	Small
concat	48.2	44.2
concat-cont	48.1	44.8
sentence		
XLM-R	**48.3**	45.2
NMT	48.0†	**45.4**
document		
XLM-R	47.9††	45.1
NMT	48.0	**45.4**
random	47.5	44.0

Table 7: BLEU scores of models trained on 3M sentence pairs from DE-EN ParaCrawl and fine-tuned to 8 automatic clusters. The Base NMT model has the same configuration as in previous experiments (*Transformer-base*), while the Small model has 5 times fewer parameters. The smaller model benefits from fine-tuning to automatic domains, but does so starting from a weaker baseline performance. Daggers mark results which are significantly lower compared to sentence-level clustering based on XLM-R († and †† denote $p \leq 0.05$ and $p \leq 0.01$, respectively).

7 Conclusion

In this work, we made a two-fold contribution. The first is to the field of NMT interpretation and analysis. We have shown that a baseline Transformer NMT encoder preserves enough domain-specific information to distinguish between oracle domains in a mixed corpus without supervision. We showed an evolution of this property across the Transformer layer using PCA and k-means clustering on the level of sentences and documents. Comparison to XLM-R based clusters demonstrated that both sentence-level and document-level NMT clusters show higher cluster purity (similarity to original text domains).

Next, we utilized our analysis insights to improve an existing practical cluster-based multi-domain NMT approach (Tars and Fishel, 2018; Currey et al., 2020). In a setting with preset domains (i.e., available corpus/domain labels), tuning to NMT clusters on average matches or surpasses XLM-R clusters. Additionally, NMT cluster-based tuning mostly matches the translation quality when tuning to original corpus labels, with some exceptions that we also analyze and explain.

Finally, in the case of a heterogeneous corpus (ParaCrawl), the performance of fine-tuned NMT models depends on the number of clusters, language pairs, and other parameters. We see significant improvement for EN-ET and EN-CS translation when comparing XLM-R and NMT-based clusters (on both sentence and document levels). For DE-EN, the domain tuning results depend on the NMT model's capacity for learning each language pair's translation.

Acknowledgements

This work has been supported by the grant No. 825303 (Bergamot[5]) of European Union's Horizon 2020 research and innovation program. The authors also thank the Unversity of Tartu's High-Performance Computing Center for providing GPU computing resources (University of Tartu, 2018).

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

[5] https://browser.mt/

Anna Currey, Prashant Mathur, and Georgiana Dinu. 2020. Distilling multiple domains for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4500–4511, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1834, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. pages 388–395.

Philipp Koehn. 2005. Europarl : A Parallel Corpus for Statistical Machine Translation. *MT Summit*, 11.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: 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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. In *Proceedings of EAMT*, pages 259–268, Alicante, Spain.

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).

University of Tartu. 2018. Ut rocket cluster, https://doi.org/10.23673/ph6n-0144.

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

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

A Analysis

A.1 Additional Figures

Figures 4 and 5 support our analysis in Section 3 while Figures 6 and 7 illustrate frameworks from Section 4.

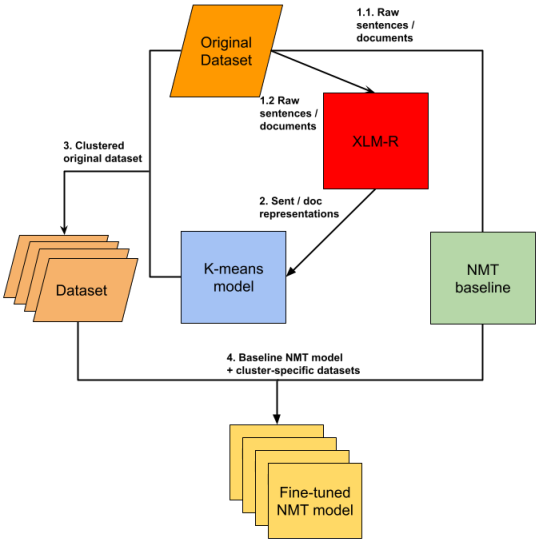


Figure 6: Existing automatic domains framework (previous approach).

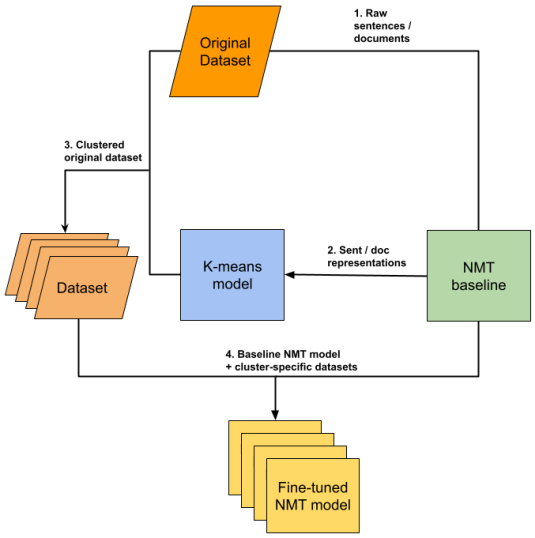


Figure 7: Updated automatic domains framework (ours).

A.2 Language Model

XLM-R Base Our model of choice from the family of BERT-like models is the Base version of the XLM-R (Conneau et al., 2020). It is a single multilingual model covering about 100 languages, which is very useful when dealing with machine translation systems, where for different language pairs we may not have a separate monolingual BERT for each source language. We choose XLM-R as opposed to the multilingual BERT (Devlin et al., 2019) since it is a more recent and better performing (Hu et al., 2020) model. We choose the Base version because it is most compatible to our NMT baseline in terms of capacity.

B Experiments Setup

B.1 Data

For multi-domain fine-tuning (Section 5.2) we experiment on German→English (DE-EN) and English→Estonian (ET-EN), and for the heterogenous corpus task (Section 5.3) we also evaluate on English→Czech (EN-CS).

We use Europarl (proceedings of the European Parliament) (Koehn, 2005), JRC-Acquis (legal documents of the European Union) (Steinberger et al., 2006), EMEA (documents of the European Medicines Agency) (Tiedemann, 2012) and OpenSubtitles (movie and TV subtitles) (Lison and Tiedemann, 2016)[6] in the multidomain fine-tuning experiments. Data from the four corpora was approximately balanced. Around 500,000 training sentence pairs were taken from each of the corpora (except for EN-ET EMEA, where only 400,000 sentence pairs were available after cleaning), making the total size of the training set 1.9M sentence pairs for EN-ET and 2M for DE-EN. Development and test sets contain at least 3,000 sentences per corpus. The exact sizes of training, development and test sets can be found in Table 9. We test sentence-level and document-level clustering of the texts. For Europarl, JRC-Acquis, EMEA and OpenSubtitles, sentence pairs coming from the same XML file were considered to belong to the same document. The training, development and test sets in all experiments were constructed in such a way that a document is always included in one set in its entirety (hence the irregular sizes of the train, development and test sets).

For the single heterogeneous corpus experiments we use v.7.1 of publicly available[7] Paracrawl dataset for all three language pairs. The training set sizes are 3M for all sentence pairs unless otherwise noted. Development and test sets contain at least 3,000 sentences per corpus in all experiments. The exact sizes of training, development and test sets in each of the experiments can be found in Tables 8, 9,

[6] https://opus.nlpl.eu/
[7] https://paracrawl.eu/

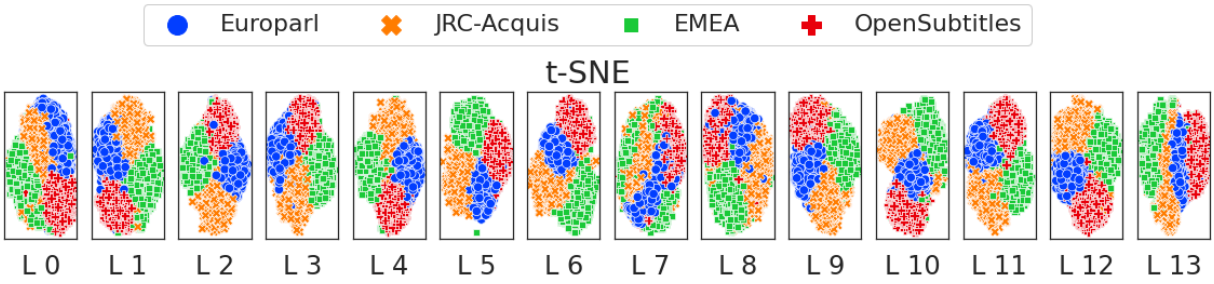


Figure 4: t-SNE plots of sentence representations extracted from all layers of the 60th checkpoint of the trained baseline NMT model. The dots, denoting sentences, are colored according to the domain the corresponding sentences come from. The model learns to distinguish between domains in its hidden space, despite not being explicitly provided with any information about domains. The figure shows that representations from the same domain cluster together.

and 10. In the ParaCrawl experiments, documents were matched by the source sentence URLs. The dataset is separated into webpage-based documents which we use to compile document-based clusters of representations produced by the baseline NMT and XLM-R models. The training, development and test sets in all experiments were constructed in such a way that a document is always included in one set in its entirety.

Several basic cleaning steps were applied to the corpora. Sentence pairs were discarded if:

- either the source or the target side was an empty string;
- either the source or the target side contained more than 100 tokens;
- one of the sentences in the pair contained at least 9 times as many tokens as the other;
- more than half of the characters in either the source or the target sentence were non-alphabetic characters (noisy source or target)

In some corpora there are many sentence pairs that occur multiple times. Therefore, to avoid unfairly inflating the test scores, sentence pairs that also occur in the training set were removed from the development and test sets for BLEU score calculation in the multi-corpus experiments.

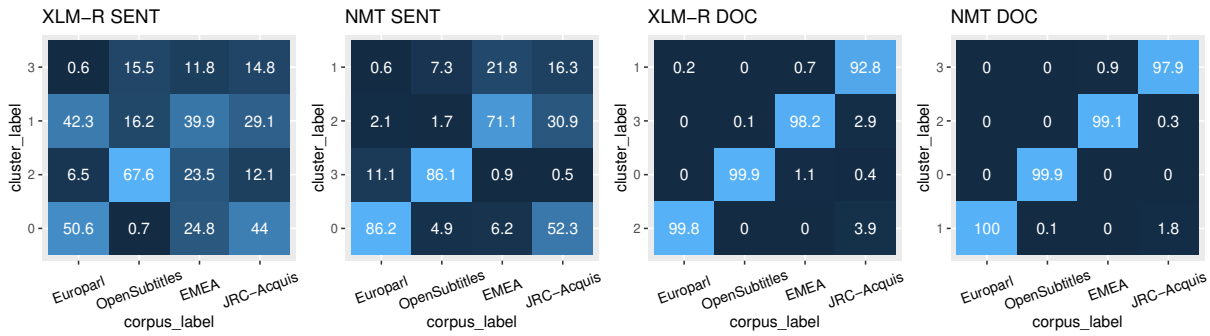
The data was split into subwords using SentencePiece (Kudo and Richardson, 2018) with vocabulary size set to 32,000. No other pre-processing steps were applied.

B.2 NMT Training

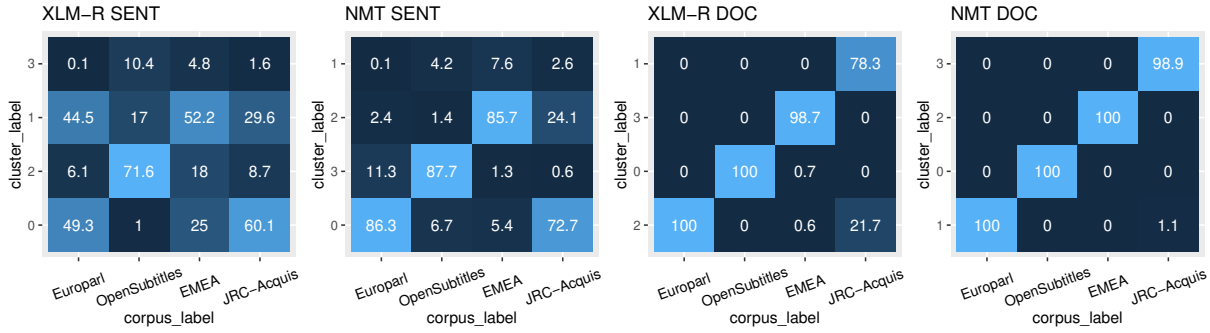
We train Transformer machine translation models using the Fairseq toolkit (Ott et al., 2019). The models have a standard configuration, mostly following the *Transformer-base* settings (Vaswani et al., 2017a): 6 encoder and 6 decoder layers, embedding dimension 512, feed-forward layer dimension 2048. The initial learning rate was set to 5×10^{-4}, with inverse square root learning rate scheduler with 4,000 warm-up updates. The loss function is label-smoothed cross entropy with label smoothing α equal to 0.1. We use Adam optimizer, with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. Dropout probability is set to 0.3. The source and target vocabularies are shared. Model checkpoints are saved at the end of each epoch.

When fine-tuning, we pre-train the model without any explicit domain specific information, and then initialize each model with the parameters of the baseline's checkpoint from the 60th epoch. In the mixture of corpora experiments, fine-tuning is performed for 50 epochs, and in the single heterogenous corpus experiments for 25 epochs (our experiments show that for the overwhelming majority of models the checkpoint which has the best BLEU score on the development set occurs before 25 epochs of fine-tuning). Fine-tuning was performed with initial learning rate 1.25×10^{-4}, reducing by a factor of 0.5 every time the development loss has not improved for 3 consecutive epochs. For comparison, we also continue training the baseline model for the same number of epochs fine-tuning is done for. For each of the models, the translation is done with the checkpoint which has the highest BLEU score on the particular model's development set.

We use the BLEU score (Papineni et al., 2002), specifically, the sacreBLEU implementation (Post, 2018) to assess the models' translation performance. To test for statistical significance, we use



(a) DE-EN Train



(b) DE-EN Test

Figure 5: Corpus-cluster confusion tables for about 2M sentences for DE-EN for the training (a) and test (b) sets. Document clusters rediscover original domains and NMT while sentence clusters tend to learn more customized notion of clusters. In general, NMT is more aligned to the oracle domains then BERT.

	Europarl	JRC-Acquis	OpenSubtitles	EMEA	total
train	500,697 (910)	500,207 (8,874)	501,510 (620)	500,070 (941)	2,002,484 (11,345)
dev	3,566 (2)	3,106 (78)	4,306 (6)	3,406 (10)	14,384 (96)
test	3,265 (12)	3,008 (65)	3,063 (3)	5,908 (18)	15,244 (98)

Table 8: Number of sentence pairs (and documents) from each corpus (Europarl, JRC-Acquis, OpenSubtitles, EMEA) in the training, development and test sets of the DE-EN model trained on a mixture of known corpora

	Europarl	JRC-Acquis	OpenSubtitles	EMEA	total
train	500,166 (1,979)	500,020 (8,877)	500,876 (563)	410,540 (732)	1,911,602 (12,151)
dev	3,716 (7)	3,005 (91)	3,044 (3)	3,348 (10)	13,113 (111)
test	3,107 (16)	3,190 (91)	3,085 (4)	3,315 (12)	12,697 (123)

Table 9: Number of sentence pairs (and documents) from each corpus (Europarl, JRC-Acquis, OpenSubtitles, EMEA) in the training, development and test sets of the EN-ET model trained on a mixture of known corpora

	EN-ET	EN-CS	DE-EN 3M	DE-EN 10M
train	3,163,124 (366,120)	3,000,000 (777,448)	3,000,013 (546,015)	10,000,000 (1,819,571)
dev	3,064 (400)	3,019 (737)	3,018 (618)	3,018 (618)
test	3,130 (300)	3,011 (770)	3,007 (563)	3,007 (563)

Table 10: Number of sentence pairs (and documents) in the training, development and test sets of the EN-ET, EN-CS, and DE-EN models trained on data from one heterogenous corpus (ParaCrawl)

paired bootstrap resampling (Koehn, 2004).

The models were pre-trained and fine-tuned either on one NVIDIA V100 GPU with 32GB of RAM with maximum batch size 15,000 tokens per node or on two NVIDIA V100 GPU's with 16GB of RAM with maximum batch size 7,500 tokens per node. The only exception is the DE-EN ParaCrawl model with 10M training sentence pairs, which has the largest volume of training data, and was pre-trained on 4 NVIDIA V100 GPU's with 32GB of RAM with maximum batch size 15,000 tokens per node.

Improving Machine Translation of Rare and Unseen Word Senses

[1]Viktor Hangya, [2]Qianchu Liu, [1]Dario Stojanovski,
[1]Alexander Fraser and [2]Anna Korhonen
[1]Center for Information and Language Processing, LMU Munich
{hangyav,stojanovski,fraser}@cis.lmu.de
[2]Language Technology Lab, TAL, University of Cambridge, UK
{ql261,alk23}@cam.ac.uk

Abstract

The performance of NMT systems has improved drastically in the past few years but the translation of multi-sense words still poses a challenge. Since word senses are not represented uniformly in the parallel corpora used for training, there is an excessive use of the most frequent sense in MT output. In this work, we propose CMBT (**C**ontextually-**m**ined **B**ack-**T**ranslation), an approach for improving multi-sense word translation leveraging pre-trained cross-lingual contextual word representations (CCWRs). Because of their contextual sensitivity and their large pre-training data, CCWRs can easily capture word senses that are missing or very rare in parallel corpora used to train MT. Specifically, CMBT applies bilingual lexicon induction on CCWRs to mine sense-specific target sentences from a monolingual dataset, and then back-translates these sentences to generate a pseudo parallel corpus as additional training data for an MT system. We test the translation quality of ambiguous words on the MuCoW test suite, which was built to test the word sense disambiguation effectiveness of MT systems. We show that our system improves on the translation of difficult unseen and low frequency word senses.

1 Introduction

Recent NMT systems have remarkable performance for many languages (Vaswani et al., 2017; Xia et al., 2019) but there are still numerous areas for improvement. One such important area concerns the disambiguation and translation of multi-sense words. It is particularly challenging to MT systems as sense distribution is skewed with some senses rarely seen or missing in the parallel corpora. This results in the MT system producing translation errors for these rare/unseen senses, causing incomprehensible output sentences.

In this work we aim at improving the translation of rare and unseen senses of ambiguous words.

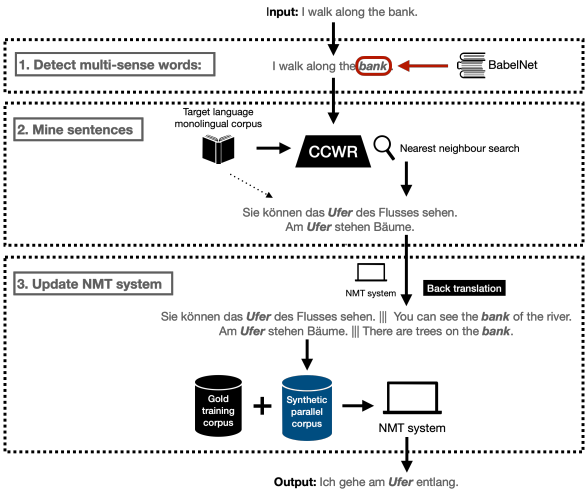


Figure 1: The pipeline of CMBT. Note that each step of the pipeline is run on the full corpus (source side of MuCoW test set). Here we just show the procedure on a single sentence as an illustration.

Previously Tang et al. (2018, 2020) showed that encoder-decoder based NMT systems integrate information relevant for WSD into the encoder hidden states, but finding the correct sense is still a challenging task and NMT systems are biased toward the most frequent senses of words (Liu et al., 2018). Disambiguation errors are often due to over-reliance on training data artifacts, such as frequent word co-occurrences (e.g. *hot spring* is always translated as the thermal activity and not as a season), instead of a deeper understanding of the multi-sense words given the input sentences (Emelin et al., 2020). Additionally, MT systems tend to learn and use frequent words more often and disregard less frequent ones (Vanmassenhove et al., 2019). Previous work has improved the translation of ambiguous words, e.g., by leveraging lexical resources (Pu et al., 2018) or sense specific embeddings (Liu et al., 2018), but they are restricted to the senses seen in the parallel training corpus and not trained on missing senses.

In contrast, we propose a method to mine addi-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 614–624
November 10–11, 2021.

tional data containing the contextual translations of rare and unseen senses without relying on them being in parallel corpora. Our method, called CMBT (**C**ontextually-**m**ined **B**ack-**T**ranslation), relies on contextualized cross-lingual word representations (CCWRs) to translate source language multi-sense words and find target language sentences containing the translations of the right senses. We then build a synthetic parallel corpus by back-translating these sentences, making sure that the original multi-sense words are contained on the source side, in order to extend the training corpus for better sense coverage. We illustrate our method in Figure 1. The advantage of our approach is that CCWRs, such as mBERT (Devlin et al., 2017) or XLM-R (Conneau et al., 2020), can be trained on cheap and large monolingual corpora covering a wide frequency range of word senses. By leveraging CCWR-mined sentences containing the translations of these senses in the form of a synthetic parallel corpus, our MT system is not restricted to frequent senses seen in the parallel training corpus.

We test our approach on English→German using the *MuCoW* test suite (Raganato et al., 2019, 2020). Although it was built to test overall WSD performance of MT systems, we create subsets of the provided training corpus to test on unseen and rare senses more directly. Our experiments show that using mined sentences as additional data for our NMT systems consistently improves the translation performance (F_1) of rare and unseen senses. Our proposed approach can be effectively applied to other language pairs as well, since the required resources are widely available.

2 Related Work

The problem of WSD is long-standing and extensively studied. Multiple neural systems were proposed, e.g., by relying on sequence-to-sequence architectures (Raganato et al., 2017), using sense embeddings (Kumar et al., 2019) or pre-trained language models (Pasini et al., 2021). It was shown that WSD positively impacts the performance of downstream applications, such as information retrieval (Zhong and Ng, 2012), sentiment analysis (Pilehvar et al., 2017) or topic classification (Shimura et al., 2019).

WSD is an important problem for MT as well. Previously it was shown that the translation performance can be improved by integrating word sense information into MT systems. In (Pu et al., 2017) sense labels were assigned to each multi-sense word using K-means clustering, which served as additional information for a statistical MT system. Similarly, explicit word sense information using WordNet was integrated into NMT systems in (Pu et al., 2018). Liu et al. (2018) leveraged sense embeddings induced by specialized LSTM modules, while lexical chains of semantically similar words within a document were employed in (Rios et al., 2017). Although these approaches do improve WSD performance, they are restricted to the senses seen frequently in the parallel training corpus.

In contrast, we focus on the improvement of senses that are missing or very rare by building a synthetic parallel corpus containing these senses using back-translation (Sennrich et al., 2016). Similarly, Huck et al. (2019) back-translated a carefully selected set of sentences to improve the translation of out-of-vocabulary (OOV) words, i.e., words that are contained in the text to be translated but not in the training corpus. They used bilingual fastText (Bojanowski et al., 2017) embeddings to find all translations of OOVs independent of their contexts. In contrast, we consider the whole sentence when translating multi-sense words in order to determine the right translation of the right sense used in the right context using CCWRs, which we show to be crucial to improve missing and rare sense translation. Arthaud et al. (2021) proposed a data augmentation approach to adapt MT systems to novel vocabulary in human-submitted translations using CCWRs. They generate training samples for the novel words by mining parallel sentence pairs with similar contexts and adding the novel words to them. In contrast, our approach does not rely on parallel sentences, using only monolingual data and back-translation.

Various datasets were proposed to test WSD, such as those released by the series of Senseval (Edmonds and Cotton, 2001; Mihalcea et al., 2004) and SemEval (Agirre et al., 2010; Navigli et al., 2013; Moro and Navigli, 2015) shared tasks. While most datasets are monolingual, Pasini et al. (2021) introduced XL-WSD supporting 18 languages allowing to evaluate zero-shot cross-lingual WSD approaches. To test how well MT systems can disambiguate multi-sense words in their outputs Rios et al. (2017) created the parallel corpus called ContraWSD where for each source sentence containing an ambiguous word two translations are given with

the correct and incorrect senses respectively which have to be scored by the MT systems. The MuCoW dataset was introduced for a more direct evaluation where instead of scoring given target language sentences the translations of multi-sense words in the MT systems' outputs are evaluated (Raganato et al., 2019, 2020). We use MuCoW to evaluate our approach.

3 Approach

The goal of CMBT is to incorporate context-dependent word translation that is able to deal with rare and unseen senses and leverage cheap monolingual data as additional training data for our NMT system for better multi-sense word translation. The main steps of our approach are the following: i) we detect multi-sense words in the source side of test corpus using BabelNet (Navigli and Ponzetto, 2012), which ii) we translate using CCWRs and mine target language sentences containing these translations. iii) We back-translate these sentences to the source language using a baseline NMT system. We use a special marker placed in the target language sentences, which are replaced with the multi-sense words on the source side, in order to ensure the presence of rare and unseen senses in the new corpus. Finally, we fine-tune our base NMT system using the gold and additional synthetic parallel data. We summarize the pipeline in our approach in Figure 1 and detail the three main steps below:

3.1 Multi-Sense Word Detection

As the first step, we identify multi-sense words in the test corpus relying on BabelNet, a publicly available multilingual lexical resource covering 284 languages (Navigli and Ponzetto, 2012). Since the MuCoW dataset focuses on nouns only, we first take all English nouns from the test corpus.[1] We then filter out single sense nouns by keeping only those which are contained in at least two synsets. However BabelNet has a very fine grained set of synsets which would result in a list containing many single sense nouns as well due to their inclusion in multiple synsets. Thus before filtering we merge some of the synsets using English-German interlingual links in BabelNet which specify possible German translations of the words in a given English synset. More precisely, we merge English synsets which have overlapping sets of translations. The filtering procedure using the merged synsets resulted in 3 732 multi-sense nouns containing 181 out of the 206 gold multi-sense words in the MuCoW test corpus.[2]

We note that although BabelNet covers a large set of languages, the language of the application area might not be supported. However, CMBT only requires a list of source language multi-sense words as input which can be acquired using unsupervised WSD systems as well, such as the word embeddings based *SenseGram* (Pelevina et al., 2016). We argue that our approach is robust against false positive multi-sense words, since we would mine sentences containing their single sense, thus the use of a high recall list is preferable in such cases.

3.2 Sentence mining

Given the multi-sense words we mine target language sentences containing the translations of their different senses. However we do not mine all possible senses of the words but only those which are contained in the input corpus to be translated, i.e., the source side of the test corpus in our case. For this we perform bilingual token-level sense retrieval (BTSR) (Liu et al., 2019) where the task given a source word in a context (sentence) is to retrieve its translation having the same sense along with a matching target context. More formally, given a $(w_s, c_s) \in V_s \times D_s$ pair the task is to retrieve $(w_t, c_t) \in V_t \times D_t$, such that w_t is the translation of w_s and the sense of w_s in context c_s matches the sense of w_t in c_t. V_s, V_t and D_s, D_t are the vocabularies and the monolingual datasets of the source and target languages respectively.

To mine relevant sentences, we take each multi-sense word contained in each source sentence as the input (w_s, c_s) pairs. Since a given word type is contained in multiple sentences, we perform mining using these sentences individually. As the translation target candidates, we take a target language monolingual corpus (see Section 4.3 for more details) and consider each word in each sentence as a candidate (w_t, c_t) pair. For each source input pair, we take the top-5[3] most similar target pair scored

[1] We used UDPipe (Straka and Straková, 2017) for POS tagging.

[2] Note that BabelNet was also used to build the list of multi-sense words in MuCoW but its output was further refined with parallel data and gold WSD annotations.

[3] Top-5 is common for bilingual lexicon induction.

by:

$$sim((w_s, c_s), (w_t, c_t)) = cos(E_{(w_s,c_s)}, E_{(w_t,c_t)}) \quad (1)$$

where $E_{(w,c)}$ is the CCWR of word w in context c and cos is the cosine similarity of two embeddings. As CCWR of a given word we averaged the corresponding vectors of the upper XLM-R layers (12-24), motivated by the findings of Ethayarajh (2019). We discuss further details of the used CCWR models in Section 4.1. Finally, the retrieved sentences are considered as the output of the mining process and used in the next step.

3.3 NMT System Update

In the last step, we update our baseline English→German NMT system with the gold parallel data and the sentences mined above. Using a system similar to the baseline but built in the reverse direction we back-translate the mined target language sentences by making sure that the original multi-sense word is contained in the back-translation. To achieve this we replace the related word in the target sentence with a special marker which is copied to the source side during translation. After translation we replace the special markers on both sides with the correct words. The following example depicts the process using a mined sentence for the multi-sense word *bank*:

Input:	*Ich gehe am **Ufer** entlang.*
Replace:	*Ich gehe am **<MARK>** entlang.*
Translate:	*I walk along the **<MARK>**.*
Restore:	*I walk along the **bank**.*

To learn the copy mechanism of the special marker we use parallel sentences containing the marker to train the NMT system used for back-translation. More precisely, 1% of the parallel sentences have one randomly selected source word and it's corresponding translation (determined by aligning the parallel data) replaced with the marker. Note that there is a small chance that the MT system does not generate the marker in the output in which case no replacement is performed. Finally, we update our baseline English→German MT system by running further training steps on the concatenated gold and synthetic parallel corpora. For further parameters we refer to Section 4.4.

4 Experimental Setup

4.1 Cross-Lingual Word Representations

As CCWRs we make use of XLM-R large[4] (Conneau et al., 2020), as previous works have shown good context-dependent cross-lingual correspondence in such multilingual models (Ethayarajh, 2019; Liu et al., 2019; Cao et al., 2019). Although they are multi-lingual, it was shown that their cross-lingual performance can be improved by applying an additional mapping step. Thus following Liu et al. (2019), we train a linear orthogonal mapping on XLM-R's context-average word type representations of word pairs extracted from the automatic word alignments in the parallel corpus which is used for MT training as well. The context-average representations are first length normalized and then mean centered prior to alignment as it was shown to improve the mapping quality (Artetxe et al., 2018). On top of the orthogonal mapping, we also apply the meeting-in-the-middle technique proposed by Doval et al. (2020) that learns additional linear mappings of both source and target languages to further improve their alignment. For exact details about the complete mapping process we refer to (Liu et al., 2019).

4.2 Baselines

Other than comparing CMBT with XLM-R to the *baseline* NMT system, we compare the approach to Huck et al. (2019), since it is able to leverage monolingual data to improve the translation of a list of words. More precisely, we translate multi-sense words with BWEs instead of XLM-R, to show the importance of context based word translation for the translation of multi-sense words. Since BWEs tend to rank the translations of words according to their frequency, this approach is comparable to the general back-translation approach, i.e., updating NMT systems on randomly sampled sentences, but focusing more on the ambiguous words. We build 300 dimensional *fastText* skipgram embeddings (Bojanowski et al., 2017) on Wikipedia dumps and align them using the same approach as for XLM-R (Liu et al., 2019). Similarly to CMBT, words are translated using cosine similarity taking top-5 most similar candidates. However, since BWE based bilingual lexicon induction (BLI) is

[4]Besides XLM-R, we experimented with mBERT (Devlin et al., 2017) as well but chose the former due to its superior word retrieval performance (See the experiment's results in Appendix A).

context independent, we pick all sentences containing any of the translations, which are then down sampled to match the number of sentences mined by CMBT for comparability. The rest of the steps, i.e., back-translation and system training, are the same as for CMBT. Our experiments show that although top-5 translations based on BWEs can cover multiple senses of some words, it is important to take the contexts into consideration as well in order to perform i) a more sense specific word translation and ii) mine sentences which do not only contain the target words but have similar contexts compared to the source sentences in the test corpus for an efficient MT tuning.

4.3 Monolingual Dataset

We use $2M$ randomly sampled German Wikipedia sentences for the mining process and restrict the vocabulary for translation candidates to the $500K$[5] most frequent words. We mine relevant sentences for all senses of the detected multi-sense words, including their frequent senses, since the frequency of senses in the training corpus is not known. The mined corpus contains 252 898 unique sentences.

4.4 MT Systems

We train base Transformer NMT models (Vaswani et al., 2017) on the gold parallel data discussed below with early-stopping based on validation perplexity. The model is trained on 4 Nvidia GTX 1080ti GPUs with a per-GPU batch size of 4096 tokens and by delaying stochastic gradient descent updates with a factor of 2. The final model is an average of the best 10 checkpoints, where checkpoints are saved every 500 updates. We use dropout and label smoothing with a value of 0.1.

Fine-tuning this initial system with the concatenation of the gold and synthetic data is done using the same hyper-parameters with early-stopping based on validation perplexity. The average of the best 10 checkpoints is chosen as the initial starting point for fine-tuning.

As a development set we merge newstest 2017-2019. We use a beam size of 4 for back-translation and 5 for the translation of MuCoW. All models are built using fairseq (Ott et al., 2019). The datasets are tokenized using Moses[6]. We use BPE splitting[7] with 32K merge operations computed jointly on the source and target data. Word alignment is performed using fastalign (Dyer et al., 2013).

4.5 MuCoW Dataset

We run experiments on the English→German translation direction of the MuCoW dataset (Raganato et al., 2020). It was created specifically to test translation quality of ambiguous words by specifying the word and its gold sense for each test sentence. The dataset provides small and big training parallel corpora containing $1.2M$ and $3.0M$ sentence pairs respectively. In order to test on rare and unseen senses more directly, we create two subsets of the latter. We remove sentence pairs containing the rarest sense of any of the given multi-sense words to test on *unseen* senses ($2.9M$ pairs). Secondly, we take a random 10% sample of the sentence pairs containing a multi-sense word and all pairs containing no multi-sense words in *sample-10* ($2.4M$ pairs) to test on rare senses. By sampling data uniformly random in case of the latter, we make sure that only the frequency of the multi-sense words gets decreased, while keeping their original sense distribution. Note that we only change the training set to have more word types with unseen and rare senses during training but keep the test set unchanged. Our *baseline* NMT system is trained on these training sets without the additional mined data.

We evaluate our MT systems on word level (F_1) using the official MuCoW evaluation script which calculates precision and recall values as:

$$P = \frac{|correct\ senses|}{|correct\ senses| + |incorrect\ senses|} \quad (2)$$

$$R = \frac{|correct\ senses|}{|test\ cases|} \quad (3)$$

where a test case is an occurrence of a multi-sense word in a test sentence. The dataset provides multiple correct translation options for a given sense, thus an occurrence of a multi-sense word (sense) is correctly translated if any of the translations of the correct sense are contained in the output sentence. A sense is incorrect if any of the translations of the wrong senses of the given multi-sense word are contained in the output sentence. A sense is

[5] We increase the $200K$ limit used in most BLI works in order to cover more rare words.

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

[7] https://github.com/rsennrich/subword-nmt

train	bin	#	system	acc@1	acc@5	F_1
unseen	0-0	10.5	baseline	-	-	17.14
			BWEs	6.69	13.51	25.39
			CMBT	**21.88**	**35.32**	**34.80**$\uparrow^{17.66}$
sample-10	0-20	5.9	baseline	-	-	35.53
			BWEs	1.93	4.18	37.70
			CMBT	**15.55**	**26.34**	**47.02**$\uparrow^{11.49}$
	20-40	3.2	baseline	-	-	60.98
			BWEs	7.92	19.78	60.80
			CMBT	**22.29**	**37.34**	**64.49**$\uparrow^{3.51}$

Table 1: Intrinsic and extrinsic evaluation in terms of acc@n and MuCoW F_1 scores. The rare senses in *sample-10* are shown broken down by relative frequency bins, while we present results of missing senses in *unseen*. The number of test cases in thousands per bin is shown in the third column (#). We compare the baseline and the improved MT systems with both BWEs and CMBT. We indicate the improvements (↑) compared to the baseline.

train	system	acc@1	acc@5	F_1
unseen	baseline	-	-	70.70
	BWEs	17.94	28.74	71.66
	CMBT	**28.30**	**43.90**	**73.51**$\uparrow^{2.81}$
sample-10	baseline	-	-	74.58
	BWEs	17.94	28.74	73.75
	CMBT	**28.30**	**43.90**	**75.86**$\uparrow^{1.28}$

Table 2: Evaluation of all (including frequent) senses when using *unseen* or *sample-10* training sets. Number of overall test cases are $25.3K$ in both sets. BLI results are the same for both training sets as they only affect the NMT system.

neither correct nor incorrect if none of the possible translations of the multi-sense word is contained. Furthermore sentences are lemmatized, thus all morphological variants of a word are accepted.

We also calculate BLEU scores using *sacrebleu* (Post, 2019) to show general MT performance. In addition, we evaluate the word translation accuracy of XLM-R and BWEs on the gold MuCoW multi-sense words contained in each test sentence. Similarly to BLI (Vulić and Korhonen, 2016), we calculate acc@n ($n \in 1, 5$) scores by testing if any of the correct translations of the gold sense in a given test example is among the n most similar translation candidates.

5 Results and Discussion

Unseen and rare sense translation We present both the intrinsic performance of BWEs (Huck et al., 2019) or CMBT (XLM-R) based word translation (acc@n) and extrinsic MT system based translation (F_1) in Table 1. We show results on senses in the test corpus which are missing from the *unseen* training corpus and detailed results on word senses that are rare (relative frequency compared to the other senses of a given word is between 20% and 40%) and very rare (with relative frequency between 0% and 20%) on *sample-10*.

In terms of acc@n CMBT word translation performs significantly better than BWEs. This is not surprising, since the context independent BWEs predict the same translations for a given multi-sense word for each sentence it is contained in. In contrast, XLM-R shows a better WSD performance by relying on the context in the sentences. Our improved NMT system using CMBT outperforms the baseline system in all setups in terms of F_1. It is especially effective on the unseen and very rare senses due to the additional synthetic sentence pairs containing these senses and their translations. In addition, it is also effective for the rare senses in the higher relative frequency range bin. BWEs based mining is also helpful for the unseen and very rare senses but it is less effective compared to CMBT. Although BWEs are context independent, by taking top-5 translations some of the senses can still be improved. On the other hand, BWEs have minor negative effects for the higher frequency range.

All sense translation We show results on all senses, i.e., senses with relative frequency higher than 40% as well, using the two training sets in Table 2. CMBT is also effective when evaluating on the whole MuCoW test dataset, but its performance is more prominent on the lower frequency ranges. In contrast, BWEs achieved only a slight improvement on *unseen* and some performance drop on *sample-10*.

Lexicon-regularized translation As mentioned, we built the list of English multi-sense words using BabelNet. Since it also contains translation options for each word, we investigate whether we can make use of this additional information. Fortunately, CMBT can be naturally extended to leverage such lexical resources.[8]

During sentence mining with the lexicon-regularized version of our approach (CMBT+), we restrict the set of translation candidates when translating a given word with XLM-R to its possible

[8]In comparison, it is not straightforward how we can add this information into a baseline NMT system where we cannot easily track translation for specific source words.

train	bin	system	F_1
unseen	0-0	ALL+	$25.21\downarrow^{9.59}$
		CMBT+	$34.55\downarrow^{0.25}$
	all	ALL+	$72.00\downarrow^{1.51}$
		CMBT+	$73.83\uparrow^{0.32}$
sample-10	0-20	ALL+	$43.37\downarrow^{3.65}$
		CMBT+	$46.75\downarrow^{0.27}$
	20-40	ALL+	$65.04\uparrow^{0.55}$
		CMBT+	$66.82\uparrow^{2.23}$
	all	ALL+	$75.95\uparrow^{0.09}$
		CMBT+	$76.50\uparrow^{0.64}$

Table 3: Evaluation of the senses per frequency bins as well as all senses using the lexicon-regularized systems on the two datasets. Differences compared to the best system (CMBT in tables 1 and 2) are indicated.

translations as given by BabelNet. Furthermore, we mine sentences based on all possible translations (ALL+) instead of taking top-5 ranked by XLM-R. Table 3 shows that the regularized systems achieved improvements compared to CMBT only in the higher frequency ranges but not in the missing or very rare sets. ALL+ achieved only minor improvements overall (*all*) on *sample-10*, while the performance decreased on *unseen*. This indicates that without focused data mining, sentences containing rare and unseen senses are suppressed by the frequent senses. In contrast, CMBT+ achieved improvements on both setups by following the sense distribution of the test set. However, the improvements are marginal which shows that the unregularized CMBT system is already able to retrieve the relevant senses of words without the additional information coming from BabelNet.

BLEU evaluation Finally, we show general MT performance on our training setups including the original MuCoW *big* setup as well for comparison in Table 4. It can be seen that our approach achieved improvements in terms of BLEU as well, further motivating its use. Similarly to F_1 scores the improvements are more prominent when evaluating only on sentences containing missing or rare senses. CMBT achieves best scores on the full test sets (all) of the *unseen* and *sample-10* setups, and a minor decrease on *big*. However, BLEU score differences are minor and they do not correlate well with F_1 improvements. As we show next, the minor differences in BLEU scores here are due to the fact that our approach mainly affects the translation of multi-sense words while leaving

train	bin	baseline	BWE	CMBT
unseen	0-0	23.0	23.2	**23.3**
	all	25.5	25.6	**25.7**
sample-10	0-20	22.3	22.3	**22.6**
	20-40	24.5	24.6	**24.7**
	all	25.0	25.0	**25.1**
big	all	**26.5**	**26.5**	26.4

Table 4: Machine translation performance (BLEU) on the complete MuCoW test set using the unmodified *big* and our two custom training sets. We show results on the missing (0-0) and rare senses (0-20 and 20-40) as well as on the complete test set (all). BLEU score achieved by Raganato et al. (2020) on big-all is 22.6.

the translation of other words intact. It is worth pointing out that BLEU scores may not be the ideal metric in this study as they are less sensitive to word-level translation improvement as compared with F1 scores. This is similar to the findings of Arthaud et al. (2021), who showed that improving the translation of a few selected words could lead even to a slight drop in BLEU.

Analysis We manually looked at the translations of a few multi-sense words to have a better understanding of our system. We present a few examples of the typical improvements and errors we found in Table 5. In example 1 both the BWEs based system and CMBT correctly translated *bank* to *Ufer* (river bank) which shows the positive effects of the additional data. In contrast, in 2 and 3 which are examples produced under the *sample-10* and *unseen* conditions respectively, only CMBT managed to pick the right senses due to the better exploitation of the given context. All systems are incorrect in 4, however the output of CMBT (brake pedal) is related to vehicle/gas pedals (the correct sense), while the base system's output, *Beschleuniger* is more related to physics and chemical reactions, such as particle accelerator or a catalyst. We reviewed the top-5 translations given by BWEs and XLM-R for *accelerator* when it has the *gas pedal* sense in a test sentence, and found that the translations reflect the outputs of the MT systems. This shows the effectiveness of the MT system's update process and that improving CCWR based translation could lead to further improvements.

In example 5 CMBT is misled by the *mossy bank*, thus outputs the *river bank* instead of the *bench* sense in contrast to the baseline which correctly used the frequent *Bank* word. Example 6 is incorrectly translated by all systems with different

1.	SRC	*It is seen from afar sprawling along the **banks** like a cowherd taking a siesta by the water-side.*
	BASE	Es scheint aus der Ferne zu sein, wie ein Kabeljau an der Wasserseite eine Siesta nimmt.
	BWE	Es scheint aus der Ferne an den **Ufer** zu ziehen wie ein Fisch, der an der Wasserseite eine Siesta nimmt.
	CMBT	Es scheint aus der Ferne an den **Ufer** zu rasen wie ein Hirsch, der an der Wasserseite eine Siesta nimmt.
	REF	*Schon von weitem sieht man den Ort am **Ufer** lang hingestreckt liegen, wie einen Kuhhirten, der sich faulenzend am Bache hingeworfen hat.*
	GLOSS	**river bank**;
2.	SRC	*Working men, kneeling on the **banks**, washed their bare arms in the water.*
	BASE	Arbeiter, die an den **Banken** knieten, wuschen ihre bloßen Waffen im Wasser.
	BWE	Arbeitende Männer knieten an den **Banken** nieder und wuschen ihre bloßen Arme im Wasser.
	CMBT	Arbeitende Männer knieten am **Ufer** nieder und wuschen ihre bloßen Arme im Wasser.
	REF	*Arbeiter kauerten am **Ufer** und wuschen sich die Arme in der Flut.*
	GLOSS	**river bank**; **bench**;
3.	SRC	*The physician, to whom the soldiers of the **watch** had carried him at the first moment...*
	BASE	Der Arzt, zu dem ihn die Soldaten der **Uhr** im ersten Augenblick getragen hatten...
	BWE	Der Arzt, zu dem ihn die Soldaten der **Uhr** im ersten Augenblick getragen hatten...
	CMBT	Der Arzt, zu dem ihn die Soldaten der **Wache** im ersten Augenblicke getragen hatten...
	REF	*Der Heilkünstler, zu welchem die Soldaten der **Wache** ihn im ersten Augenblicke getragen...*
	GLOSS	**guard**; **timepiece**;
4.	SRC	*Try to avoid depressing the **accelerator** pedal beyond the pressure point (kickdown).*
	BASE	Versuche zu vermeiden, den **Beschleuniger**-Pedal über den Druckpunkt hinaus zu deprimieren (Kickdown).
	BWE	Versuche, den **Beschleunigerpedal** über den Druckpunkt hinaus nicht zu deprimieren (Kickdown).
	CMBT	Versuche, das **Bremspedal** über den Druckpunkt hinaus nicht zu deprimieren (Kickdown).
	REF	*Treten Sie das **Fahrpedal** möglichst nicht über den Druckpunkt durch (Kickdown).*
	GLOSS	**gas pedal**; **brake pedal**; **catalyst, (particle) accelerator**;
5.	SRC	*A lover finds his mistress asleep on a mossy **bank**;...*
	BASE	Ein Liebhaber findet seine Geliebte schlafend auf einer feuchten **Bank**;...
	BWE	Ein Liebhaber findet seine Geliebte schlafend auf einem feuchten **Bankett**;...
	CMBT	Ein Geliebter findet seine Geliebte schlafend auf einem feuchten **Ufer**;...
	REF	*Ein Liebender findet seine Geliebte auf einer moosigen **Bank** eingeschlafen;...*
	GLOSS	**bench**; **banquet**; **river bank**;
6.	SRC	*I should like to deal with one concrete point, the question of the electronic **counter**.*
	BASE	Ich möchte auf einen konkreten Punkt eingehen, die Frage des elektronischen **Gegensatzes**.
	BWE	Ich möchte mich mit einem konkreten Punkt befassen, der Frage des elektronischen **Automaten**.
	CMBT	Ich möchte auf einen konkreten Punkt eingehen, die Frage des elektronischen **Zählers**.
	REF	*Eingehen möchte ich auf einen konkreten Punkt, den Punkt der elektronischen **Schalter**.*
	GLOSS	**checkout counter**; **contrast, opposition, difference**; **vending machine**; **electricity/energy meter**;

Table 5: Example sentences highlighting the multi-sense words and their translations. For each source sentence (SRC) with given reference translation (REF) we compare the baseline (BASE) to the BWE and CMBT based systems. Word senses (GLOSS) are color coded.

errors.

Additionally, by comparing the full outputs of the systems it can be seen that our approach is non-invasive, i.e., it mostly affects the translations of multi-sense words and leaves the other parts of the sentences unchanged compared to the baseline, which is a big advantage of our approach and also explains the small BLEU differences in Table 4.

Finally, we present mined sentences based on two example source sentences containing the word bank in Table 6. The sentences indicate that our XLM-R based mining technique not only outputs the translation of the right sense but the mined sentences have similar contexts to the source sentences. This allows the fine-tuned MT system to leverage information learned from sentences that are closely related to the input sentence during translation.

6 Conclusions

In this paper we proposed CMBT, a simple and effective approach for improved rare and unseen word sense translation. It serves as a general framework that effectively exploits the context-dependent cross-lingual correspondence from a pre-trained CCWR for an MT system. We show CMBT brings significant improvements for multi-sense word translation on the English→German MuCoW test set. The improvements are the most pronounced when we directly targeted the evaluation of the difficult rare and unseen senses. As the only requirement of CMBT, on top of the parallel data necessary for the training of the MT system, is a monolingual corpus and an off-the-shelf pre-trained multilingual model, CMBT can be applied

1.	SRC	*For example, a wife learns that her husband put money in the **bank** in his name rather than in a joint account.*
	top-1	Der Abfluss bei einer Überweisung erfolgt im Zeitpunkt der Abgabe des Überweisungsauftrags an die **Bank**...
	top-1 BT	The flow of a transfer is made when the contract is delivered to the **bank**...
	top-2	Osama und Yeslam bin Laden hatten von 1990 bis 1997 ein gemeinsames Konto bei der Schweizer **Bank** UBS.
	top-2 BT	Osama and Yeslam bin Laden shared an account at the Swiss **Bank** UBS between 1990 and 1997.
2.	SRC	*At this decisive moment in Dutch history my father was positioned on the **bank** of the river Waal near the city of Nijmegen.*
	top-1	Der Highway führt nördlich am Stadtzentrum vorbei und gelangt von dort an das **Ufer** des Ontariosees.
	top-1 BT	The Highway passes north of the center of the city and then reaches the **bank** of Lake Ontario.
	top-2	Die Großstadt Pakokku liegt auf der nördlichen **Uferseite**$^{[bank-side]}$ des Irrawaddy 30 Kilometer nordöstlich von Bagan...
	top-2 BT	The big city of Pakokku is situated on the northern **bank** of Irrawaddy, 30 kilometres northeast of Bagan...

Table 6: Mining examples with XLM-R for two source sentences (SRC) containing the two senses (financial and river) of the word bank. We show the 2 highest scoring candidates and their back-translations (BT).

easily to other languages and MT systems in the future.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback and the Cambridge LMU Strategic Partnership for seed funding for this project.[9] We acknowledge Peterhouse College at University of Cambridge for funding Qianchu Liu's PhD research. The work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (No. 640550) and by the German Research Foundation (DFG; grant FR 2829/4-1) awarded to Alexander Fraser as well as by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (No. 648909) and the ERC PoC Grant MultiConvAI (No. 957356) awarded to Anna Korhonen.

References

Eneko Agirre, Oier Lopez de Lacalle, Christiane Fellbaum, Shu-K ai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers. 2010. SemEval-2010 Task 17: All-Words Word Sense Disambiguation on a Specific Domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 75–80.

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*, pages 789–798.

Farid Arthaud, Rachel Bawden, and Alexandra Birch. 2021. Few-shot learning through contextual data augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1049–1062.

[9] https://www.cambridge.uni-muenchen.de

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.

Steven Cao, Nikita Kitaev, and Dan Klein. 2019. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2017. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2020. Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 294–304.

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 Associat ion for Computational Linguistics: Human Language Technologies*, pages 644–648.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5.

Denis Emelin, Ivan Titov, and Rico Sennrich. 2020. Detecting Word Sense Disambiguation Biases in Machine Translation for Model-Agnostic Adversarial Attacks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7635–7653.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 55–65.

Matthias Huck, Viktor Hangya, and Alexander Fraser. 2019. Better OOV Translation with Bilingual Terminology Mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681.

Frederick Liu, Han Lu, and Graham Neubig. 2018. Handling Homographs in Neural Machine Translation. In *Proceeding of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, pages 1336–1345.

Qianchu Liu, Diana Mccarthy, Ivan Vulić, and Anna Korhonen. 2019. Investigating Cross-lingual Alignment Methods for Contextualized Embeddings with Token-Level Evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, pages 33–43.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluatio n of Systems for the Semantic Analysis of Text*, pages 25–28.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linki ng. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual 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 222–231.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. XL-WSD: An Extra-Large and Cross-Lingual Evaluation Framework for Word Sense Disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Maria Pelevina, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. 2016. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183.

Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a seamless integration of word senses into downstream nlp applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1857–1869.

Matt Post. 2019. A Call for Clarity in Reporting BLEU Scores. In *Proceedings ofthe Third Conference on Machine Translation*, pages 186–191.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating Weakly Supervised Word Sense Disambiguation into Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 6:635–650.

Xiao Pu, Nikolaos Pappas, and Andrei Popescu-Belis. 2017. Sense-Aware Statistical Machine Translation using Adaptive Context-Dependent Clustering. In *Proceedings of the Second Conference on Machine Translation*, pages 1–10.

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.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW Test Suite at WMT 2019: Automatically Harvested Multilingual Contrastive Word Sense Disambiguation Test Sets for Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 470–480.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020. An Evaluation Benchmark for Testing the Word Sense Disambiguation Capabilities of Machine Translation Systems. In *Proceedings of the 12th Conference on Language Resources and Evaluation*, pages 3668–3675.

Annette Rios, Laura Mascarell, and Rico Sennrich. 2017. Improving Word Sense Disambiguation in Neural Machine Translation with Sense Embeddings. In *Proceedings of the Conference on Machine Translation*, pages 11–19.

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*, pages 86–96.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2019. Text categorization by learning predominant sense of words as auxiliary task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1109–1119.

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.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation*, pages 26–35.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2020. Encoders Help You Disambiguate Word Senses in Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1429–1435.

Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. 2019. Lost in Translation: Loss and Decay of Linguistic Richness in Machine Translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 222–232.

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.

Ivan Vulić and Anna Korhonen. 2016. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 247–257.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, et al. 2019. Microsoft Research Asia's Systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation*, pages 424–433.

Zhi Zhong and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 273–282.

A mBERT vs. XLM-R

We report the results of our initial word translation accuracy experiments using off-the-shelf mBERT and XLM-R (large) on all the multi-sense words provided by the gold MuCoW test set in Table 7. For efficiency, we randomly sampled $100K$ target sentences from Wikipedia as the candidate pool instead of the $2M$ described in Section 4.3. We take the average of the top-half layers of mBERT (top 6 layers) and XLM-R (top 12 layers) respectively when calculating word representations. We show that XLM-R performs significantly better than mBERT.

Model	$acc@1$	$acc@5$	$acc@10$
mBERT	21.40	32.16	37.29
XLM-R	**27.13**	**38.76**	**43.81**

Table 7: Comparing the translation accuracy of off-the-shelf mBERT and XLM-R on the MuCoW test set.

Pushing the Right Buttons: Adversarial Evaluation of Quality Estimation

Diptesh Kanojia[1], **Marina Fomicheva**[2], **Tharindu Ranasinghe**[3], **Frédéric Blain**[4], **Constantin Orăsan**[5], **Lucia Specia**[6]
[1,5]Centre for Translation Studies, University of Surrey
[2]University of Sheffield [3,4]University of Wolverhampton [6]Imperial College London
[1,5]{d.kanojia,c.orasan}@surrey.ac.uk, [2]m.fomicheva@sheffield.ac.uk,
[3,4]{t.d.ranasinghehettiarachchige,f.blain}@wlv.ac.uk,
[6]l.specia@imperial.ac.uk

Abstract

Current Machine Translation (MT) systems achieve very good results on a growing variety of language pairs and datasets. However, they are known to produce fluent translation outputs that can contain important meaning errors, thus undermining their reliability in practice. Quality Estimation (QE) is the task of automatically assessing the performance of MT systems at test time. Thus, in order to be useful, QE systems should be able to detect such errors. However, this ability is yet to be tested in the current evaluation practices, where QE systems are assessed only in terms of their correlation with human judgements. In this work, we bridge this gap by proposing a general methodology for adversarial testing of QE for MT. First, we show that despite a high correlation with human judgements achieved by the recent SOTA, certain types of meaning errors are still problematic for QE to detect. Second, we show that on average, the ability of a given model to discriminate between meaning-preserving and meaning-altering perturbations is predictive of its overall performance, thus potentially allowing for comparing QE systems without relying on manual quality annotation.

1 Introduction

Quality Estimation (QE) is the task of predicting the quality of Machine Translation (MT) output in the absence of human reference translation. Recent QE models based on multilingual pre-trained representations (Ranasinghe et al., 2020) have shown impressive results achieving up to 0.9 Pearson correlation with human judgements of translation quality at sentence level (Specia et al., 2020). Not unlike other NLP systems, QE systems are typically tested on held-out datasets. On the one hand, such evaluation usually requires collecting additional human judgements and thus cannot be easily extrapolated to a different usage scenario, for example, a new language pair. On the other hand, evaluation on a given test set can hide performance issues related to the phenomena that are underrepresented in the data but are critical to the reliable performance of the system. Finally, a single statistic capturing overall performance does not provide any insights on the strengths and weaknesses of a given approach. As a way to overcome these limitations, we explore adversarial evaluation for QE. Specifically, we introduce two types of changes to high-quality MT outputs: meaning-preserving perturbations (MPPs) and meaning-altering perturbations (MAPs). Intuitively, we expect a strong QE system to assign lower scores to the sentences containing MAPs compared to the sentences with MPPs. Based on this intuition, we devise experiments to systematically test a set of five different QE systems by comparing the scores they produce for sentences containing MPPs and MAPs. We use the difference in the predicted scores as a way of detecting specific problems as well as for assessing the overall performance of the systems. Our main findings[1] can be summarised as follows:

- Overall, SOTA QE models are robust to MPPs and are sensitive to MAPs, thus supporting the claims that such models are indeed strong predictors of MT quality.
- SOTA QE models fail to properly detect certain types of MAPs, such as negation omission, which highlights the weaknesses of these models that cannot be detected using standard evaluation methods.
- The overall results of our probing experiments on a set of QE models are consistent with their correlation with human judgements. This suggests that the proposed evaluation methodology can be used to assess the performance of QE models with no need for collecting gold standard human annotation.

[1]Code available from https://github.com/dipteshkanojia/qe-evaluation.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 625–638
November 10–11, 2021.

In the remainder of this paper, we first discuss related work on probing for NLP (Section 2). We then describe the dataset (Section 3) and QE models used in our experiments (Section 4). We introduce our probing setup and strategies in Section 5 and present and discuss the results in Section 6.

2 Related Work

Very few studies have analysed the performance of QE models beyond correlation with human judgements on held-out datasets. To the best of our knowledge, the only work that analyses the behaviour of QE models is Sun et al. (2020). On various datasets popularly used for training QE models, they show that they contain certain biases, such as a skew towards high-quality MT outputs and lexical artefacts that are picked up by the SOTA architectures, *e.g.,* sentences with certain tokens tend to have high or low scores. They also show that QE models can perform very well on these datasets by encoding only the source or target sentences. By contrast, we study the behaviour of the models under specific linguistic conditions. Our experiments show that the models are not sensitive to certain meaning errors, which is in line with (Sun et al., 2020)'s assumption that SOTA QE models do not capture adequacy.

For MT, various studies have shown that models can achieve high performance on clean data, they are very brittle to noisy inputs, where both synthetic (e.g. character flips) or natural (social media data) noise is used to probe models (Belinkov and Bisk, 2018; Khayrallah and Koehn, 2018; Li et al., 2019; Passban et al., 2020). For other NLP tasks, black-box methods for adversarial evaluation have been proposed that apply meaning-preserving perturbations in order to test whether the models are sensitive to changes in the input (Ribeiro et al., 2018). Different from this line of work, we probe the robustness of QE models to spurious changes but also sensitivity to relevant changes, such as meaning errors. Ribeiro et al. (2020) recently devised a general methodology for behavioural testing of NLP models. They generate a subset of simple examples meant to test general linguistic capabilities expected from an NLP system. However, the linguistic capabilities tested within this framework are not directly applicable to the QE task. They could not, for example, capture the ability of a QE system to detect omission errors or copy errors in translation.

3 Dataset

The dataset used in this paper is a subset from the WMT 2020 Quality Estimation Shared Task 1, sentence-level prediction (Specia et al., 2020). This data consists of seven language pairs which can be classified as high-resource [English-German (En-De), English-Chinese (En-Zh)], medium-resource [Russian-English (Ru-En), Romanian-English (Ro-En), Estonian-English (Et-En)], and low-resource [Sinhala-English (Si-En), Nepalese-English (Ne-En)] pairs. Except for Ru-En, sentences are extracted solely from Wikipedia. The Ru-En data also contains additional sentences from Reddit (Fomicheva et al., 2020). The data was collected by machine translating sentences sampled from source-language articles using SOTA NMT models built using the *fairseq* toolkit (Ott et al., 2019). The data was annotated with a variant of Direct Assessment (DA) scores (Graham et al., 2017) by professional translators. Each translation was rated with a score in 1-100, according to the perceived translation quality by at least three translators (Specia et al., 2020). The goal of QE systems built on this data is to predict a *z-score* normalised mean DA for each *test* source-target pairs, which we further standardise between 0 and 1.

In the original dataset, 9K sentences per language pair were randomly split in training (7K), validation (1K) and test (1K). In this study, we focus on probing the models by modifying the target side (translations) with various perturbations. To keep the experiments consistent across the language pairs, we only consider the five pairs with English as the target language.

We use the standard training partition of the data to train our QE models. To evaluate our probes, the assumption made is that sentences with perturbations should lead to lower predicted QE scores than original sentences. However, this assumption only holds if we can ensure that the original sentences have high enough quality since perturbing very low-quality sentences with already very low scores would not necessarily lead to further degradations. Therefore, we create a subset of the validation + test sets by applying the threshold of 0.7 on the standardised human (DA) scores to reflect high quality, based on the definition of the DA scores used as guidelines for annotators in this dataset. Table 1 shows the resulting number of validation + test instances for each language. We hereafter refer to this set as our **test set**.

Language Pair	Ru-En	Ro-En	Et-En	Si-En	Ne-En
#sentences	1245	1035	766	404	100
Low-resource	No	No	No	Yes	Yes

Table 1: The number of selected sentences in our test set for each language pair. These are sentences judged to have high-enough quality by human translators.

4 QE Models

We choose three categories of heavy- to light-weight models for sentence-level QE models: first, the SOTA TransQuest with three variants MonoTransQuest, SiameseTransQuest and MultilingualTransQuest (Ranasinghe et al., 2020); second, the LSTM-based Predictor-Estimator approach (Kim et al., 2017) and third, the unsupervised method SentSim (Song et al., 2021).

MonoTransQuest (MonoTQ) This regression architecture encodes a concatenated source-target sentence pair using a transformer encoder. The architecture adds a softmax layer on top of the CLS token of the transformer to predict the quality of the translation. MonTransQuest architecture has separate pretrained QE models based on XLM-Roberta-Large (Conneau et al., 2020) for all seven language pairs from WMT 2020 QE Task 1.

SiameseTransQuest (SiameseTQ) This architecture uses a siamese network with two transformer models to encode the source and the target sentences separately. The architecture adds a max-pooling layer on top of the token embeddings of each transformer and calculates the cosine similarity between the outputs of the two pooling layers to predict the quality of the translation. Similar to *MonoTQ*, SiameseTQ has separate pretrained QE models based on XLM-Roberta-Large for all seven language pairs from WMT 2020 QE Task 1.

MultilingualTransQuest (MultiTQ) This architecture is based on MonoTQ but is trained on aggregated QE data for all seven language pairs from the WMT 2020 QE Task 1, resulting in one model for all the language pairs. This model is also based on XLM-Roberta-Large.

Predictor-Estimator (OpenKiwi) This is a two-stage architecture, where the Predictor model is an encoder-decoder RNN trained on parallel data (source-reference); in this case, the same data is used to train the respective NMT model for each language pair. Its output is then fed to the Estimator, a unidirectional RNN trained on QE data, to produce the quality estimates. Compared to TransQuest, the PredEst architecture does not rely on heavily pre-trained representations, resulting in a lighter model. For our experiments, we use the implementation in OpenKiwi (Kepler et al., 2019), which was provided as the baseline for the WMT 2020 QE Shared Task.

SentSim This is an unsupervised method to QE that uses a combination of cross-lingual word and cross-lingual sentence similarity scores to produce a sentence-level quality score. The word-level similarity is extracted using BERTScore (Zhang et al., 2020) between source and MT sentences, while sentence-level similarity is measured as the cosine similarity between the source and MT sentences representations. Both word and sentence-level representations are extracted using a cross-lingual pre-trained model, namely, XLM-Roberta-Base (Conneau et al., 2020).

5 Probing Strategies

In this section, we introduce the rationale for two types of probes: meaning-preserving and meaning-altering perturbations. We then describe each perturbation and discuss the experimental setup for the probing.

We define a **meaning-preserving perturbation (MPP)** as a small change in the target-side translation that might affect the translation but should not affect the overall meaning of the sentence. For example, removing punctuation marks from the translated sentence should not affect the meaning conveyed by the text. By contrast, **meaning-altering perturbations (MAP)** should alter the meaning conveyed by the translation, for example, replacing a random word with its antonym or randomly replacing a content word. By introducing MAPs, we focus on probing models for whether they capture (lack of) adequacy in translations. Given that SOTA QE models are based on pre-trained representations obtained from strong language models, it has been hypothesised that they could be biased by the fluency of translations (Sun et al., 2020).

We, therefore, design two types of perturbations: MPPs, which might affect fluency but not adequacy, and MAPs, which affect adequacy. Perturbations are only introduced in the translations to mimic translation errors. We have chosen perturbations that can be introduced using automated methods,

Source	**În alegerile europarlamentare din 2014, UKIP, partid de extremă dreaptă, a obținut peste 20 de locuri in parlamentul european.**	
Reference	**In the 2014 European Parliamentary elections, UKIP, a right-wing party, obtained more than 20 seats in the European Parliament.**	**S1**
Translation	**In the 2014 European Parliamentary elections, UKIP, party of extreā dreaptā, obtained more than 20 seats in the European Parliament.**	0.81
MPP1	In the 2014 European Parliamentary elections UKIP party of extreā dreaptā obtained more than 20 seats in the European Parliament	0.79
MPP2	In the 2014 European Parliamentary elections**!** UKIP(party of extreā dreaptā. obtained more than 20 seats in the European Parliament**?**	0.69
MPP3	In 2014 European Parliamentary elections, UKIP, party of extreā dreaptā, obtained more than 20 seats in European Parliament.	0.80
MPP4	In **such** 2014 European Parliamentary elections , UKIP , party of extreā dreaptā , obtained more than 20 seats in **those** European Parliament.	0.69
MPP5	**IN** the 2014 **EUROPEAN** Parliamentary **ELECTIONS**, UKIP, party of extreā **DREAPTĀ**, **OBTAINED** more **THAN 20 SEATS** in **THE EUROPEAN PARLIAMENT**.	0.76
MPP6	in the 2014 European parliamentary elections, **ukip**, party of extreā dreaptā, obtained more than 20 seats in the European Parliament.	0.75

Table 2: An example of each MPP from our dataset for Ro-En. 'Translation' is the original machine translated sentence for the given source sentence, which was assigned an average DA score of 0.70 by human annotators (in 0-1). S1 are scores from the MonoTransQuest architecture. The reference translation is only shown for readability, as it was not used by humans nor QE models.

Source	На слушании в декабре Блэквуд сказал, что не имел намерения оскорбить буддизм, когда размещал изображение, а после того, как осознал, что оно вызвало массовое возмущение, удалил его и опубликовал извинение.	
Reference	**At a hearing in December, Blackwood said he had not intended to offend Buddhism when he posted the image, and after realizing it had caused widespread outrage, deleted it and issued an apology.**	**S1**
Translation	**At a hearing in December, Blackwood said he had not intended to offend Buddhism when he posted the image, and after realizing it had caused widespread outrage, deleted it and issued an apology.**	0.83
MAP1	At a hearing in December, Blackwood said he **had intended** to offend Buddhism when he posted the image, and after realizing it had caused widespread outrage, deleted it and issued an apology.	0.82
MAP2	At a hearing **in,** Blackwood said he had not intended to offend Buddhism when he posted the image, and after realizing it had caused widespread outrage, deleted it and issued an apology.	0.82
MAP3	At a hearing in December, Blackwood said he had not intended to offend Buddhism when he posted the image, and after realizing **realizing** it had caused widespread outrage, deleted it and issued an apology.	0.81
MAP4	At a hearing in December, Blackwood said he had not intended to offend Buddhism **party** when he posted the image, and after realizing it had caused widespread outrage, deleted it and issued an apology.	0.82
MAP5	At a hearing in December, Blackwood said he had not intended to offend Buddhism when he posted the image, and after realizing it had caused widespread **Ferris**, deleted it and issued an apology.	0.80
MAP6	**at** a hearing in **japan**, **bailey admitted graham did** not intended to offend **buddhism** when **buddhist** posted the **video**, and after realizing he has caused widespread outrage, deleted it and issued **her** apology.	0.77
MAP7	At a hearing in December, Blackwood said he **lack** not intended to **keep** Buddhism when he posted the image, and after realizing it **refuse** caused widespread outrage, **record** it and **recall** an apology.	0.76
MAP8 (Russian)	На слушании в декабре Блэквуд сказал, что не имел намерения оскорбить буддизм, когда размещал изображение, а после того, как осознал, что оно вызвало массовое возмущение, удалил его и опубликовал извинение.	0.83

Table 3: An example of each MAP from our dataset for Ru-En. 'Translation' is the original machine translated sentence for the given source sentence, which was assigned an average DA score of 0.88 by human annotators (in 0-1). S1 are scores from MonoTransQuest architecture, and the reference translation is only shown for readability, as it was not used by humans nor QE models.

and we carefully select perturbations relevant for MT, *e.g.,* rare errors such as the omission of negation, and known errors such as omission of words from translation. Each type of perturbation is introduced independently of others, one perturbation per target sentence. We note that most of our perturbations are general enough such that they apply to all sentences in our test set. An exception is the removal of negation which can only be applied to sentences which contain a negation marker.

We analyse the behaviour of QE models by comparing the difference in the scores predicted after MPP/MAPs are applied to the test set compared to the original, unperturbed test set. We expect a strong QE model to predict lower scores to the version of the test set containing sentences with MPP and MAP, and – more importantly, a higher score to sentences with MPP than to sentences with MAP. Each of our probes is detailed below, categorised either as an MPP or as an MAP.

5.1 Meaning-Preserving Perturbations

We designed the following MPPs. In order to ensure sufficient randomisation of the experiments, we repeat MPP2, 4, 5 and 6, twenty times for each sentence and average the QE scores obtained for these twenty perturbations. Other MPPs, *e.g.,* removing all punctuations in the translation, can only result in one new version of the translation, and therefore, repetitions are not needed.

Removal of Punctuations (MPP1): We remove any punctuation marks from the translation using the standard *string* library in Python, for this perturbation.

Replacing Punctuations (MPP2): In this perturbation, each punctuation mark in the transla-

tion is replaced with another randomly chosen punctuation mark.

Removal of Determiners (MPP3): We use the *spaCy*[2] Part-of-speech (POS) tagger to identify determiners, and then remove them from the translation.

Replacing Determiners (MPP4): Each word labelled as a determiner with the help of spaCy POS tagger in the translation is replaced with another randomly chosen determiner from a list.

Change in Word-casing (MPP5/MPP6): We select random content words from the translation and convert them to UPPERCASE to generate a set of perturbed translations (MPP5). Additionally, we select content words randomly from the translation and convert them to lowercase to generate another set of perturbed translations (MPP6).

For each of the perturbations described above, we provide an example in Table 2, along with the scores predicted from our SOTA (MonoTQ) QE system.

5.2 Meaning-Altering Perturbations

We choose the following probes as MAP. We ensure sufficient randomisation of the experiments by repeating MAP2, 3, 4, 5, 6, and 7, twenty times for each sentence, and average the QE scores obtained for these twenty perturbations. For MAPs 1, and 8, we can produce only one version of the sentence.

Removal of Negation Markers (MAP1): For this perturbation, all the *negation markers* like "no", "not", "n't" *etc.* are removed.

Removal of Random Content Words (MAP2): We select a random content word from the translation and remove it.

Duplication of Random Content Words (MAP3): We choose a random content word from the translation and add it at the immediate next position index, thus duplicating its occurrence.

Insertion of Random Words (MAP4): We populate a vocabulary of words from the complete set of translations in our test set. From this vocabulary, we choose a word and insert it at a random position in the sentence, ensuring that the previous word and the next word are not the same to avoid duplication.

Replacing Random Content Words (MAP5): We choose a random content word from the translation and replace it with another word from the vocabulary created as discussed in MAP4.

BERT-based Sentence Replacement (MAP6): We obtain sentence replacements based on the BERT-base model (Devlin et al., 2019), with the help of a data augmentation library[3] (Ma, 2019). This library uses a word replacement approach proposed by Kobayashi (2018) and generates a sentence synonymous to the input provided. We observe that BERT-generated synonymous sentences replace content words which alter the inherent meaning of the input sentence and hence, treat this perturbation as MAP.

Replacing Words with Antonyms (MAP7): With the help of the data augmentation library[3], we generate perturbed translations where we replace random words in the sentence with their antonyms from the English Wordnet (Miller et al., 1990).

Source Sentence as Target (MAP8): We replace the translation with the source side sentence to observe the effect on QE scores when the source sentence is evaluated by the QE model, instead of the target side translation. Such a perturbation results in the model input to become *source-source* instead of *source-target*.

For each of the perturbations described above, we provide an example in Table 3, along with the scores predicted via SOTA (MonoTQ) system.

6 Results and Discussion

In this section, we discuss the results obtained from our probing experiments using various QE models.

6.1 Do perturbations affect SOTA QE models?

We start by analysing the behaviour of MonoTQ, as the best performing SOTA QE model on the dataset used in this paper, under different types

[2] spaCy API

[3] GitHub: makcedward/nlpaug

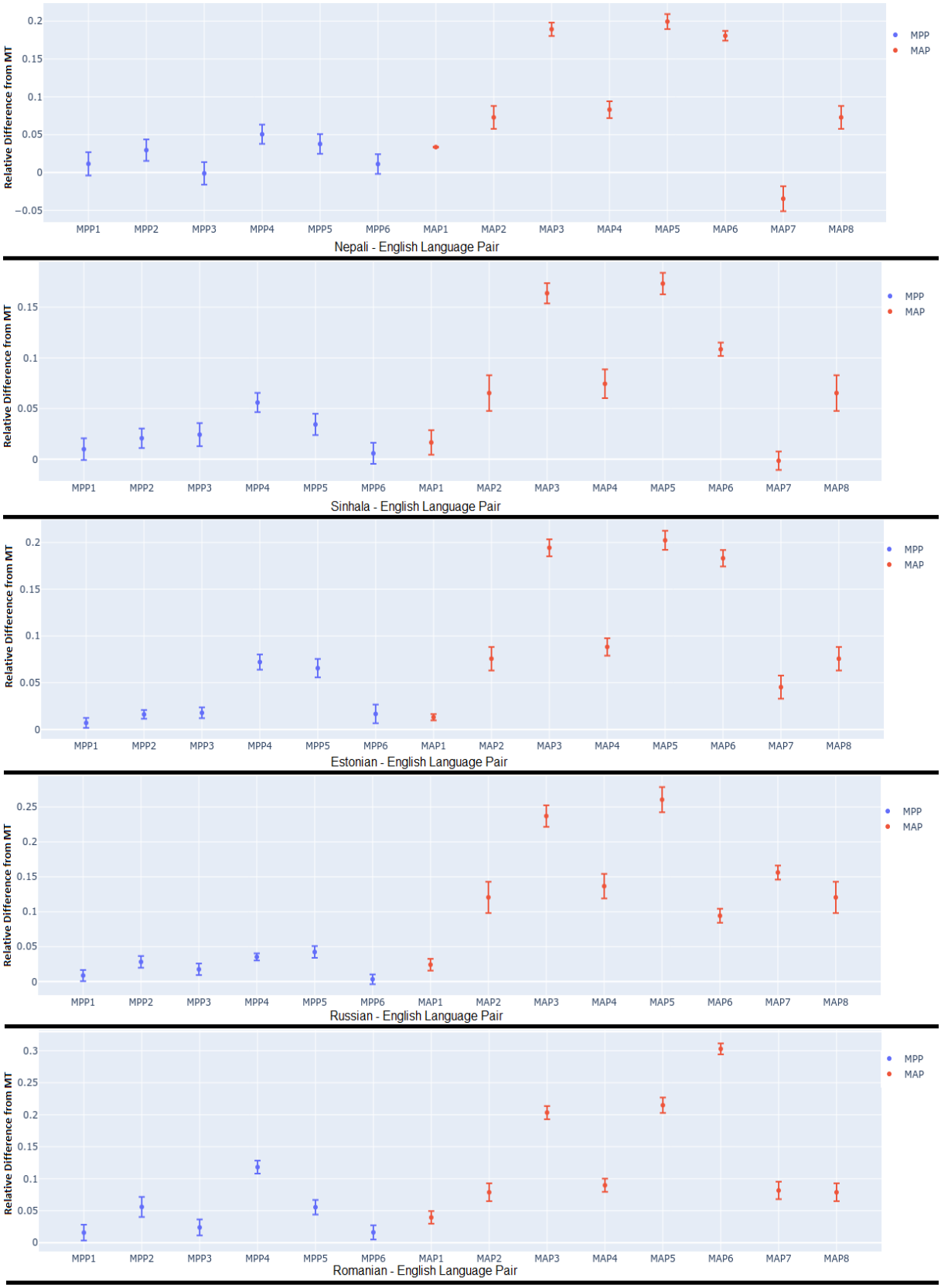


Figure 1: Average difference between the predicted QE scores for original translations and each perturbation across the test set for each language pairs (Y-axis->MT - x, where x is the perturbation as labelled on the X-axis), using the SOTA MonoTQ architecture.

	Ru-En			Ro-En			Et-En			Si-En			Ne-En		
	MT	MPP	MAP	MT	MPP	MAP	MT	MPP	MAP	MT	MPP	MAP	MT	MPP	MAP
MonoTQ	0.81	0.78	**0.66**	0.82	0.80	**0.74**	0.81	0.79	**0.73**	0.71	0.65	**0.64**	0.75	0.74	**0.68**
SiameseTQ	0.86	**0.85**	0.86	0.58	0.57	**0.52**	0.92	**0.91**	**0.91**	0.58	0.57	**0.52**	0.68	0.68	**0.65**
MultiTQ	0.79	0.75	**0.68**	0.79	0.74	**0.66**	0.77	0.73	**0.66**	0.62	0.58	**0.52**	0.63	0.60	**0.52**
OpenKiwi	0.78	0.78	0.78	0.78	**0.75**	0.77	0.71	**0.70**	**0.70**	0.62	0.60	**0.57**	0.50	**0.48**	**0.48**
SentSim	0.54	0.57	0.57	0.78	0.76	**0.72**	0.50	0.53	0.52	0.41	0.43	0.41	0.47	0.52	0.50

Table 4: Average predicted scores by all QE models on the test set for the original (unperturbed) machine translation (MT), versus its version with meaning-preserving perturbations (MPP) and meaning-altering perturbations (MAP). Between MPP and MAP, we boldface the lowest average scores, if lower than MT.

of perturbations. Figure 1 shows the difference between the average predicted score for our original test set (Table 1) before perturbations and the same subset of sentences perturbed using MPP and MAP. In comparison to the average scores for the initial set of translations, the expected behaviour for a strong QE model is to assign the same or slightly lower scores to and their MPP counterparts, but substantially lower scores to the MAP variants. Based on this premise, we can make the following observations from Figure 1. The other graphs obtained from SiameseTQ model, MultiTQ model, OpenKiwi system, and the Unsupervised method are present in Appendix A.

Models are robust to MPPs and sensitive to MAPs Overall, sentences with MPPs result in a small drop in the scores with respect to the original set of translations, especially when compared to the sentences containing MAPs. Conversely, perturbations that affect sentence meaning have a larger impact on the scores. Thus, SOTA QE models are indeed capable of discriminating between the two types of changes.

Models fail to detect important MAPs However, MonoTQ fails to discriminate between MPPs and specific types of MAPs. In particular, *perturbations that affect sentence polarity*, *i.e.,* MAP1 (Removal of Negation Markers) and MAP7 (Replacing Words with Antonyms) result in a similar drop in the predicted scores as MPPs. An exception is a slight increase in the case of Nepali-English where the number of instances with negation markers were limited to only 4, which makes it impossible to draw any conclusions. Omitting negation is a critical error in the practical applications of MT. But it does not frequently occur in the data, and therefore, cannot be detected by using the standard way of assessing the performance of QE systems, *i.e.,* by computing the correlation with human judgements on a test set.

MAPs that correspond to *omission and addition errors in translation* (MAP2 and MAP4, respectively) also result in a relatively small drop in the predicted scores and thus hardly be distinguished from MPPs. Omitting contents is a well-known issue for the current neural MT models (Yang et al., 2019). An omission is particularly dangerous as it can go unnoticed by the end-user of the MT system. The ability to detect such errors is thus a crucial task for QE and, as highlighted by our analysis, requires further work in this direction.

Finally, *copying the source sentence in the translation* (MAP8) is not adequately captured by MonoTQ. Note that this represents another critical translation error, as the source sentence is left untranslated. We hypothesise that the inability to detect copy errors is due to the fact that MonoTQ relies on the multilingual pre-trained representations and, unless presented with such cases during fine-tuning, would treat the two sentences in the source language as equivalent.

Comparison across languages Interestingly, we observe similar trends across language pairs. For all the language pairs, sentences with MAP produce a larger drop in performance than MPP, and the same MAPs result in incorrect behaviour.

6.2 Do perturbations affect other QE models?

Table 4 shows the actual average scores produced by different QE systems for the initial subset of high-quality MT sentences (column MT) and the same subset of sentences perturbed using MPP (column MPP) and using (column MAP). For strong QE models, we would expect both MPP and MAP scores to be lower than the initial MT outputs, especially for MAP. For most of the models and languages, the sentences perturbed with MAP receive lower average scores, thus confirming that,

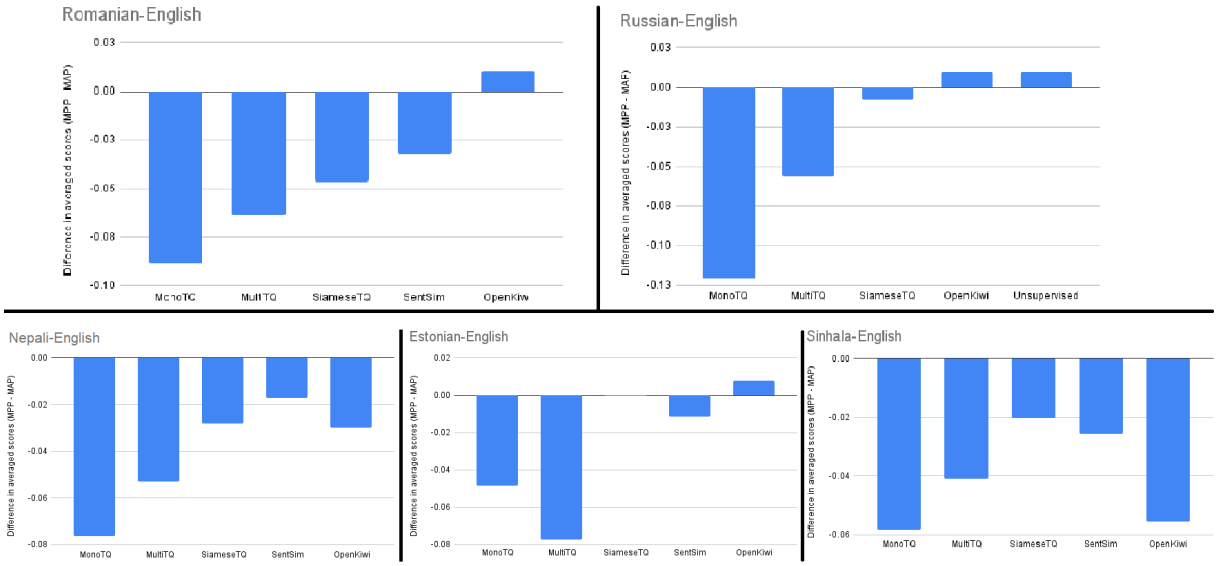


Figure 2: Ranking QE models using our method (MPP - MAP), where different QE models are shown on the X-axis, sorted as per the ranks obtained via Pearson correlation (among QE scores and human DA judgements). The size of the bars corresponds to the ability of the QE models to distinguish between MAP and MPP perturbations - the higher the negative bar, the better the QE model is at this task.

in general, QE models are sensitive to the changes that affect meaning. It is clear, however, that for some models, the difference between the MT, MAP and MPP is negligible. These cases are observed with OpenKiwi and SentSim, which are weaker QE models compared to the TransQuest variants (Specia et al., 2020) (see Table 5 for the overall results on the complete test+validation set of 2K sentences). Thus, we hypothesise that the ability of a QE model to discriminate between MAP and MPP could be predictive of its overall performance. We empirically test this hypothesis, and discuss below.

6.3 Can we use perturbations to rank QE models?

We pose that the overall performance of a QE system can be predicted based on how well it is able to discriminate between meaning-preserving and meaning-altering perturbations. To test this claim, we contrast the ability of a set of QE systems to discriminate between MAP and MPP with their overall performance measured in terms of Pearson correlation with human judgements. Table 5 shows sentence-level Pearson correlation with human judgements on the WMT 2020 QE Shared Task test set for all the QE models and language pairs considered in our experiments. As shown in Table 5, QE models vary a lot in terms of overall performance, the weakest system being OpenKiwi and SentSim, and the strongest corresponding to the SOTA approaches based on XLM-Roberta. To assess the discriminative power of the models, we compute the average difference (MPP - MAP) between the relative scores obtained via our method (such as shown in Figure 1). In Figure 2, we sort all the probed QE models in the decreasing order, according to the correlation with human judgements on the x-axis, and plot the corresponding MAP/MPP difference on the y-axis.

	Et-En	Ru-En	Ro-En	Si-En	Ne-En
MonoTQ	0.72	**0.77**	**0.88**	**0.88**	**0.75**
MultiTQ	**0.76**	**0.77**	0.87	0.87	0.74
SiameseTQ	0.55	0.71	0.84	0.84	0.60
SentSim	0.53	0.46	0.77	0.77	0.56
OpenKiwi	0.47	0.59	0.68	0.36	0.39

Table 5: Pearson correlation with human judgements for all QE models on the original, complete test+validation (2K) set. This is the metric used to rank participating QE systems in the WMT 2020 QE Shared Task 1. As can be seen, MonoTQ and MultiTQ consistently outperform all other models, with OpenKiwi performing the poorest.

Interestingly, for most of the language pairs, we observe that the system rankings are similar or identical to the Pearson correlation-based rankings; indicating that the ability of the model to distinguish between the proposed types of perturbations is indeed indicative of its overall performance. One exception is the difference corresponding to the OpenKiwi system for Sinhala-English and Nepali-

English. We attribute this to the fact that, by difference from the SOTA QE models, OpenKiwi is good at capturing the copy errors (MAP8) for these languages. OpenKiwi uses different vocabularies for the source and target languages, and therefore, copying the source sentence results in unknown tokens on the target side, leading to a low predicted score. Another exception is Estonian-English, where the systems appear to be ranked differently based on correlation vs. MAP/MPP difference. We note, however, that even in this case, the two top-performing systems (MonoTQ and MultiTQ) are clearly distinguished from the low-performing ones (SentSim and OpenKiwi).

Although generating MAPs and MPPs requires some initial set of high-quality translations, this could be selected using reference sentences from parallel data. Therefore, the proposed methodology allows for assessing the performance of QE models with no need for collecting explicit human judgements (*e.g.,* direct assessments).

7 Conclusions

In this work, we have proposed a methodology for analysing the performance of QE systems beyond correlation with human judgements. We have devised a set of perturbations to probe both the robustness of QE models towards changes in the input that do not affect sentence meaning and their sensitivity to meaning errors in translation. First, by applying the proposed methodology to a set of QE systems of varying accuracy, we are able to detect specific failures that cannot be detected by computing correlations between predicted scores and human judgements. Second, we have shown that, on an average, the ability of a given model to discriminate between the two types of perturbations is predictive of its overall performance, thus allowing us to compare QE systems without relying on manual quality annotation.

Our choice of specific perturbations was motivated by the errors that occur in neural MT and the potential weaknesses of QE models. In the future, we plan to extend this set by including perturbations that capture other critical MT errors. Furthermore, we plan to study whether the proposed perturbations can be used at training time to improve the ability of QE systems to detect critical errors in translation.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised Quality Estimation for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An Open Source Framework for Quality Estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. 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.

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*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. 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 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the First Shared Task on Machine Translation Robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy. Association for Computational Linguistics.

Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Peyman Passban, Puneeth S. M. Saladi, and Qun Liu. 2020. Revisiting Robust Neural Machine Translation: A Transformer Case Study. *arXiv preprint arXiv:2012.15710*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation Quality Estimation with Cross-lingual Transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically Equivalent Adversarial Rules for Debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Yurun Song, Junchen Zhao, and Lucia Specia. 2021. SentSim: Crosslingual Semantic Evaluation of Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3143–3156, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 Shared Task on Quality Estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Shuo Sun, Francisco Guzmán, and Lucia Specia. 2020. Are we Estimating or Guesstimating Translation Quality? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6262–6267, Online. Association for Computational Linguistics.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing Word Omission Errors in Neural Machine Translation: A Contrastive Learning Approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

A Appendix

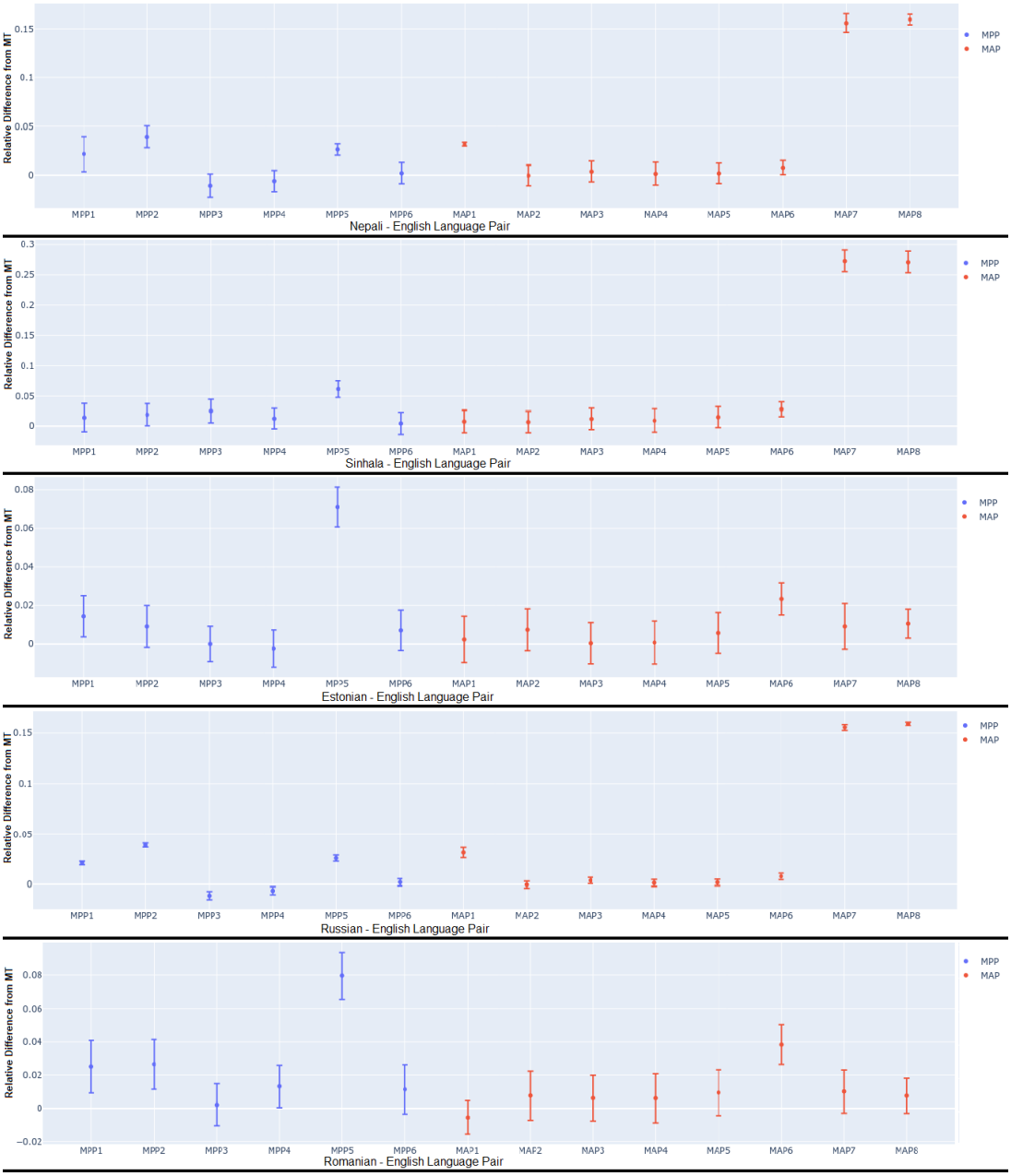


Figure 3: Difference between the predicted QE scores for original sentences and each perturbation for all language pairs (Y-axis->MT - x, where x is the perturbation as labelled on the X-axis), using the ***OpenKiwi*** system.

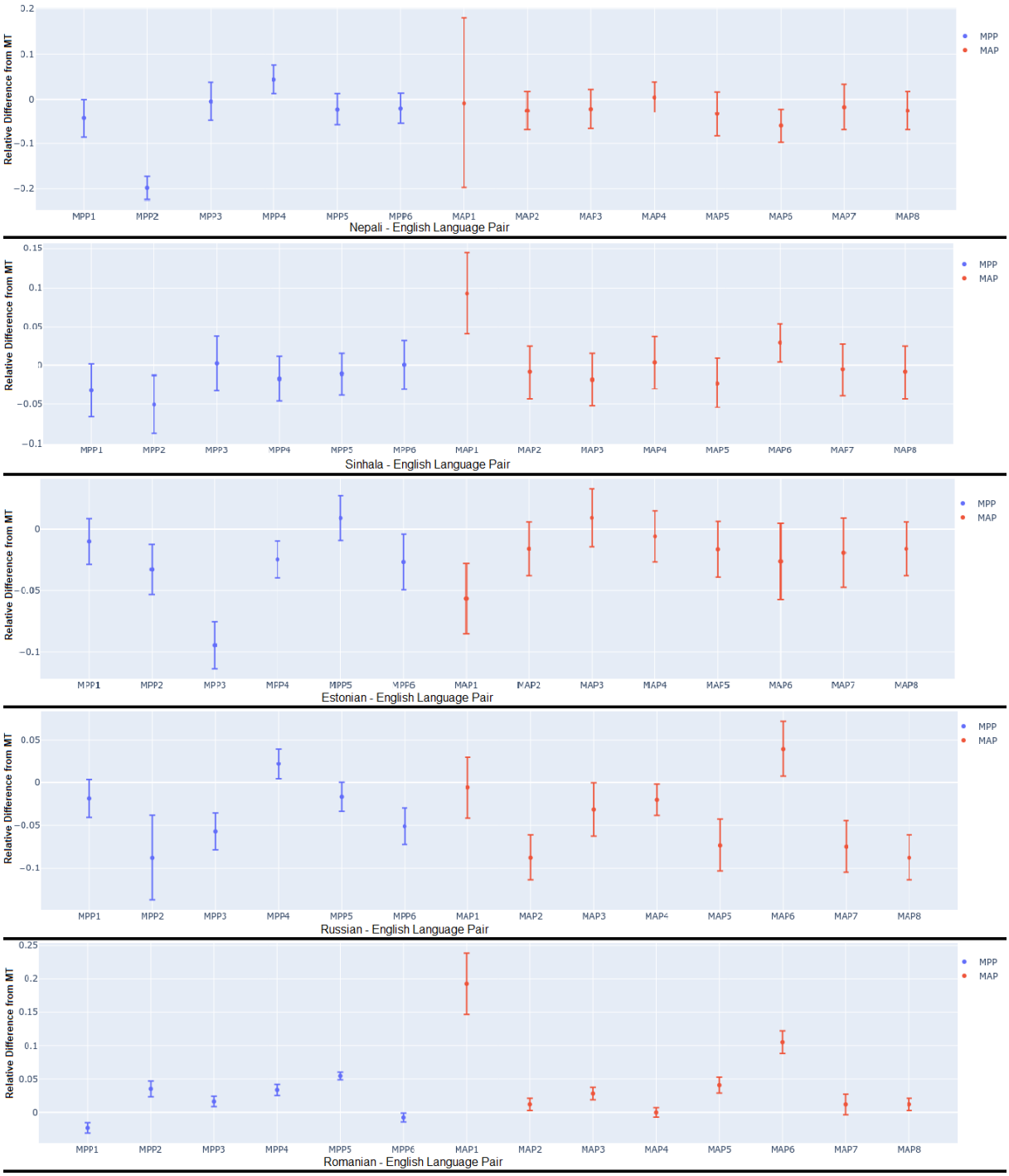


Figure 4: Difference between the predicted QE scores for original sentences and each perturbation for all language pairs (Y-axis->MT - x, where x is the perturbation as labelled on the X-axis), using ***Unsupervised SentSim*** method.

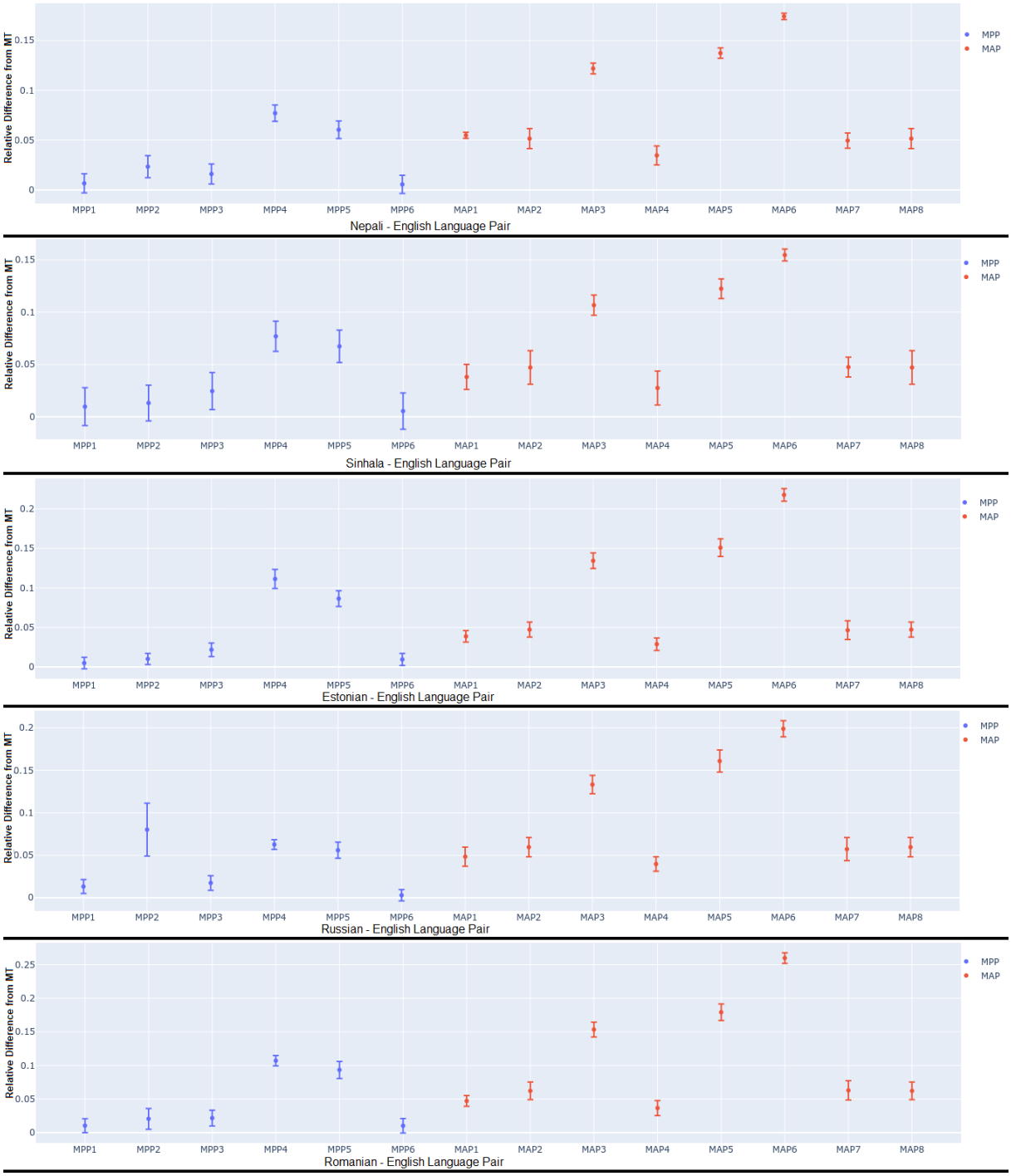


Figure 5: Difference between the predicted QE scores for original sentences and each perturbation for all language pairs (Y-axis->MT - x, where x is the perturbation as labelled on the X-axis), using ***MultilingualTransQuest*** architecture.

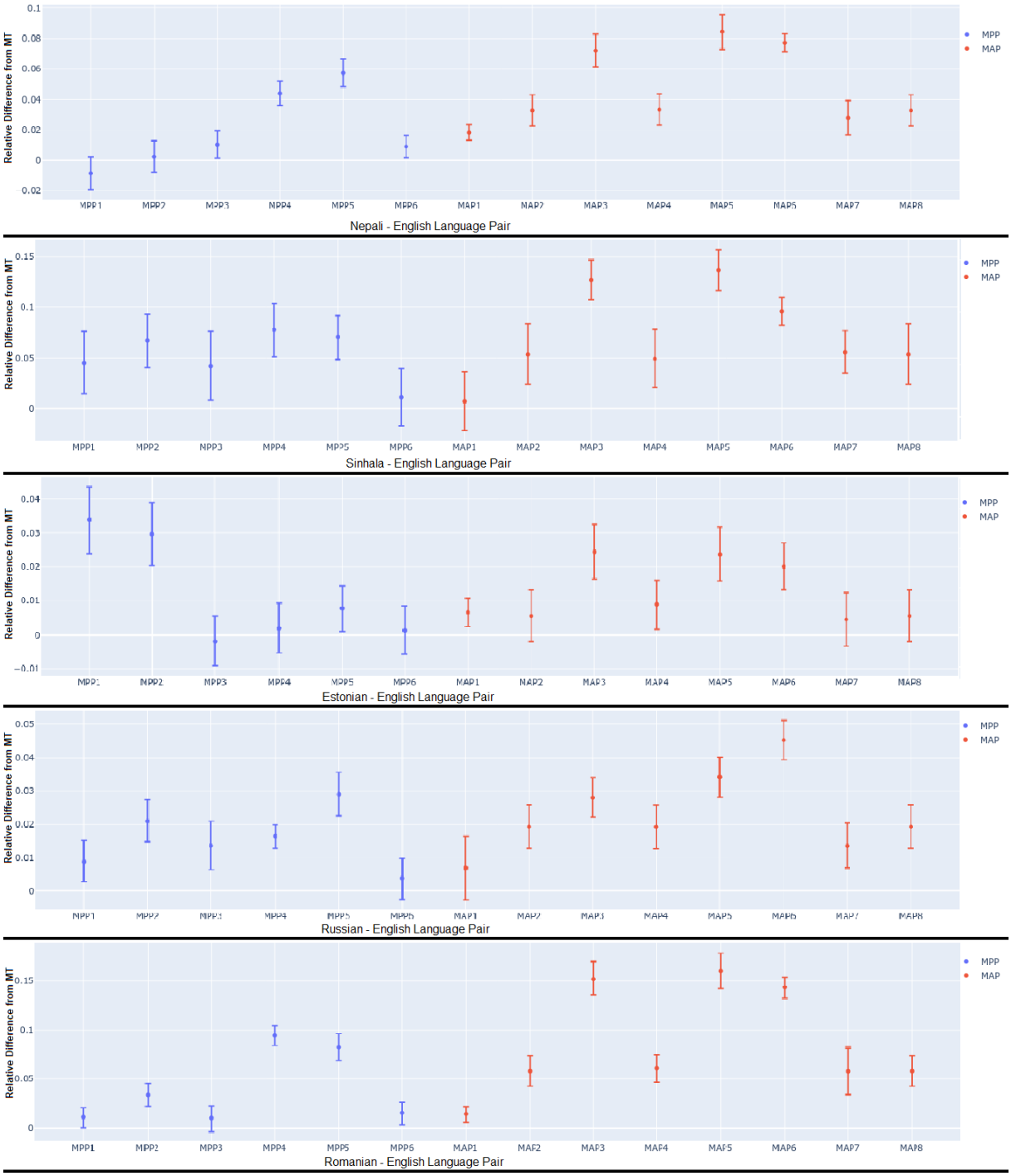


Figure 6: Difference between the predicted QE scores for original sentences and each perturbation for all language pairs (Y-axis->MT - x, where x is the perturbation on the X-axis), using the ***SiameseTransQuest*** architecture.

Findings of the WMT 2021 Shared Task on Efficient Translation

Kenneth Heafield[†] **Qianqian Zhu**[†] **Roman Grundkiewicz**[†§]
[†]University of Edinburgh
10 Crichton Street
Edinburgh, Scotland EH8 9AB
[§]Microsoft
1 Microsoft Way
Redmond, WA 98052, USA
{Kenneth.Heafield,Qianqian.Zhu,rgrundki}@ed.ac.uk

Abstract

The machine translation efficiency task challenges participants to make their systems faster and smaller with minimal impact on translation quality. How much quality to sacrifice for efficiency depends upon the application, so participants were encouraged to make multiple submissions covering the space of trade-offs. In total, there were 53 submissions by 4 teams. There were GPU, single-core CPU, and multi-core CPU hardware tracks as well as batched throughput or single-sentence latency conditions. Submissions showed hundreds of millions of words can be translated for a dollar, average latency is 5–20 ms, and models fit in 7.5–150 MB.

1 Introduction

The efficiency task complements the collocated news task by challenging participants to make their machine translation systems computationally efficient. This is the fourth edition of the task, expanding upon previous editions (Heafield et al., 2020; Hayashi et al., 2019; Birch et al., 2018).

Participants built English→German machine translation systems following the constrained data condition of the 2021 Workshop on Machine Translation news translation task. For translation quality measurement, we use the same news-focused WMT21 test set and human evaluation protocol as the news task. However, human assessment was conducted separately from the evaluation of the news task submissions.

Submissions are made as Docker containers so we can consistently measure their performance in terms of quality, speed, memory usage, and disk space. We run the containers in three different hardware environments: one GPU, one CPU core, and multiple CPU cores. Systems were tested for throughput by providing 1 million sentences upfront to allow batching and parallelization. We also tested for latency with a program that drip-feeds one input sentence, waits for the translation, and then provides the next input sentence. There were five conditions in total: GPU Batch (for throughput), GPU Latency, 1 CPU Core Batch, 1 CPU Core Latency, and 36 CPU cores Batch. We did not measure latency in a multi-core CPU setting because the test hardware has 36 cores and overhead for 36 threads is larger than the cost of arithmetic for the small tensors in optimized models.

	Edinburgh	HuaweiTSC	NiuTrans	TenTrans
GPU Batch	✓		✓	✓
GPU Latency	✓			
1 Core Batch	✓			
1 Core Latency	✓	✓		
36 Cores Batch	✓		✓	

Table 1: Participation in each of the hardware and batching conditions. Core refers to CPU hardware with 1 core or all 36 cores.

Participants were free to choose which conditions to participate in. The condition was passed to the Docker container as command line arguments. Table 1 shows the four participants and the conditions they submitted to.

Machine translation is used in a range of settings where users might choose different trade-offs between quality and efficiency. For example, a high-frequency trading system might prefer the lowest latency at the expense of quality given that the output will only be read by a machine. Conversely, in a post-editing scenario the personnel costs outweigh many computational costs. Therefore there is not a single best system, but a range of options that trade between quality and efficiency. We emphasize the Pareto frontier: the fastest systems at each level of quality, or the smallest systems at each level of quality. To explore the Pareto frontier, participants were encouraged to make multiple submissions covering the range of trade-offs. In total, 53 combinations of models, hardware, and batching were benchmarked.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 639–651
November 10–11, 2021.

2 Hardware

We chose modern hardware to encourage exploiting new hardware features. The GPU is an NVidia A100 from the Oracle Cloud `BM.GPU4.8` instance. The instance has eight GPUs and we limited Docker to using only one GPU. The GPU machine has an AMD EPYC 7542 CPU with all cores allowed. In practice, most submissions used only one core while NiuTrans's submissions used the CPU cores to parallelize preprocessing and postprocessing.

The CPU-only condition used a dual-socket Intel Xeon Gold 6354 from Oracle Cloud `BM.Optimized3.36` with a total of 36 cores. For the single-core CPU track, we reserved the entire machine then ran Docker with `-cpuset-cpus=0`. In the 36-core CPU track, participants were free to configure their own CPU sets and affinities.

The Oracle Cloud machines are bare metal servers, meaning there was no shared tenancy, no virtualization, and the test machines were otherwise quiescent.

3 Input Text

To amoritize loading time, avoid starving highly parallel submissions, and reduce the ability to cheat, we benchmark systems on 1 million sentences of input. The test set is hidden inside these 1 million sentences, shuffled with filler sentences. Many filler sentences are drawn from parallel corpora to check that systems are in fact translating all sentences, though we do not consider scores on noisy corpora reliable enough to report. The composition of this set changes each year and is decided after the submission deadline.

Filler data was gathered from parallel corpora and gender bias challenge sets: WMT news test sets from 2008 through 2021 (Barrault et al., 2020), the additional test inputs in WMT 2021, Khresmoi summary test v2 (Dušek et al., 2017), IWSLT 2019 (Jan et al., 2019), SimpleGen (Renduchintala et al., 2021), WinoMT (Stanovsky et al., 2019), TED 2020 (Reimers and Gurevych, 2020), and Tilde RAPID 2019 (Rozis and Skadiņš, 2017). We capped sentence lengths at 150 space-separated tokens, except for the WMT 2021 test set to preseve the ability to evaluate with it. Because WMT 2020 includes excessively long segments that are actually concatenated sentences, we also added sentence split versions of WMT 2020 and WMT 2021, though the difference on WMT 2021 was minor. Source sentences were concatenated, deduplicated, and shuffled. The Tilde RAPID corpus was clipped to make a total of 1 million deduplicated lines. Counts are shown in Table 2.

Corpus	Sentences
WMT 08–19	32,477
WMT 20 under 150 tokens	1,416
WMT 20 sentence split	2,048
WMT 21 sentence split	1,096
WMT 21 including additional tests	14,938
Khresmoi Summary Test v2	1,000
IWSLT 2019	2,278
SimpleGen	2,664
WinoMT	3,888
TED 2020 v1	293,562
Tilde RAPID 2019	654,995
Total	1,010,362
Deduplicated	1,000,000

Table 2: Corpora used for input text.

Input text and tools to extract test sets from system outputs are available at `http://data.statmt.org/heafield/wmt21-testdata.tar.xz`.

The input file has 1,000,000 lines, 19,951,184 space-separated words, and 124,257,215 bytes (most of which are characters since the file is English in UTF-8). This is an average of 20 words per sentence compared to 15 words per sentence the previous year (Heafield et al., 2020) due to raising the cap from 100 to 150 tokens per sentence and the lengthy text in the RAPID corpus.

Teams were responsible for their own tokenization and detokenization. We provided raw UTF-8 English input text with one sentence per line.

4 Metrics

4.1 Cost

Time was measured with wall (real) time reported by `time` and CPU time reported by the kernel for the process group. We do not measure loading time because it is small compared to translating 1 million sentences, some tools load lazily, and it is easily gamed by padding loading time.

Peak RAM consumption was measured using `memory.max_usage` in bytes from the kernel for the CPU and by polling `nvidia-smi` for the GPU. Swap was disabled.

Participants were told to separate their Docker

	Edinburgh	HuaweiTSC	NiuTrans	TenTrans
GPU Batch	3/10		4/4	4/4
GPU Latency	0/11			
1 Core Batch	0/6			
1 Core Latency	3/6	4/4		
36 Cores Batch	0/6		0/2	
Total	6/39	4/4	4/6	4/4

Table 3: Number of submissions by participant and condition (cores refers to the CPU hardware). The number after / is all submissions by the participant. The number before / is how many participants selected for focused human evaluation based on automatic metrics.

images into model and code files so that models could be measured separately from the relatively noisy size of code and libraries. A model was defined as "everything derived from data: all model parameters, vocabulary files, BPE configuration if applicable, quantization parameters or lookup tables where applicable, and hyperparameters like embedding sizes." Code could include "simple rule-based tokenizer scripts and hard-coded model structure that could plausibly be used for another language pair." They were also permitted to use standard compression tools such as `xz` to compress models; decompression time was included in results but small relative to the cost of translation. We report size of the model directory captured before the model ran. We also measured the total size of the Docker image (after compressing with `xz`), though participants were encouraged to prioritize shipping one container for multiple hardware conditions over the size of the container.

4.2 Quality

Translation quality is measured on the WMT 2021 news test set. The automatic metrics are COMET (Rei et al., 2020) `wmt20-comet-da` from version `1.0.0rc6`, BLEU from sacrebleu (Post, 2018) `nrefs:3|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0`, and chrF also from sacrebleu. We use references A, C, and D because the organizers found postedited DeepL output in reference B. COMET does not natively support multiple references so we averaged as recommended by the authors.[1] We also averaged chrF across references. Results were presented to participants[2] who were encouraged to whittle down systems for a focused human evaluation. HuaweiTSC and TenTrans included all of their submissions. NiuTrans included their GPU submissions but not their CPU submissions that have lower automatic scores than Edinburgh's. This left GPU Batch and 1 Core Latency as the only conditions with multiple teams. Edinburgh kept systems that have competitors and are near the Pareto frontier. The number of submissions evaluated is shown in Table 3. Out of 53 submissions, we ran direct assessment on 18.

For human evaluation, as a source of the absolute quality measure we used document-level source-based direct assessments (DA) (Graham et al., 2013; Cettolo et al., 2017) following the procedure established at the WMT20 News Translation Task (Barrault et al., 2020). We also conducted contrastive evaluation using segment-level pairwise direct assessments (Novikova et al., 2018; Sakaguchi and Van Durme, 2018), because it can be a better discriminative tool for measuring relative quality difference between pairs of systems. We compared the 18 systems using source-based direct assessment and 58 pairs of systems with contrastive direct assessment. In total, we gathered 21,487 and 20,416 direct assessment scores in standard and contrastive campaigns respectively. All annotations were made by bilingual native German speakers with a translation or linguistics background. Annotations were collected using Appraise[3] (Federmann, 2018).

5 Results

All submissions are shown in Table 4. Source-based direct assessment scores appear for the submissions in the focused human evaluation with the number of wins against other systems (including those in other conditions), raw direct accessment score, and z-score after standardizing annotator scores to mitigate differences in annotator scores. Scores were averaged ("Ave.") across sentences. Rows are sorted by COMET because only some submissions have human assessment.

The system ranking based on the standard DA is presented in Table 5. Systems are ordered by the number of respective wins against other systems and average DA z-score. Ordering solely by z-scores would produce three clusters with all systems within a cluster considered tied according to Wilcoxon rank-sum test with $p < 0.05$.

[1] https://github.com/Unbabel/COMET/issues/20

[2] Only reference A was available at the time.

[3] https://github.com/AppraiseDev/Appraise

NVIDIA A100 GPU Batch													
		Human			Automatic			Seconds		Disk MB		RAM MB	
Team	Variant	Win	Ave.	Ave. z	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU	GPU
Edinburgh	base	17	90.3	0.352	0.527	55.25	61.54	140	152	150	455	1725	36140
Edinburgh	tiny11	14	85.9	0.185	0.492	52.74	60.52	115	120	60	364	1622	36092
Edinburgh	2.12-2.tied.tiny.heads-0.3				0.473	52.36	60.32	126	130	59	363	1618	36090
Edinburgh	2.6-2.tied.tiny.heads-0.3				0.459	51.52	60.00	116	120	53	357	1611	36088
Edinburgh	2.12_1.tiny.heads-0.3				0.445	52.20	60.25	117	121	62	366	1620	36092
Edinburgh	2.12_1.micro.heads-0.3				0.440	51.73	60.02	117	121	60	364	1617	36092
Edinburgh	2.8-4.tied.tiny.4bit				0.432	50.20	59.47	140	144	8	355	1639	29054
NiuTrans	6_1_512	9	83.5	0.057	0.423	50.05	59.96	95	377	73	303	2447	4254
NiuTrans	12_1_512	4	88.8	0.016	0.422	50.50	59.83	124	411	109	335	2458	4356
Edinburgh	2.12_1.tiny.4bit	6	85.6	0.104	0.422	51.78	59.86	118	122	10	357	1659	29062
NiuTrans	6_1_0	4	80.4	-0.019	0.384	49.78	59.71	94	400	72	302	2467	3998
Edinburgh	3.12_1.micro				0.382	50.40	59.29	116	121	66	370	1627	36094
NiuTrans	3_1_512	3	85.6	-0.035	0.354	48.72	59.25	81	380	55	287	2475	4134
Edinburgh	2.12_1.micro.rowcol-0.5				0.352	48.73	58.59	107	110	42	346	1603	36082
TenTrans	tea-20_6-h512-ffn4096	3	81.1	-0.046	0.335	46.26	57.19	456	638	643	1804	2380	25318
TenTrans	stu-20_1-h512-ffn2048	2	81.8	-0.104	0.291	45.89	57.06	340	528	355	1272	2120	17126
TenTrans	stu-10_1-h512-ffn2048	2	82.5	-0.138	0.263	44.88	56.89	280	458	234	1049	2006	17128
TenTrans	stu-20_1-h256-ffn1024	2	84.3	-0.091	0.238	44.34	56.68	257	443	114	829	1864	17126
NVIDIA A100 GPU Latency													
		Human			Automatic			Seconds		Disk MB		RAM MB	
Team	Variant	Win	Ave.	Ave. z	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU	GPU
Edinburgh	base				0.527	55.25	61.54	16851	16859	150	455	1573	36140
Edinburgh	tiny11				0.491	52.80	60.55	15101	15102	60	364	1247	36092
Edinburgh	2.12_1.base.4bit				0.476	53.81	60.86	15239	15243	22	369	1653	38174
Edinburgh	2.12-2.tied.tiny.heads-0.3				0.473	52.39	60.32	18269	18271	59	363	1233	36090
Edinburgh	2.6-2.tied.tiny.heads-0.3				0.460	51.60	59.99	17204	17205	53	357	1216	36088
Edinburgh	2.12_1.tiny.heads-0.3				0.445	52.13	60.22	13839	13841	62	366	1241	36092
Edinburgh	2.12_1.micro.heads-0.3				0.436	51.66	59.99	13952	13952	60	364	1236	36092
Edinburgh	2.8-4.tied.tiny.4bit				0.431	50.26	59.49	26635	26637	8	355	1264	29054
Edinburgh	2.12_1.tiny.4bit				0.419	51.79	59.87	13876	13878	10	357	1299	29062
Edinburgh	3.12_1.micro				0.379	50.40	59.34	13944	13945	66	370	1251	36094
Edinburgh	2.12_1.micro.rowcol-0.5				0.352	48.73	58.61	13665	13665	42	346	1184	36082
1 Core Ice Lake CPU Batch													
		Human			Automatic			Seconds		Disk MB		RAM MB	
Team	Variant	Win	Ave.	Ave. z	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU	
Edinburgh	base				0.520	54.72	61.36	11067	11066	45	63	1569	
Edinburgh	3.12_1.large				0.485	53.71	60.89	30342	30338	129	386	2428	
Edinburgh	tiny11				0.464	52.24	60.17	5108	5107	21	468	621	
Edinburgh	4.12_1.tiny.rowcol-0.5.ft8				0.328	48.34	58.33	3288	3287	52	302	1040	
Edinburgh	4.12_1.micro.rowcol-0.5.ft8				0.326	48.97	58.41	3497	3497	17	302	912	
Edinburgh	4.12_1.micro.rowcol-0.5				0.318	47.66	58.01	4046	4045	53	338	781	
1 Core Ice Lake CPU Latency													
		Human			Automatic			Seconds		Disk MB		RAM MB	
Team	Variant	Win	Ave.	Ave. z	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU	
Edinburgh	base	13	88.3	0.205	0.465	53.53	60.69	16815	16814	45	63	542	
HuaweiTSC	base	7	90.3	-0.019	0.450	53.00	60.82	14939	14937	37	53	377	
Edinburgh	3.12_1.large				0.430	52.95	60.34	40518	40514	129	386	1175	
Edinburgh	tiny11	3	81.4	-0.008	0.413	51.18	59.63	9272	9272	21	468	241	
HuaweiTSC	sm9	4	86.1	-0.001	0.391	50.58	59.74	8866	8865	20	36	206	
HuaweiTSC	sm6	2	77.5	-0.025	0.338	48.75	58.85	7714	7713	17	33	173	
Edinburgh	4.12_1.micro.rowcol-0.5	0	84.0	-0.444	0.257	47.56	57.88	6343	6343	53	338	342	
HuaweiTSC	tiny	0	81.9	-0.363	0.197	44.20	56.84	5138	5138	10	27	107	
Edinburgh	4.12_1.tiny.rowcol-0.5.ft8				-0.073	37.43	56.33	8148	8147	52	302	335	
Edinburgh	4.12_1.micro.rowcol-0.5.ft8				-0.173	37.67	55.71	7564	7563	17	302	239	
36 Core Ice Lake CPU Batch													
		Human			Automatic			Seconds		Disk MB		RAM MB	
Team	Variant	Win	Ave.	Ave. z	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU	
Edinburgh	base				0.519	54.69	61.35	500	17790	45	63	28630	
Edinburgh	3.12_1.large				0.484	54.02	60.92	1509	53528	129	386	34903	
Edinburgh	tiny11				0.465	52.17	60.16	237	8434	21	468	15594	
NiuTrans	6_1_512				0.430	50.08	60.02	520	36015	146	142	57636	
NiuTrans	3_1_512				0.358	48.53	59.34	417	28727	109	126	56415	
Edinburgh	4.12_1.tiny.rowcol-0.5.ft8				0.336	48.38	58.37	159	5682	52	302	18606	
Edinburgh	4.12_1.micro.rowcol-0.5.ft8				0.329	48.95	58.42	167	5948	17	302	15825	
Edinburgh	4.12_1.micro.rowcol-0.5				0.318	47.98	58.16	184	6540	53	338	16469	

Table 4: All submissions. Human source-based DA is shown for selected submissions. Total time measured in seconds is equivalent to microseconds/sentence because the input is 1 million sentences.

Team	Variant	Win	Ave.	Ave. z	Time (s)	Condition
Edinburgh	base	17	90.3	0.352	140	GPU Batch
Edinburgh	tiny11	14	85.9	0.185	115	GPU Batch
Edinburgh	base	13	88.3	0.205	16815	1 Core Latency
NiuTrans	6_1_512	9	83.5	0.057	95	GPU Batch
HuaweiTSC	base	7	90.3	-0.019	14939	1 Core Latency
Edinburgh	2.12_1.tiny.4bit	6	85.6	0.104	118	GPU Batch
NiuTrans	12_1_512	4	88.8	0.016	124	GPU Batch
HuaweiTSC	sm9	4	86.1	-0.001	8866	1 Core Latency
NiuTrans	6_1_0	4	80.4	-0.019	94	GPU Batch
Edinburgh	tiny11	3	81.4	-0.008	9272	1 Core Latency
NiuTrans	3_1_512	3	85.6	-0.035	81	GPU Batch
TenTrans	tea-20_6-h512-ffn4096	3	81.1	-0.046	456	GPU Batch
HuaweiTSC	sm6	2	77.5	-0.025	7714	1 Core Latency
TenTrans	stu-20_1-h256-ffn1024	2	84.3	-0.091	257	GPU Batch
TenTrans	stu-20_1-h512-ffn2048	2	81.8	-0.104	340	GPU Batch
TenTrans	stu-10_1-h512-ffn2048	2	82.5	-0.138	280	GPU Batch
HuaweiTSC	tiny	0	81.9	-0.363	5138	1 Core Latency
Edinburgh	4.12_1.micro.rowcol-0.5	0	84.0	-0.444	6343	1 Core Latency

Table 5: System ranking based on the standard direct assessment (DA) human evaluation. The rows are ordered by the number of respective wins against other systems, followed by the DA z-score. Systems within a cluster are considered tied according to Wilcoxon rank-sum test $p < 0.05$ with standard DA.

Figure 1 shows the trade-off between quality and speed of batched translation submissions. Since source-based DA is available for select GPU submissions, we include that comparison; the other plots rely on COMET to approximate quality. Each plot shows the Pareto frontier as a black staircase to highlight the best combinations of quality and speed. In Figure 2, we combine GPU and 36 Core CPU speed by using Oracle Cloud pricing. The GPU is cheaper for throughput-oriented tasks that allow batching.

Latency is shown in Figures 3 and 4. HuaweiTSC and Edinburgh were the two participants and shared the Pareto frontier. While the GPU is cheaper for throughput, both CPU and GPU entries appear on the Pareto frontier for latency. In fact, the lowest latencies are achieved by single-core CPU submissions, likely due to the overhead of launching small kernels on a GPU.

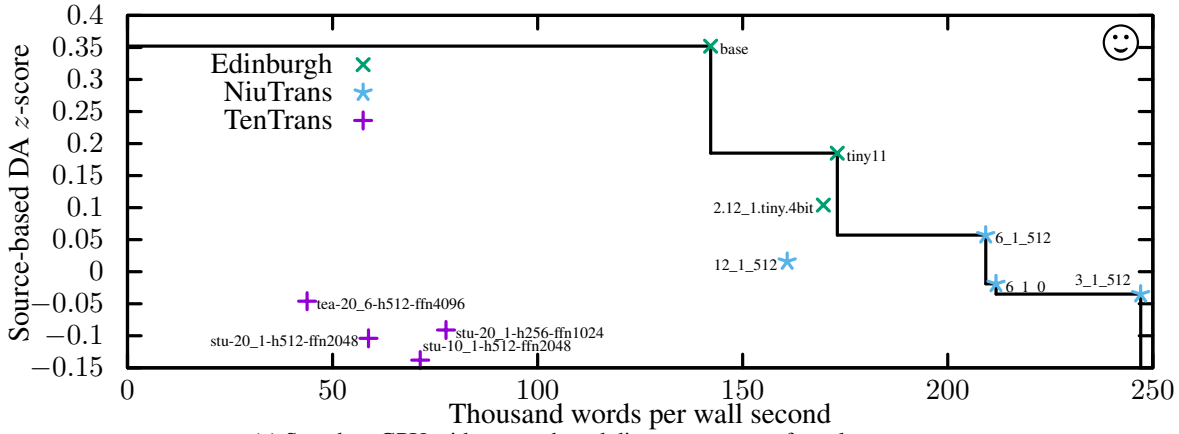
Model sizes at rest on disk appear in Figures 5 and 6. Participants were allowed to compress their models using their own tools and standard tools like `xz`. The entire Pareto frontier consists of Edinburgh submissions, resting partly on 4-bit integer compression. Docker image sizes, which include model and software, appear in Figure 7. HuaweiTSC optimized their image size well. Conversely, some others opted to optimize other metrics and included large Linux installations. We compressed all docker images with `xz` before measuring.

Memory (RAM) consumption appears in Figure 8. GPU memory consumption reflects batch size and some participants set a large batch size to maximize speed. Optimizing speed for multi-socket CPU machines implies having a copy of the model in RAM close to each socket, so memory consumption is larger beyond simply having temporary space for more batches. Finally, participants may have sorted the entire 118 MB input file in RAM to form batches of equal length sentences. NiuTrans is the clear winner on GPU RAM consumption and curiously the clear loser on CPU RAM consumption.

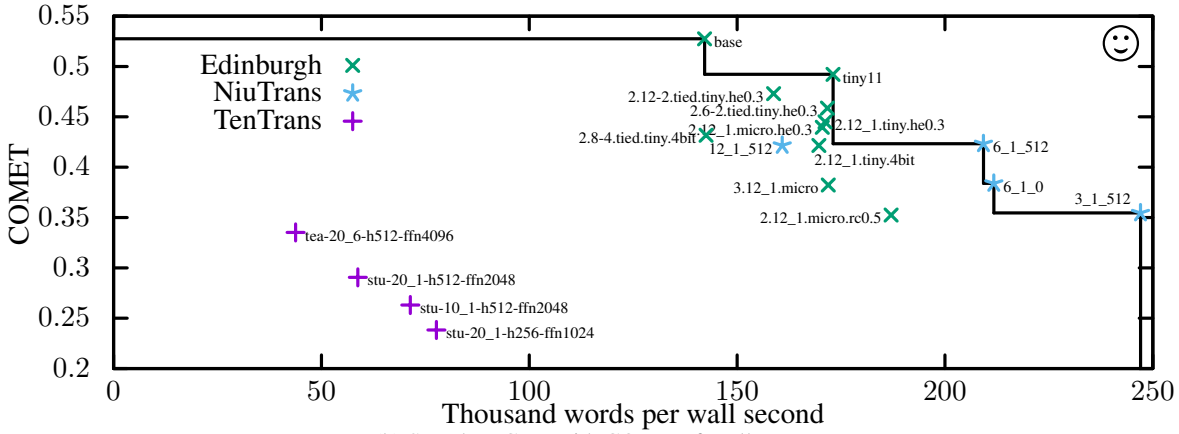
Many of the systems tied on standard DA and contrastive DA helps us pull them apart by directly comparing system outputs. Table 6 shows detailed results of contrastive DA including average scores, respective deltas between two systems and the outcome of significance testing. For groups of systems for which we evaluated each system from a group against each other system from the same group, we created separate rankings based solely on pairwise comparisons within the group, presented in Table 7.

6 Conclusion and Future Tasks

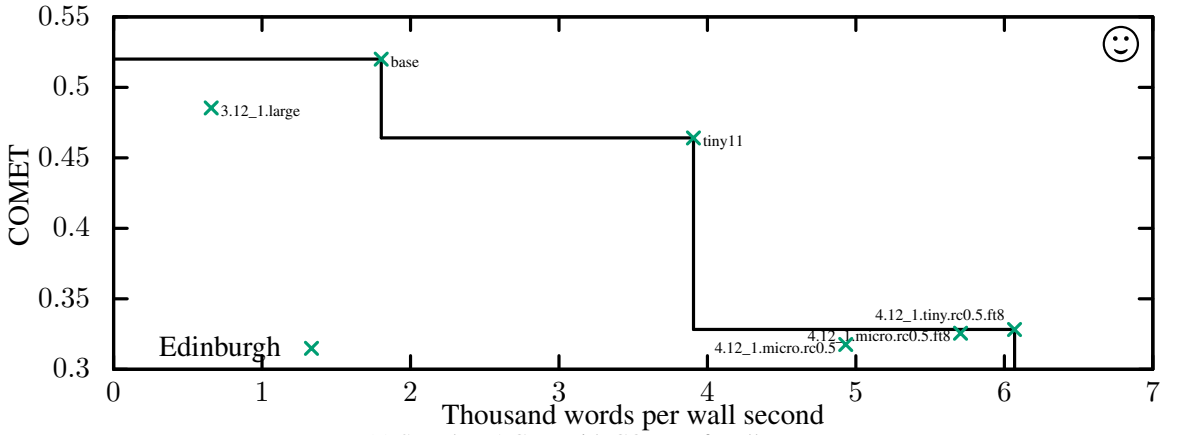
Using the highest quality system in this evaluation, translating 124,257,215 characters took 140 seconds on an A100 GPU that costs $3.05/hr in a cloud. That is $0.001/million characters. By comparison, Google Translate's cost is $20/million



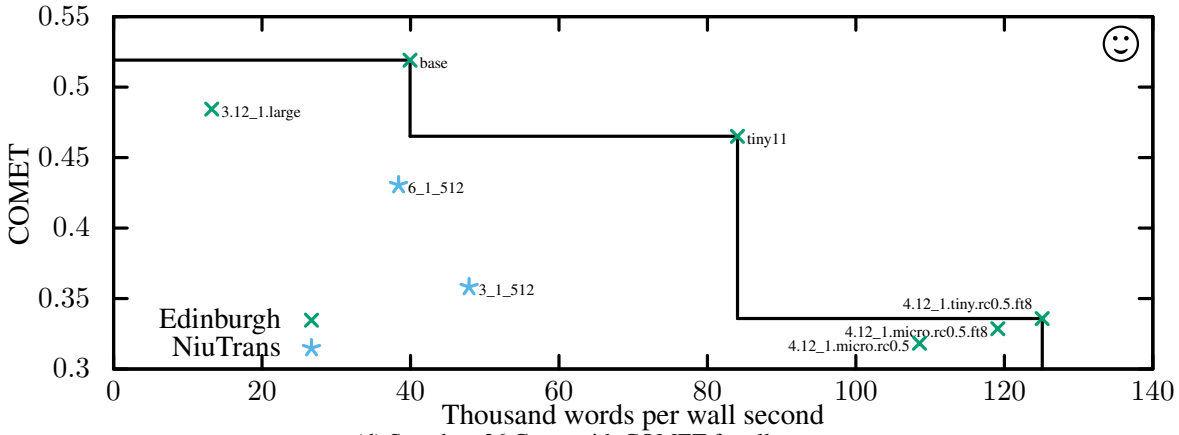
(a) Speed on GPU with source-based direct assessment for select systems



(b) Speed on GPU with COMET for all systems



(c) Speed on 1 Core with COMET for all systems



(d) Speed on 36 Cores with COMET for all systems

Figure 1: Speed and quality of batched submissions. The staircase shows the Pareto frontier.

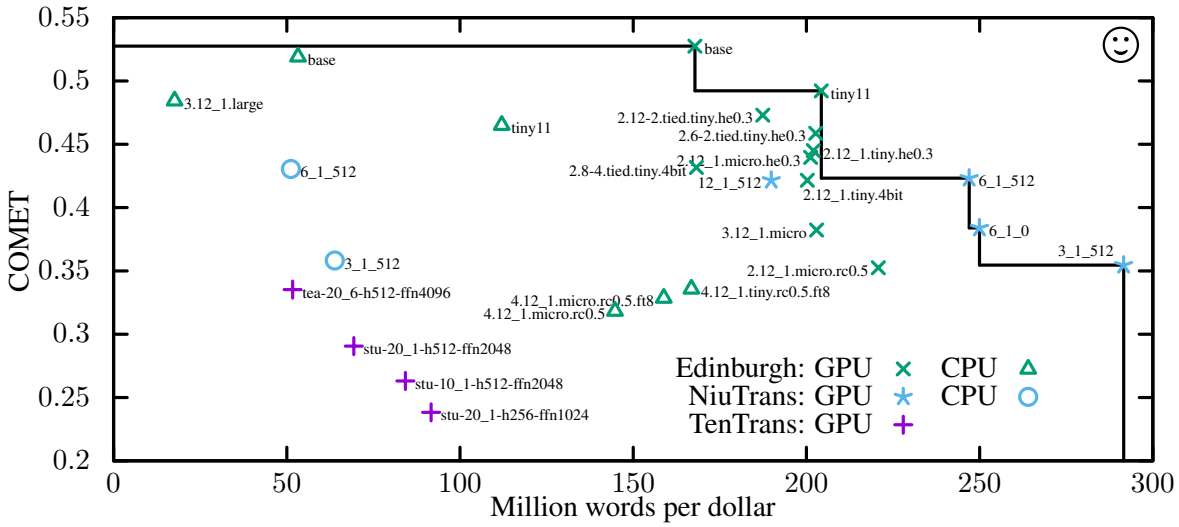


Figure 2: Cost of batched translation for an A100 GPU at $3.05/hr or 36 Cores of CPU at $2.7/hr on Oracle Cloud.

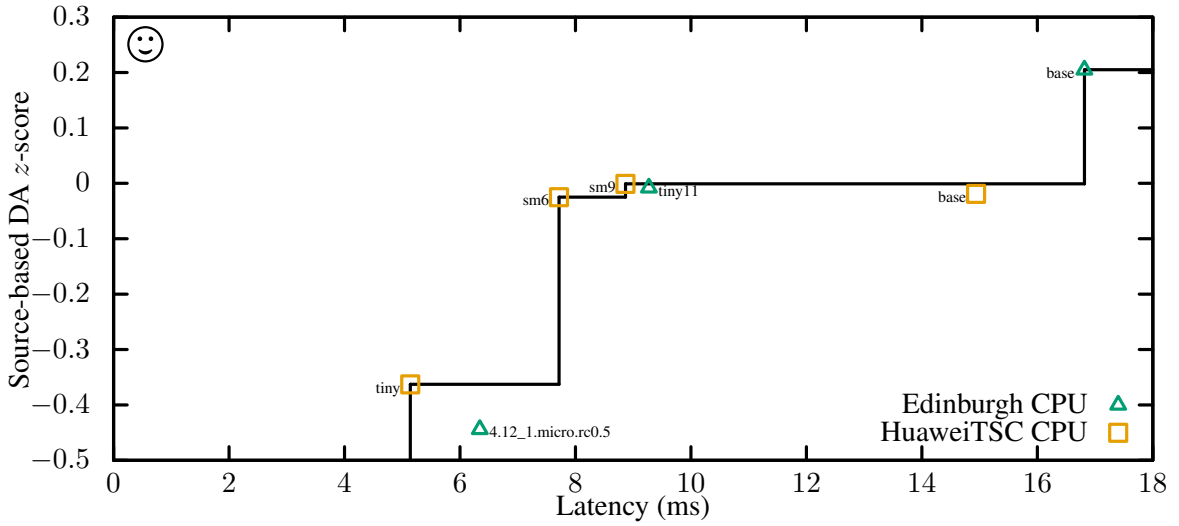


Figure 3: Latency of select CPU systems with source-based direct assessment. Contrastive direct assessment (Table 7) insignificantly ranked HuaweiTSC's base > Edinburgh's tiny11 > HuaweiTSC's sm9.

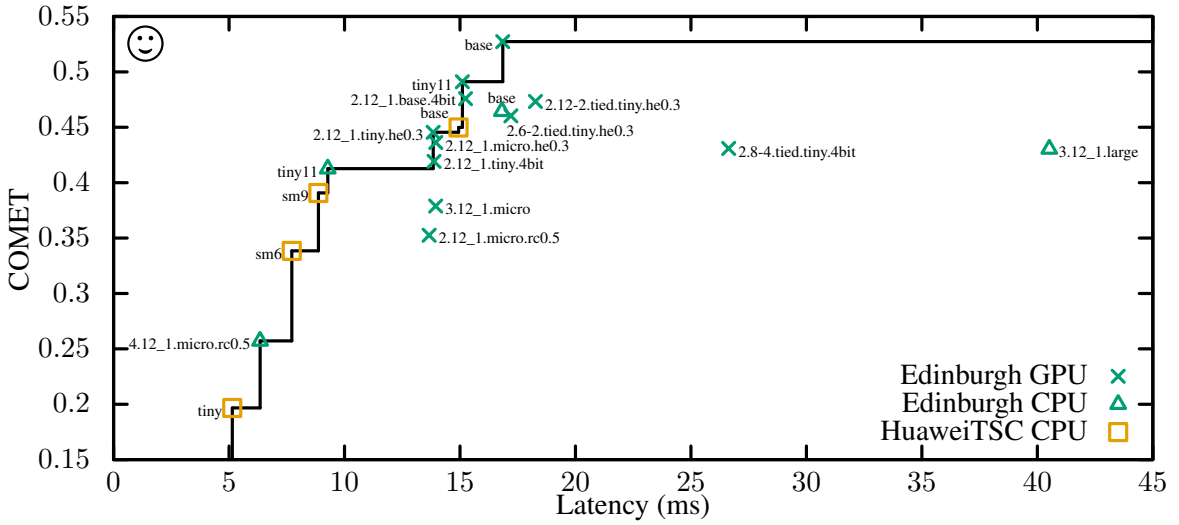


Figure 4: Latency of combined CPU and GPU systems with COMET scores. To improve the scale of the graph, low-quality variants 4.12_1.tiny.rowcol-0.5.ft8 and 4.12_1.micro.rowcol-0.5.ft8 from Edinburgh are not shown. Their respective COMET scores are -0.073 and -0.173.

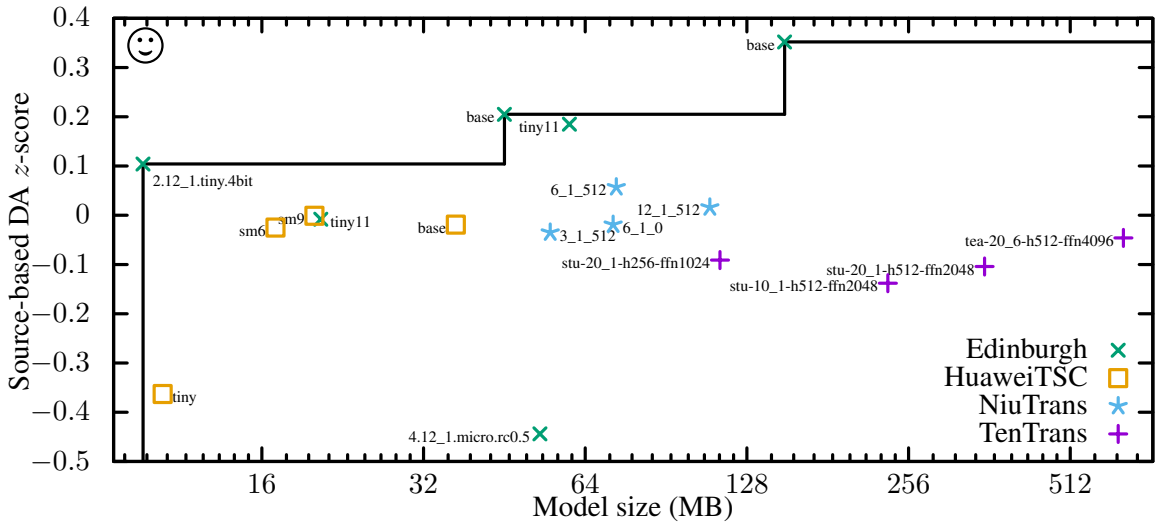


Figure 5: Model sizes of select systems in the human evaluation with source-based DA. Because selection for human evaluation focused on speed (and not model size), this is missing the smallest model, Edinburgh's 2.8-4.tied.tiny.4bit and a few other Pareto optimal systems identified by automatic metrics.

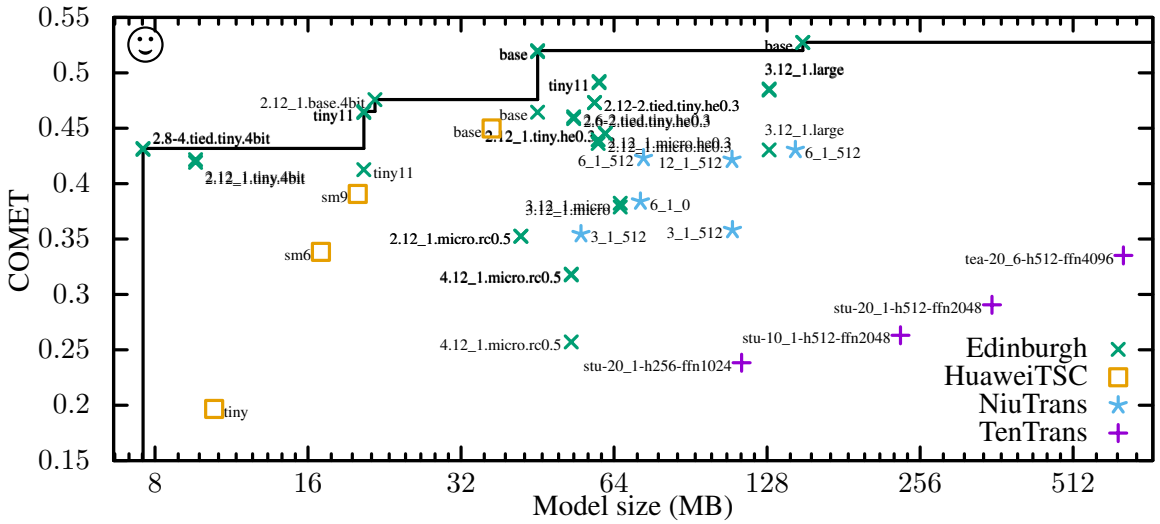


Figure 6: All model sizes with quality by COMET. Because models had slightly different output in different hardware conditions, the same variant label can appear multiple times like a shadow. Low-quality variants 4.12_1.tiny.rowcol-0.5.ft8 and 4.12_1.micro.rowcol-0.5.ft8 from Edinburgh are omitted for scale.

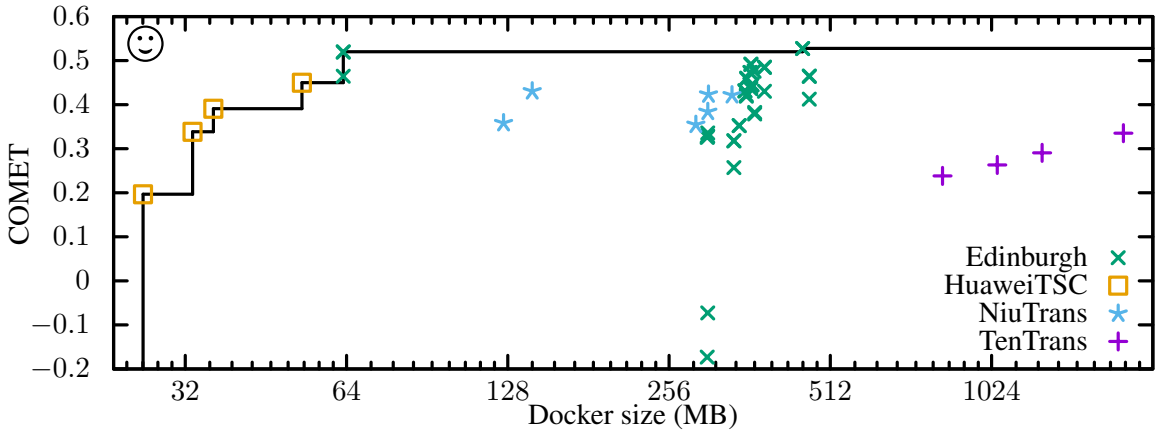
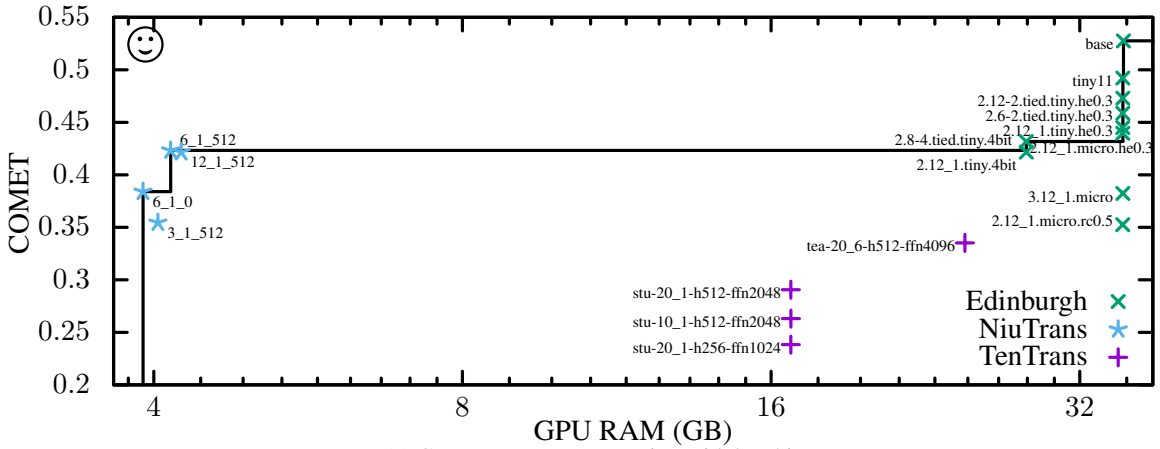
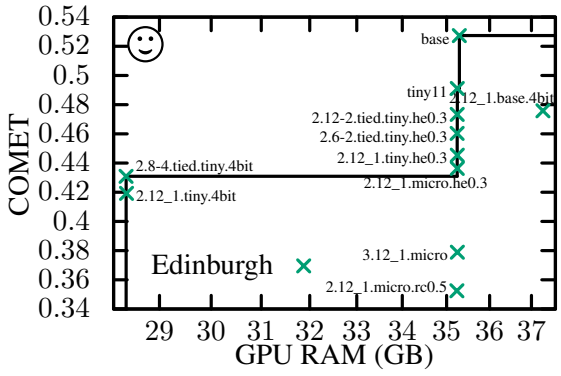


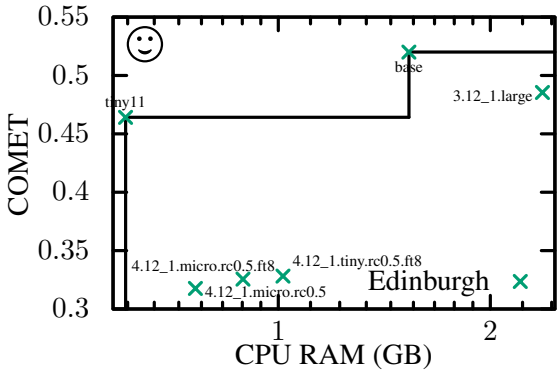
Figure 7: Size of all Docker images after compression with `xz` on a logarithmic scale. Some participants did not seek to prune image size and included large Linux installations. Labels are not shown due to crowding.



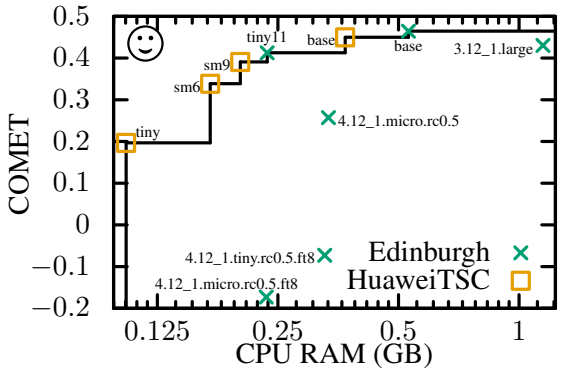
(a) GPU memory consumption with batching.



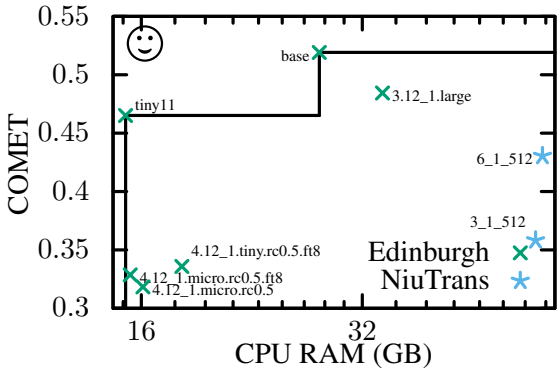
(b) GPU memory consumption with latency.



(c) 1 core CPU memory consumption with batching.



(d) 1 core CPU memory consumption with latency.



(e) 36 core CPU memory consumption with batching.

Figure 8: RAM consumption of all submissions on a logarithmic scale. Some participants used large batches to favor speed over memory consumption.

Stronger System			Weaker System			Stronger	Weaker		
Team	**Variant**	**Condition**	**Team**	**Variant**	**Condition**	**DA Score**	**DA Score**	**Delta**	p**-val**
Edinburgh	base	GPU Latency	Edinburgh	base	GPU Batch	92.2	92.2	0.0	
Edinburgh	base	GPU Batch	Edinburgh	tiny11	GPU Batch	74.9	74.7	0.2	
Edinburgh	tiny11	GPU Batch	Edinburgh	tiny11	GPU Latency	86.6	86.4	0.2	
Edinburgh	tiny11	GPU Batch	Edinburgh	2.12_1.tiny.4bit	GPU Batch	85.6	83.6	1.9	
NiuTrans	12_1_512	GPU Batch	NiuTrans	6_1_0	GPU Batch	78.1	75.2	2.8	**
NiuTrans	12_1_512	GPU Batch	NiuTrans	3_1_512	GPU Batch	67.7	65.1	2.6	*
NiuTrans	12_1_512	GPU Batch	NiuTrans	6_1_512	GPU Batch	65.9	65.1	0.8	
NiuTrans	6_1_0	GPU Batch	NiuTrans	3_1_512	GPU Batch	86.7	85.8	0.9	
NiuTrans	6_1_512	GPU Batch	NiuTrans	3_1_512	GPU Batch	79.7	78.2	1.4	
NiuTrans	6_1_512	GPU Batch	NiuTrans	6_1_0	GPU Batch	82.7	82.6	0.0	
TenTrans	stu-10_1-h512-ffn2048	GPU Batch	TenTrans	tea-20_6-h512-ffn4096	GPU Batch	85.4	83.3	2.1	
TenTrans	stu-10_1-h512-ffn2048	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	83.6	83.3	0.3	
TenTrans	stu-10_1-h512-ffn2048	GPU Batch	TenTrans	stu-20_1-h512-ffn2048	GPU Batch	82.0	79.8	2.2	
TenTrans	tea-20_6-h512-ffn4096	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	73.9	63.2	10.7	***
TenTrans	stu-20_1-h512-ffn2048	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	66.9	66.3	0.6	
TenTrans	tea-20_6-h512-ffn4096	GPU Batch	TenTrans	stu-20_1-h512-ffn2048	GPU Batch	88.4	88.0	0.4	*
Edinburgh	base	GPU Batch	TenTrans	tea-20_6-h512-ffn4096	GPU Batch	84.4	78.9	5.5	**
Edinburgh	base	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	88.7	82.1	6.6	***
Edinburgh	tiny11	GPU Batch	TenTrans	tea-20_6-h512-ffn4096	GPU Batch	88.0	81.1	6.9	**
Edinburgh	tiny11	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	72.1	57.6	14.5	***
Edinburgh	base	GPU Batch	NiuTrans	6_1_512	GPU Batch	87.4	74.7	12.7	***
Edinburgh	base	GPU Batch	NiuTrans	3_1_512	GPU Batch	83.0	73.7	9.3	***
Edinburgh	tiny11	GPU Batch	NiuTrans	6_1_512	GPU Batch	68.6	65.8	2.8	
Edinburgh	tiny11	GPU Batch	NiuTrans	3_1_512	GPU Batch	91.8	87.4	4.4	***
TenTrans	tea-20_6-h512-ffn4096	GPU Batch	NiuTrans	6_1_512	GPU Batch	67.2	65.9	1.3	
TenTrans	tea-20_6-h512-ffn4096	GPU Batch	NiuTrans	3_1_512	GPU Batch	89.2	87.3	1.9	***
NiuTrans	6_1_512	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	94.6	93.5	1.1	**
NiuTrans	3_1_512	GPU Batch	TenTrans	stu-20_1-h256-ffn1024	GPU Batch	84.4	82.3	2.1	
Edinburgh	base	GPU Latency	HuaweiTSC	base	1 Core Latency	91.4	86.9	4.6	***
Edinburgh	base	GPU Latency	HuaweiTSC	sm9	1 Core Latency	77.3	69.7	7.6	***
Edinburgh	base	GPU Latency	HuaweiTSC	sm6	1 Core Latency	86.0	77.6	8.4	***
Edinburgh	base	GPU Latency	HuaweiTSC	tiny	1 Core Latency	90.8	77.2	13.6	***
Edinburgh	tiny11	GPU Latency	HuaweiTSC	base	1 Core Latency	89.3	84.2	5.1	**
Edinburgh	tiny11	GPU Latency	HuaweiTSC	sm9	1 Core Latency	88.5	83.2	5.4	***
Edinburgh	tiny11	GPU Latency	HuaweiTSC	sm6	1 Core Latency	92.9	89.2	3.7	***
Edinburgh	tiny11	GPU Latency	HuaweiTSC	tiny	1 Core Latency	82.4	73.7	8.7	***
Edinburgh	base	1 Core Latency	Edinburgh	tiny11	1 Core Latency	67.5	65.0	2.5	**
Edinburgh	base	1 Core Latency	Edinburgh	4.12_1.micro.rowcol-0.5	1 Core Latency	66.9	62.2	4.8	***
Edinburgh	tiny11	1 Core Latency	Edinburgh	4.12_1.micro.rowcol-0.5	1 Core Latency	81.1	74.5	6.7	***
HuaweiTSC	base	1 Core Latency	HuaweiTSC	sm9	1 Core Latency	87.5	85.0	2.5	*
HuaweiTSC	base	1 Core Latency	HuaweiTSC	sm6	1 Core Latency	89.2	86.0	3.2	**
HuaweiTSC	base	1 Core Latency	HuaweiTSC	tiny	1 Core Latency	94.5	86.4	8.2	***
HuaweiTSC	sm9	1 Core Latency	HuaweiTSC	sm6	1 Core Latency	68.8	68.0	0.9	
HuaweiTSC	sm9	1 Core Latency	HuaweiTSC	tiny	1 Core Latency	90.2	85.8	4.3	***
HuaweiTSC	sm6	1 Core Latency	HuaweiTSC	tiny	1 Core Latency	79.3	73.2	6.1	***
HuaweiTSC	base	1 Core Latency	Edinburgh	base	1 Core Latency	84.7	84.6	0.1	
Edinburgh	base	1 Core Latency	HuaweiTSC	sm9	1 Core Latency	78.5	74.8	3.7	*
Edinburgh	base	1 Core Latency	HuaweiTSC	sm6	1 Core Latency	89.0	85.7	3.3	**
Edinburgh	base	1 Core Latency	HuaweiTSC	tiny	1 Core Latency	87.9	79.8	8.2	***
HuaweiTSC	base	1 Core Latency	Edinburgh	tiny11	1 Core Latency	90.7	90.4	0.2	
Edinburgh	tiny11	1 Core Latency	HuaweiTSC	sm9	1 Core Latency	81.5	78.9	2.5	
Edinburgh	tiny11	1 Core Latency	HuaweiTSC	sm6	1 Core Latency	90.9	90.1	0.7	
Edinburgh	tiny11	1 Core Latency	HuaweiTSC	tiny	1 Core Latency	85.9	77.9	8.0	***
HuaweiTSC	base	1 Core Latency	Edinburgh	4.12_1.micro.rowcol-0.5	1 Core Latency	89.4	81.9	7.5	***
HuaweiTSC	sm9	1 Core Latency	Edinburgh	4.12_1.micro.rowcol-0.5	1 Core Latency	93.4	90.8	2.6	
HuaweiTSC	sm6	1 Core Latency	Edinburgh	4.12_1.micro.rowcol-0.5	1 Core Latency	84.8	82.0	2.8	*
HuaweiTSC	tiny	1 Core Latency	Edinburgh	4.12_1.micro.rowcol-0.5	1 Core Latency	84.6	82.1	2.4	*

Table 6: Results of the pairwise contrastive direct assessment human evaluation for each evaluated system pair. The stronger system on the left is considered better than the weaker system on the right according to the Wilcoxon rank-sum test with p < 0.05 for $*$, p < 0.01 for $**$, p < 0.001 for $***$.

Team	Variant	Win	Ave.	Ave. z	Time (s)
NiuTrans	12_1_512	1	70.0	0.060	124
NiuTrans	6_1_512	0	77.3	0.017	95
NiuTrans	6_1_0	0	81.6	-0.023	94
NiuTrans	3_1_512	0	78.1	-0.057	81

(a) NiuTrans GPU Throughput

Team	Variant	Win	Ave.	Ave. z	Time (s)
TenTrans	stu-10_1-h512-ffn2048	1	85.9	0.104	280
TenTrans	stu-20_1-h512-ffn2048	0	87.7	-0.011	340
TenTrans	tea-20_6-h512-ffn4096	0	85.6	-0.069	456

(b) Tentrans GPU Throughput

Team	Variant	Win	Ave.	Ave. z	Time (s)
HuaweiTSC	base	2	91.6	0.181	14939
HuaweiTSC	sm9	1	79.5	0.139	8866
HuaweiTSC	sm6	1	75.6	0.005	7714
HuaweiTSC	tiny	0	81.3	-0.250	5138

(c) HuaweiTSC 1 Core Latency

Team	Variant	Win	Ave.	Ave. z	Time (s)
Edinburgh	base	3	83.8	0.214	140
Edinburgh	tiny11	3	75.3	0.106	115
TenTrans	tea-20_6-h512-ffn4096	1	76.3	-0.067	456
NiuTrans	3_1_512	0	83.8	-0.064	81
NiuTrans	6_1_512	0	74.7	-0.087	95
TenTrans	stu-20_1-h256-ffn1024	0	75.8	-0.210	257

(d) GPU Throughput

Team	Variant	Win	Ave.	Ave. z	Time (s)
Edinburgh	base	4	82.1	0.122	16815
Edinburgh	tiny11	2	88.7	0.078	9272
HuaweiTSC	sm9	1	85.6	-0.003	8866
HuaweiTSC	base	0	85.5	0.051	14939
HuaweiTSC	sm6	0	86.8	-0.027	7714
Edinburgh	4.12_1.micro.rowcol-0.5	0	86.9	-0.065	6343
HuaweiTSC	tiny	0	78.5	-0.131	5138

(e) Latency on 1 Core CPU. Total wall Time (s) is the same value as μs/sentence because there are 1 million sentences.

Team	Variant	Win	Ave.	Ave. z	Time (s)	Condition
Edinburgh	tiny11	4	86.7	0.165	15101	GPU Latency
Edinburgh	base	3	89.3	0.238	16851	GPU Latency
HuaweiTSC	sm9	2	79.8	-0.146	8866	1 Core Latency
HuaweiTSC	sm6	0	89.7	-0.155	7714	1 Core Latency
HuaweiTSC	base	0	81.8	-0.161	14939	1 Core Latency
HuaweiTSC	tiny	0	72.8	-0.342	5138	1 Core Latency

(f) Latency on GPU vs 1 Core CPU. Total wall Time (s) is the same value as μs/sentence because there are 1 million sentences.

Table 7: System rankings based on contrastive DA human-evaluation within selected groups of systems. Each system within a group was evaluated against each other system. Systems are ordered by the number of respective wins against other systems and DA z-score.

characters.[4]

The GPU latency track had been intended to attract non-autoregressive machine translation submissions in their ideal condition with a large GPU and no batch to parallelize. However, non-autoregressive papers (Libovický and Helcl, 2018; Gu and Kong, 2021) often rely on unreasonably poor autoregressive baselines in order to claim impressive-sounding speedups, when they are in fact slower than optimized autoregressive models seen here. While previous editions of the task did not measure latency, disabling batching is a simple command line modification to systems that existed at the time (Birch et al., 2018) but were omitted as baselines in non-autoregressive literature. All submissions this year are autoregressive.

An efficient training task is a natural extension. The challenge lies in defining proper development and testing conditions. Otherwise, participants will overfit by searching for the random seed that trains the fastest on a particular parallel corpus. Perhaps a parallel corpus could be halved to form development and test sets, but that would reveal the test set by omission and require trusting all participants. One participant was already caught cheating in a past edition of this shared task. Another option is that the test corpus could be a different surprise language pair, which would have the potentially positive effect that it also measures generalizability across languages. An interesting aspect of efficient training is that systems relying on backtranslation (Sennrich et al., 2016) incur substantial inference costs during their training cycle.

The one-month gap between the news task dead-

[4] https://cloud.google.com/translate/pricing

line and the efficient task deadline was too short and some teams noted this reduced the conditions they participated in. In addition, scaffolding would reduce the barrier to participation. This could take the form of providing a trained high-quality model, providing distilled (Kim and Rush, 2016) training data, or even optimized models where only the toolkit code is optimized. Providing this scaffolding would effectively require the organizers to perform the full task before releasing it to participants. If the training and test data are renewed each year as a countermeasure to overfitting and a participant that cheated, this would require more time between the news task and release of the news test set references.

German is a high resource language, which raises the computational cost of participation. A medium resource language would generally reduce training costs and explore whether results apply in this data condition.

The next task should aim to recruit more participants and perhaps separate the organization from one of the participants.

Acknowledgements


This work was conducted within the scope of the Horizon 2020 Research and Innovation Action *Bergamot*, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825303. GPU efficiency was supported by the European Union's Connecting Europe Facility under grant agreement No INEA/CEF/ICT/A2019/1927024, User-Focused Marian. This paper reflects the authors' views.

Intel Corporation has supported organization. The human evaluation was funded by Microsoft. We would like to thank Christian Federmann and Hitokazu Matsushita for their help with conducting human evaluation. Cloud credits were provided by the Oracle for Research program; we thank Rich Pitts for timely delivery.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. 2018. Findings of the second workshop on neural machine translation and generation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the IWSLT 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Ondřej Dušek, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová. 2017. Khresmoi summary translation test data 2.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. 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 and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, Online. Association for Computational Linguistics.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In

Proceedings of the Fourth Workshop on Neural Generation and Translation, pages 1–9, Online. Association for Computational Linguistics.

Niehues Jan, Roldano Cattoni, Stuker Sebastian, Matteo Negri, Marco Turchi, Salesky Elizabeth, Sanabria Ramon, Barrault Loic, Specia Lucia, and Marcello Federico. 2019. The IWSLT 2019 evaluation campaign. In *16th International Workshop on Spoken Language Translation 2019*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. 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 72–78, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Adithya Renduchintala, Denise Diaz, Kenneth Heafield, Xian Li, and Mona Diab. 2021. Gender bias amplification during speed-quality optimization in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 99–109, Online. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. 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.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Findings of the WMT Shared Task on Machine Translation Using Terminologies

Md Mahfuz ibn Alam[1], **Ivana Kvapilíková**[5], **Antonios Anastasopoulos**[1], **Laurent Besacier**[2], **Georgiana Dinu**[3], **Marcello Federico**[3], **Matthias Gallé**[2], **Philipp Koehn**[4], **Vassilina Nikoulina**[2], **Kweon Woo Jung**[2]
[1]Department of Computer Science, George Mason University [2]NAVER
[3]AWS [4]Facebook / Johns Hopkins University [5]Charles University
{malam21, antonis}@gmu.edu

Abstract

Language domains that require very careful use of terminology are abundant and reflect a significant part of the translation industry. In this work we introduce a benchmark for evaluating the quality and consistency of terminology translation, focusing on the medical (and COVID-19 specifically) domain for five language pairs: English to French, Chinese, Russian, and Korean, as well as Czech to German. We report the descriptions and results of the participating systems, commenting on the need for further research efforts towards both more adequate handling of terminologies as well as towards a proper formulation and evaluation of the task.

1 Introduction

Language domains that require very careful use of terminology are abundant. The need to adequately translate within such domains is undeniable, as shown by e.g. the different WMT shared tasks on biomedical translation.

More interestingly, as the abundance of research on domain adaptation shows, such language domains are (a) not adequately covered by existing data and models, while (b) new (or "surge") domains arise and models need to be adapted, often with significant downstream implications: consider the new COVID-19 domain and the large efforts for translation of critical information regarding pandemic handling and infection prevention strategies.

In the case of newly developed domains, while parallel data are hard to come by, it is fairly straightforward to create word- or phrase-level terminologies, which can be used to guide professional translators and ensure both accuracy and consistency.

This shared task[1] replicated such a scenario, and invited participants to explore methods to incorporate terminologies into either the training or the inference process, in order to improve both the accuracy and consistency of MT systems on a new domain.

2 Shared Task Details

The shared task focused on five language pairs, with systems evaluated on:

- English to French
- English to Chinese
- English to Russian
- English to Korean
- Czech to German

The last three language pairs were "surprise" language pairs. This shared task construction follows a three-phase approach to ensure the generalizability of the findings, inspired by other multilingual shared tasks (Vylomova et al., 2020). In this setting, only part of the evaluation language pairs (or languages) are revealed from the beginning (the **Development Phase**). In this elongate period (a couple of months), the participants are provided with data in some language pairs to *develop* their methods. The second phase is the **Generalization phase**, which is a short time period (two to three weeks in this task's case), in which additional (surprise) language settings are revealed, only giving the shared task participants enough time to deploy a system, as opposed to allowing them enough time to also perform extensive optimization on the datasets. The final stage is the **Evaluation phase**, in which the test data are released and the methods are evaluated on these held-out data.

The goal of this 3-stage approach (with both development and surprise language pairs) is to avoid approaches that overfit on language selection, and instead evaluate the more realistic scenario of needing to tackle the new domain in a new language in a limited amount of time. The surprise language pairs were announced 3 weeks before the start of the evaluation campaign.

The organizers provided training/development

[1] http://statmt.org/wmt21/terminology-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 652–663
November 10–11, 2021.

data and terminologies for the above language pairs. Test sets were released at the beginning of the evaluation period. The participating teams were invited to participate in any or all of the language pairs.

2.1 Data

Training The shared task primarily focused on a constrained submission setting, in which the participants could only use any parallel or monolingual data listed in previous versions of WMT shared tasks to train their systems. Some pre-trained systems listed at the shared task announcement (mBERT, XLM, XLM-R, mBART, mT5, M2M100) were also allowed, but should be disclosed by the participants. We note that the training data allowed come from a "general" domain, as opposed to e.g. highly specialized biomedical data, which in theory should be more helpful for this setting.

Terminologies The shared task focused on adapting MT systems to the health domain in general, with a particular interest in the surge COVID-19 domain.

The terminologies for the English to French, Chinese, Russian, and Korean language pairs were taken from the publicly available TICO-19 project (Anastasopoulos et al., 2020), a multi-organizational project that created data to aid translators and evaluate MT systems on the COVID-19 domain. The terminologies were created by linguists at Google and Facebook in consultation with domain experts, providing translations for about 600 terms in each language. The terminologies are publicly available.[2]

The Czech-German medical terminology was generated automatically from Wikipedia. We considered all Wikipedia titles corresponding to the category *Health care* or to one of its subcategories, and all titles linked from the text. The list of (sub)categories was manually filtered to only include relevant articles. We treated all page titles as terms and relied on the Wikipedia language links to provide their translations. Furthermore, we used redirection links to obtain synonyms of both source and target terms.

For all terminologies, we truecased the terms using a pretrained truecaser and manually checked the results. The Czech-German terminology was eventually further reduced to only include terms which occurred in the EMEA medical corpus.

Development and Test The development and test data for French, Chinese, and Russian were taken from the publicly available TICO-19 evaluation data. The organizers additionally created Korean translations of the English source-side sentences, which will be made available as part of the original TICO-19 datasets.[3]

The primary source of the Czech-German development and test data is the EMEA[4] parallel corpus of the European Medicines Agency. We cleaned it using the Moses tools, searched for terms and their translations and tagged the occurrences. The surface forms used for the search were collected from a corpus of in-domain Wikipedia articles which includes links to the lemmatized Wikipedia titles/terms next to their inflected forms. Target options were retrieved from the terminology and enriched with surface forms. Out of all sentences with terms, we selected around 3.5k sentences for the dev set and 1.1k for the test set. The development and test sets were tagged automatically but the test set was manually corrected to get rid of the artifacts caused by the automatic generation.

2.2 Ensuring Terminology Consistency on the Evaluation Datasets

It is worth noting that, originally, none of the development and test data were created under the constraints imposed by the specific terminologies we use. As such, we needed to ensure that the data 'complied' with the terminologies in order to guarantee a meaningful, accurate, and fair to the participants evaluation of the shared task's research questions.

The TICO-19 project created the evaluation dataset independently of the terminologies.[5] In our preliminary analysis, we first searched for all terminology terms on the English side of the parallel data, also searching over the lemmatized versions of English sentences. The choice of starting from the English side is due to two reasons: (a) it reflects the actual translation direction the data was created with and that we evaluate on, (b) it reduces the rate of possible false negative/positive term matches due to the lack of morphological complexity of English.

[2] https://tico-19.github.io/

[3] The data are freely available here: https://tico-19.github.io/.

[4] https://opus.nlpl.eu/EMEA.php

[5] Although we note that the dataset went through an independent quality assurance process and several correction iterations, if required.

Example 1 (ID: Wikipedia_handpicked_4:1709)	
Source:	after blowing your nose , <term, src='coughing', tgt='tousser'> coughing </term> or <term, src='sneezing', tgt='éternuer'> sneezing </term> .
Translation:	après s ' être mouché ou avoir toussé / éternué ;
Annotation 1:	<term, src='coughing', tgt='tousser'> Label: c) variation_correct
Annotation 2:	<term, src='sneezing', tgt='éternuer'> Label: c) variation_correct
Tagged translation:	après s ' être mouché ou avoir <term, src='coughing'> toussé </term> / <term, src='sneezing'> éternué </term>;
Term-compl. transl.:	N/A
Example 2 (ID: Wikipedia_handpicked_4:1703)	
Source:	people can also become <term, src='infected', tgt='infecté'> infected </term> with <term, src='respiratory disease', tgt='maladie respiratoire'> respiratory diseases </term> such as <term, src='influenza', tgt='grippe'> influenza </term> or the <term, src='common cold', tgt='rhume'> common cold </term> , for example , if they do not wash their hands before <term, src='touch', tgt='toucher'> touching </term> their eyes , nose , or mouth (i . e . , mucous membranes) .
Translation:	il est possible de contracter des maladies respiratoires telles que la grippe ou le rhume , par exemple , en omettant de se laver les mains avant de se toucher les yeux , le nez ou la bouche (c . - à - d . les muqueuses) .
Annotation 1:	<term, src='infected', tgt='infecté'> Label: e) not_used
Annotation 2:	<term, src='respiratory disease', tgt='maladie respiratoire'> Label: c) variation_correct
Annotation 3:	<term, src='influenza', tgt='grippe'> Label: b) exact_match_correct
Annotation 4:	<term, src='common cold', tgt='rhume'> Label: b) exact_match_correct
Annotation 5:	<term, src='touch', tgt='toucher'> Label: b) exact_match_correct
Tagged translation:	il est possible de contracter des <term, src='respiratory disease'> maladies respiratoires</term> telles que la grippe ou le rhume , par exemple , en omettant de se laver les mains avant de se toucher les yeux , le nez ou la bouche (c . - à - d . les muqueuses) .
Term-compl. transl.:	il est possible d'etre <term, src= infected> infecté </term> avec des <term, src='respiratory disease'> maladies respiratoires</term> telles que grippe ou le rhume , par exemple , en omettant de se laver les mains avant de se toucher les yeux , le nez ou la bouche (c . - à - d . les muqueuses) .
Example 3 (ID: CMU_1:77	
Source:	I have hay <term, src='fever', tgt='fièvre'> fever </term> though too
Translation:	mais j ' ai le rhume des foins aussi
Annotation 1:	<term, src='fever', tgt='fièvre'> Label: a) does_not_apply
Tagged translation:	N/A
Term-compl. transl.:	N/A
Example 4 (ID: Wikipedia_handpicked_1:1311	
Source:	the strongest <term, src='self quarantine', tgt='auto - quarantaine'> self - <term, src='quarantine', tgt='quarantaine'> quarantine </term> </term> instructions have been issued to those in high risk groups .
Translation:	les instructions de quarantaine individuelle les plus strictes ont été données aux personnes des groupes les plus à risque .
Annotation 1:	<term, src='self quarantine', tgt='auto - quarantaine'> Label: e) not_used
Annotation 2:	<term, src='quarantine', tgt='quarantaine'> Label: b) exact_match_correct
Tagged translation:	N/A
Term-compl. transl.:	les instructions d' <term, src='self quarantine'> auto - quarantaine </term> les plus strictes ont été données aux personnes des groupes les plus à risque

Table 1: Examples (from English-French TICO-19) of expected annotations that ensure that the evaluation datasets are compliant with the terminologies. ('Term-compl. transl.' == 'terminology-compliant translation').

Having the source-side terms identified, we assume all of them should be translated according to the terminology. We then search the target side (both original and lemmatized) for the translation required by the terminology, and created a tag on the source-side term if we found an exact match. Last, we showed all sentences to professional translators, who were instructed to produce three types of annotations, for each source-side term. The first is a **label** describing whether (a) the automatically-annotated source-side term should not be translated by the terminology i.e. it is not really a term, (b) the tagged exact match is correct, (c) the translation is compliant with the terminology even though there is not an exact match, (d) the tagged exact match is incorrect, or (e) the source term translation is applicable in the context, but not used. The second annotation is a **tagged translation** for any terms labeled as (a), (c), or (d), denoting exactly which part of the target-side corresponds to the source-side term. The third annotation is a **tagged terminology-compliant translation**, where if any

source-side terms are labeled with (d) or (e), we ask the translators to rephrase the target side in order to make it compliant with the terminology.

Table 1 shows example sentences from the dataset, along with their expected annotations from the translators. Below we provide the exact instructions given to the annotators, which also reference the same examples.

[begin annotation instructions]
About: This task is about determining if a translation is compliant with a terminology data base and perform inline annotations on the translations to mark the terms used.
Annotators receive: Source side input, together with approximate terminology matches on the source side.
Annotators return: For each term match, please annotate a Label:

(a) `does_not_apply`: The terminology is not applicable in the context because of wrong meaning on the source side (Example 3). Please use *a) if you think the translation should not comply with the terminology matched*, irrespective of whether the translation uses it or not.
(b) `exact_match_correct`: The term translation is found exactly as is in the target and its usage is correct (it fits the context and agrees grammatically with the sentence). (Example 2)
(c) `variation_correct`: The translation is compliant with the terminology, however the term translation appears in a different form in the target (Examples 1 and 2). If only part of the term was preserved, use this label if this partial term is sufficient and completely preserves the meaning. Please use *b) or c) if you think the translation is compliant with the terminology*.
(d) `incorrect`: The term is found in the target, as an exact match or as a variant, but it is used incorrectly, either semantically or grammatically: e.g. the term use does not convey the required meaning, there is a wrong inflection or other grammatical disagreement.
(e) `not_used`: The term translation is applicable in the context, but not used (Example 2, 4). Make this only for clear omissions: everything else should be variation (correct or incorrect variation) Please use *d) or e) if you think the translation is not compliant with the terminology, but it should*.

Tagged translation: For any terms that are labeled as a), c) or d) please add inline markup to identify the fragments of the translation that they match.
For each source sentence, please generate a **Tagged Terminology-compliant translation**: if any of d) to e) apply to any term in the sentence, meaning the translation is not compliant with the terminology for at least one term, please provide an alternate translation that is compliant with the terminology w.r.t all the terms in the sentence. If there is no acceptable translation that would use the expected target term then you should annotate the target with a) does_not_apply. If all terms in sentence match a) b) or c), leave this empty.
[/end annotation instructions]

Through this process, we ended up modifying 284 (9.25%), 251 (8.17%), 450 (14.65%), and 809 (26.34%) sentences in the French, Chinese, Russian, and Korean datasets respectively, in order to make them terminology-compliant. Last, the Czech-German terminologies were directly derived from the parallel data hence they implicitly directly reflect the underlying data, so there was no need for the aforementioned process.

2.3 Evaluation

The evaluation of the shared task used several metrics, focusing on both translation accuracy and terminological consistency.

- Translation accuracy was evaluated with standard reference-based MT metrics (BLEU, chrF, BERTscore, COMET). In light of recent work (Kocmi et al., 2021), we rank systems according to the COMET metric.

- we also performed terminology-targeted evaluation (to evaluate for consistency). We use the metrics outlined by Alam et al. (2021), namely exact-match term accuracy, 1-TERm score, and window overlap accuracy. We rank systems according to term exact-match accuracy.

Briefly, the lemmatized exact-match term accuracy is an accuracy score that searches for exact term translation matches (of the terminology required output) over either the lemmatized or the original hypothesis. The window overlap accuracy identifies the translation of the term, and then

scores its context, to measure how well a translated term is placed in the hypothesis. Last, the 1-TERm score is a modification of the TER metric (Snover et al., 2006), biased to assign higher edit cost weights for words belonging to a term (and then simply reversed so that a higher score is better). We refer the reader to Alam et al. (2021) for further discussion of the metrics and supporting arguments for their use.

Last, we evaluate whether differences between systems are statistically significant using paired bootstrap resampling (Koehn, 2004), over sentence-level COMET and exact-match accuracy scores. Based on this information we cluster statistically-insignificantly-different (i.e. similarly performing) systems when we produce their final rankings.

Winning submissions will be the ones that are Pareto-optimal along the two evaluation metrics that a good but also terminology-compliant system should maximize: exact-match accuracy (which captures terminology consistency) and COMET (which captures general translation quality). As such, there is the possibility that each language pair will have multiple winning submissions.

3 Participants and System Descriptions

We received a total of 43 submissions from 9 teams. Below we provide a short description of each submission.

CUNI (Jon et al., 2021b) Authors competed on `En-Fr` language pair. The terminology constrains are inserted as done in (Jon et al., 2021a). The target translation of specific terms is appended to the source sentence as a suffix and separated by a special token (if multiple constraints occur for a single sentence, an additional token separator is added). In order to have more training data of this form, synthetic constraints are added by sampling random token subsequences from the target sentence and appending them to the source sentence as described earlier. Note that since no modification is done on target side of the parallel data, no post-processing of the MT output is needed. As NMT systems trained from this pre-processed data sometimes fail to generate inflection in the translation output, terminology tokens appended to the source are lemmatized for both training and inference which brings improvements over the different shared task metrics.

Huawei (HW-TSC) (Wang et al., 2021b) Authors submitted output of an unconstrained system to `En-Zh` language pair. They train a Transformer *big* architecture on both out-of-domain and in-domain (biomedical) data. Parallel data in biomedical domain is augmented using more resources from TAUS[6] and back-translation of monolingual in-domain data is also applied. For the terminology shared-task, authors applied the system created for the biomedical translation shared task (described in (Wang et al., 2021b)) without any specific adaptation except appending the terminology dictionary to the end of training data. No separate paper was submitted for the terminology task.

Kakao Enterprises (KEP) (Bak et al., 2021) Authors submitted to `En-Fr`, `En-Zh`, `En-Kr`, `Cz-De`. A detailed data cleaning is performed, removing between 6% and 14% of the data. In-domain data is back-translated (only for `En-Fr` and `En-Kr`) and is selected by a a combination of keywords spotting and domain similarity, measured as perplexity of an in-domain language model. A first model is obtained by adding to that synthetic language pairs obtained by verbalizing the terminology database. The only language pair were this verbalization does not yield improvement is `Cz-De`, whose terminology was automatically constructed. Models obtained in this manner were submitted for `En-Zh`, `En-Kr`, `Cz-De`.

For `En-Fr` additional techniques are used: as those obtained the highest COMET score we detail them there. The final system for that language pair is trained inspired by techniques from (Bergmanis and Pinnis, 2021a; Dinu et al., 2019), but without modifying the model architecture. The source data is modified by adding immediately after a source term the corresponding target *lemma*, separated by special tokens. The model is pre-trained on randomly selected verbs and nouns, and fine-tuned using the terminology ontology. Interestingly, the pre-trained model - while improving Exact Match with respect to the baselines - degrades all other metrics. That degradation is however recovered and even improved when fine-tuning. For `En-Ko` and `Cs-De` ensemble models were used.

Lingua Custodia (LC) (Ailem et al., 2021a) The team participated in `En-Fr`, `En-Ru` and `En-Zh` tasks. They build on top of (Ailem et al., 2021b) by inserting the terminology as constraints in the

[6] https://md.taus.net/corona

source sentence. Such constraints represent special tags around the detected source term followed by the target term from the terminology, the original source term is masked. Presence of such constraints at training encourages the model to copy the correct term translation. In case where multiple translations are proposed by the terminology, the one which is present in the target sentence is chosen at training time. At inference time the translation is selected at random. In order to enforce learning signal, the team enriched parallel data with back-translation of monolingual data that contains terminology. Authors show that the proposed method allows to improve significantly for standard MT evaluation metrics, as well as terminology oriented metrics (Alam et al., 2021) over the standard baseline without terminological constraints.

PROMT (Molchanov et al., 2021) The team submitted two systems (`En-Fr` and `En-Ru`), both of which are transformer models implemented on MarianMT (Junczys-Dowmunt et al., 2018). The first approach uses a rule-based system (SmartMT) to modify the neural system's output, which extracts rules only for noun phrases. If the desired output of a source term is not found in the NMT output, the rule-based system identifies the term's current translation and its morphological analysis (case and number) in order to substitute it with the terminology-provided translation in the desired inflection. The second approach is an adaptation of (Dinu et al., 2019) to MarianNMT toolkit. Each source terminological term is followed by its translation using special tokens to signal these terminological entries in the text (and impose a soft-constraint to the translation system). Model is retrained from such pre-processed data. Data augmentation is also performed to create more synthetic data with terminology markup. Both approaches are rather close in performance.

SPECTRANS (Ballier et al., 2021) This team sumbitted to `En-Fr` language pair. They experimented with 2 open source NMT toolkits JoyeNMT (Kreutzer et al., 2019) and OpenNMT (Klein et al., 2017). After the first experiments with Europarl they retained OpenNMT which gave better performance. Their best runs were trained on CommonCrawl augmented with terminological data. They provided qualitative analysis of terminology-related translations and discuss the limitations of the terminologies provided for the task.

SYSTRAN (Pham et al., 2021) This participant submitted to `En-Fr` language pair and proposed two methods to incorporate terminology. The first approach, based on (Michon et al., 2020), replaces source and target terminological terms by placeholders including a unique identifier plus morphological information (masculine/feminine and singular/plural). In a variant of this method, the source terminology word form is also incorporated in the source stream. At training time, NMT model is learnt on such pre-processed data and a post-processing step recovers the word tokens from the placeholders after inference. The second approach (which lead to better performance) consists in learning a copy behaviour for terminological tokens at training time: terminology translations are inserted in the source sentence either by appending the target term (its surface or lemma form) to its source version, or by directly replacing the original term with the target one. A NMT system is trained on such pre-processed data and no post-process for recovering terminology tokens is needed at inference as target side of parallel data remains untouched. For both approaches however, a grammatical error correction is applied to the MT hypotheses in order to limit morphology errors. The impact of such post-processing on BLEU is positive, although small.

TermMind (Wang et al., 2021a) The team submitted to `En-Zh` task. Similar to (Ailem et al., 2021a) they build on top of (Ailem et al., 2021b) by inserting terminological constraints in the training data. In the case where multiple translations are available they augment source sentence with all possible translations (which is different from (Ailem et al., 2021a) who kept only one translation). In order to strengthen the learning signal participants extend given terminologies with biphrases extracted from parallel data and integrate the constraints for those biphrases as well. Finally, they used backtranslation, fine tuning on pseudo in-domain data and ensembling to strengthen the baseline model. Ensembling methods seem to lead to the best results.

TILDE (Bergmanis and Pinnis, 2021c) The team participated to `En-Fr`, `En-Ru` and `Cz-De` language pairs. They focused primarily on terminology filtering, outlining several notable shortcomings of the Shared Task's terminologies, most of which are due to the use of terminologies intended

for human translators (as opposed to terminologies created specifically for integration with MT systems). They devise two strategies for selecting among multiple target candidates for a source term, finding that an alignment-based technique outperforms the option of always selecting the first terminology entry. The MT systems are transformer-based using MarianMT, also integrating the method of Bergmanis and Pinnis (2021b) for incorporating terminology constraints in a soft manner.

4 Results and Discussion

The results and rankings for English-French are listed in Table 2 and for English-Chinese in Table 3. The results for the surprise language pairs are in Table 4 for English-Russian and Table 5 for English-Korean and Czech-German.

In the English-French translation task, there are two winning submissions. Two ProMT submissions ranked first according to exact-match accuracy (along with a CUNI submission), but the ProMT.soft submission is statistically significantly better than the other two with respect to COMET, hence it is one of the winning submission. The second winning submission is the one by KEP, which ranks first according to COMET, but also according to 1-TER, which indicates that it might strike a good balance between general translation quality and term consistency.

In the English-Chinese translation task there is a single winning submission, the one by TermMind (system 2), which ranks first according to both metrics. We note that another submission (HW-TSC) is statistically significantly better than all submissions in all metrics except for 1-TERm, but this submission is an unconstrained one, and hence it is excluded from the rankings.

In English-Russian the ProMT submission ProMT.soft is the clear winner, ranking as the single best system according to exact-match accuracy, as well as one of the two best systems according to COMET. Interestingly, the other system that ranks first according to COMET (ProMT.smartnd.v2) ranks first according to 1-TERm score, but also *last* according to exact-match accuracy, denoting perhaps an orthogonality between the goals of terminological consistency and general translation quality, where prioritizing one over the other leads to performance drops along the other dimension.

Last, the submissions by KEP are the winning ones for English-Korean and Czech-German. For the former language pair it was the only submitted system (see discussion on potential reasons), while for Czech-German it ranked for best system according to exact match accuracy with the other submission (by TildeMT), but was significantly better according to COMET. Although TildeMT used a more sophisticated approach to the terminology translation, the KEP team had a stronger baseline and used ensembling which significantly increased both general translation quality and the term accuracy.

4.1 General Quality

It was pointed out by Bergmanis and Pinnis (2021c) that a majority of terms from the terminologies were represented in the training corpora, which could lead to an underestimation of the importance of terminology in the metrics. The results show that using terminology constrains leads to an improvement over the baselines trained without it, but the effect would be more substantial if the training corpora were filtered to exclude sentences with terms.

Perhaps a future iteration of the shared task could include an explicitly novel domain, although how well such a domain indeed exists or is even possible in the age of big data where our models can be trained on a large part of the Internet is debatable. An alternative is to carefully filter the training corpora to remove sentences with the terms, to create a truly challenging domain adaptation with terminologies setting.

4.2 Terminology Consistency

The discussion of the Shared Task taught us that narrow terminology with unambiguous translations is more suitable for terminology-focused machine translation than a broader and more universal terminology with several target options. Unlike human translators who naturally choose from translation alternatives, it is difficult for a MT system to filter out noisy or inappropriate word forms. While a narrow terminology can ensure a proper and exact translation of terms, e.g. when translating a lecture with several special terms known in advance, we believe that a broad terminology can serve for more general domain adaptation using existing lexical resources. We note that several participating teams highlighted this issue, e.g. Ballier et al. (2021); Bergmanis and Pinnis (2021c).

The TICO terminologies in a few cases included additional comments aimed at translators who are

English-French	Rankings		Terminology-focused				Translation Quality	
	accoding to		Exact-Match	Window Overlap		1-TERm		BLEU
System	ex-m. acc.	COMET	Accuracy	(2)	(3)	Score	COMET	(truecased)
ProMT.soft	1-3	3	**0.974**	**0.359**	**0.352**	0.625	0.752	47.69
ProMT.smartnd	1-3	4-5	0.966	0.357	0.348	0.626	0.746	47.89
CUNI-Primary_not_scored	1-3	6-10	0.967	0.342	0.334	0.601	0.732	46.92
KEP	4-6	1	0.950	0.34	0.337	**0.632**	**0.781**	**49.60**
CUNI-Primary_lemm	4-6	6-10	0.946	0.34	0.332		0.729	46.80
CUNI-Contr._not_scored	4-6	12-18	0.950	0.33	0.331	0.588	0.693	45.48
SYSTRAN-app+_corr	7-17	2	0.934	0.355	0.349	0.631	0.766	48.87
SYSTRAN-app_corr	7-17	6-10	0.938	0.283	0.297	0.614	0.729	45.81
SYSTRAN-mrk_corr	7-17	6-10	0.938	0.283	0.297	0.614	0.729	45.81
SYSTRAN-mrk+_corr	7-17	6-10	0.938	0.283	0.297	0.614	0.729	45.81
TildeMT	7-17	11	0.939	0.329	0.322	0.593	0.706	45.04
CUNI-Contr._sf_choices	7-17	12-18	0.923	0.313	0.310	0.557	0.682	42.72
LinguaCustodia-Sys1	7-17	12-18	0.920	0.343	0.336	0.595	0.677	44.49
LinguaCustodia-Sys2_new	7-17	12-18	0.919	0.344	0.335	0.598	0.681	44.90
LinguaCustodia-Sys2	7-17	12-18	0.919	0.345	0.334	0.591	0.676	44.21
CUNI-Contrastive_sf	7-17	12-18	0.918	0.321	0.317		0.684	44.08
CUNI-Contr._lemm_choices	7-17	12-18	0.913	0.323	0.317	0.567	0.678	43.78
ProMT.baseline	18	4-5	0.898	0.33	0.331	0.624	0.745	47.50
SPECTRANS3-CC-fr_en	19	19	0.871	0.296	0.296	0.507	0.596	40.02
SPECTRANS	20	20	0.795	0.27	0.267	0.495	0.296	34.93
SPECTRANS_2	21	21	0.640	0.248	0.241	0.480	0.212	33.59

Table 2: English-French results. The systems are ranked and clustered according to exact-match accuracy (secondarily according to COMET) based on statistical significance tests. We **highlight** the best score per metric.

English-Chinese	Rankings		Terminology-focused				Translation Quality	
	accoding to		Exact-Match	Window Overlap Acc.		1-TERm		BLEU
System	ex-m. acc.	COMET	Accuracy	(2)	(3)	Score	COMET	(truecased)
*HW-TSC**	*1**	*1**	***0.886***	***0.282***	***0.285***	*0.514*	***0.716***	***40.73***
TermMind-sys2	2	2	0.856	0.27	0.274	**0.534**	0.709	40.47
LinguaCustodia - Sys1-v2	3-6	4	0.828	0.225	0.227	0.438	0.643	29.61
LinguaCustodia - Sys1	3-6	5-7	0.829	0.223	0.225	0.437	0.637	29.16
LinguaCustodia - Sys2	3-6	5-7	0.829	0.222	0.225	0.433	0.635	28.92
LinguaCustodia - Sys1-v3	3-6	5-7	0.828	0.241	0.244	0.472	0.641	33.73
TermMind	7-8	3	0.668	0.22	0.227	0.513	0.696	37.51
KEP	7-8	8	0.645	0.18	0.187	0.249	0.229	27.12

Table 3: English-Chinese results. The systems are ranked and clustered according to exact-match accuracy based on statistical significance tests. We **highlight** the best score per metric. *: unrestricted system.

English-Russian	Rankings		Terminology-focused				Translation Quality	
	accoding to		Exact-Match	Window Overlap Acc.		1-TERm		BLEU
System	ex-m. acc.	COMET	Accuracy	(2)	(3)	Score	COMET	(truecased)
ProMT.soft	1	1-2	**0.909**	**0.254**	**0.255**	0.482	0.631	31.06
ProMT.smartnd.v1	2-5	3	0.857	0.25	0.250	0.482	0.624	31.52
LinguaCustodia - Sys1	2-5	5-8	0.854	0.24	0.249	0.472	0.598	28.84
LinguaCustodia - Sys1-v2	2-5	5-8	0.849	0.24	0.247	0.473	0.600	28.81
TildeMT-v2	2-5	9-10	0.863	0.22	0.226	0.457	0.550	28.14
LinguaCustodia - Sys2-v2	6-7	5-8	0.849	0.247	0.248	0.474	0.604	29.13
LinguaCustodia - Sys2	6-7	5-8	0.847	0.242	0.244	0.471	0.601	28.97
ProMT.baseline	8-9	4	0.823	0.24	0.241	0.481	0.620	31.49
TildeMT	8-9	9-10	0.817	0.21	0.219	0.456	0.548	28.16
ProMT.smartnd.v2	10	1-2	0.788	0.243	0.241	**0.487**	**0.634**	**31.92**

Table 4: English-Russian results. The systems are ranked and clustered according to exact-match accuracy (and secondarily according to COMET) based on statistical significance tests. We **highlight** the best score per metric.

directly looking at them, as opposed to the format that terminologies aimed at machines would use. We will take this into account in future iterations of the shared task – it is worth noting, though, that if most available terminologies are designed for human translators, it should probably be up to the

Language Pair	System	Rankings according to ex-m. acc.	COMET	Terminology-focused Exact-Match Accuracy	Window Overlap Acc. (2)	(3)	1-TERm Score	Translation Quality COMET	BLEU (truecased)
English-Korean	KEP	1	1	**0.569**	**0.067**	**0.065**	**0.251**	**0.581**	**16.52**
Czech-German	KEP	1-2	1	0.866	**0.428**	**0.424**	**0.474**	**0.694**	**34.10**
	TildeMT	1-2	2	**0.871**	0.390	0.385	0.434	0.641	30.01

Table 5: English-Korean and Czech-German results. The systems are ranked and clustered according to exact-match accuracy based on statistical significance tests. We **highlight** the best score achieved per metric.

NLP/ML/MT practitioners to figure out how to best use the existing data, rather than demanding new, dedicated resources. Similarly, when compiling the Czech-German terminology, we aimed at creating a universal lexicon of medical terms with a wide coverage. Many terms have multiple translations and we used the Wikipedia redirection links as a proxy for synonyms. Unfortunately, they became a source of noise because not all redirects are synonyms and not all synonyms are appropriate in every context. We tackled the former by semi-automatic filtering and left the latter up to the candidate translation engine to select the version of the word appropriate for the given context. Unfortunately, some problematic terms remained even in the final version of the terminology, as pointed out by Bergmanis and Pinnis (2021c).

4.3 Development vs Surprise Language Pairs

The participants had significantly more time to develop systems for English-French and English-Chinese, as opposed to the other three surprise language pairs. This is reflected partly on the total submitted systems in each language pair, where English-Korean and Czech-German received only 1 and 2 submissions respectively. We hypothesize that another explanation for this lies in the much more low-resource setting of these two language pairs, which generally tend to lead to lower quality systems, which might in turn discourage the participants.

A second potential explanation could lie in the general cohort of participants, which is largely comprised of teams from industry (the only exception is the CUNI team that is an academic one). Perhaps the two low-resource language pairs are simply translation directions that the participating institutions are less interested in – which we take as an indication for the importance of including such less-researched, low-resource, under-served language pairs in future iterations of this shared task, to encourage research in languages and language pairs beyond those with the most obvious commercial value.

4.4 Czech-German Analysis

We believe that even with the automatically generated resources this task provided an important insight into translation of terms between two linguistically different and morphologically rich languages such as German and Czech.

When analyzing the results, we focused on the phenomenon of nominal compounding in German. A natural translation of terms into German often results in a compound of a term and a general word, e.g. *Hormonproduktion* (production of hormones), or two terms, e.g. *Plasmaprotein* (plasma protein). Compounding is an important aspect of terminology-based translation to German that the model should have the capacity to create compounds from terminology entries.

The automatic metrics favor translations into two separate words, even though a compound is often more natural. We analyzed how candidate translations handled concrete cases; see Table 6 for an example. Out of 262 sentences with this phenomenon in the reference, the correct compound word was generated in 112 and 133 cases by the TildeMT and KEP systems, respectively. Both systems generate compounds from terms, although the former was trained with terminology constraints and the latter only saw the terms during explicit training on the terminology entries.

5 Related Work

Phrase-based statistical MT systems (Koehn et al., 2003) allowed for fine-grained control over the system's output by design, e.g. by incorporating domain-specific dictionaries into the phrase table, or by forcing translation choices for certain words or phrases. On the other hand, the currently state-of-the-art approach of neural machine translation (NMT) does not inherently allow for such control over the system's output. Some approaches

SRC	Mozkové metastázy vykazovaly nekonsistentní nebo žádnou fluorescenci.	...krvácení do svalů nebo hematom.
TGT	Hirnmetastasen zeigten inkonsistente oder keine Fluoreszenz.	...Muskelblutung oder Hämatom.
TildeMT	Zerebrale Metastasen zeigten eine inkonsistente oder keine Fluoreszenz.	...Blutungen in den Muskeln oder Hämatom
KEP	Hirnmetastasen zeigten eine inkonsistente oder keine Fluoreszenz.	...Muskelblutung oder Hämatom.

Table 6: Examples of term compounding in German where candidates handle term translation differently.

incorporate dictionaries through interpolation of the decoder's probability with a lexical probability based on source-side attention matches (Arthur et al., 2016). Perhaps the most common paradigm is *constrained decoding* (Hokamp and Liu, 2017; Anderson et al., 2017; Post and Vilar, 2018, *inter alia*), where the terminology matches are presented as hard constraints that the beam search must satisfy.

Constrained decoding is not without disadvantages: it can be computationally expensive and it is often brittle when applied in realistic conditions (Dinu et al., 2019). To this end, some works (Dinu et al., 2019; Bergmanis and Pinnis, 2021b; Exel et al., 2020; Niehues, 2021) introduced approaches where the terminological constraints are provided as input to the NMT as additional annotations inline with the source sentence. As such, these can be considered as "soft" constraints, as there is no guarantee that the NMT system will indeed produce an output containing them.

In any case, the best practice for incorporating terminological constraints in NMT is both underresearched and still not settled yet, especially in the case of morphologically rich languages, underlying the need for this shared task.

6 Conclusion

We presented the results of the first edition of the WMT21 shared task on MT using Terminologies. For the purposes of the task we created new evaluation datasets, annotated by professional translators for their terminology consistency, based on the TICO-19 data for English to French, Chinese, Russian, and Korean, as well as a dataset for Czech-German based on the EMEA corpus.

The Shared Task received 43 submissions from 9 teams, 8 from industry and 1 from academia, underscoring the general applicability of our focus problem ('how best can we use a terminology in MT?') on real-world settings. Most submissions add soft or hard constraints on the source side that the MT learns to handle, as proposed in (Dinu et al., 2019), but other novel approaches include terminology filtering for selecting between multiple options provided by the terminology, or replacing terms with placeholders to be inserted after the MT has produced the output. We devised multiple terminology-targeted metrics and evaluated systems along both these metrics as well as general translation quality. In most cases we find that, encouragingly, one does not necessarily have to sacrifice general translation quality for terminology compliance, as long as the terminology is of adequate standards.

In future iterations of the Shared Task, we will take into account the distinction between terminologies created for humans (which are abundant) and terminologies created specifically for MT systems which need to be created, and have different requirements/specifications that the former. In addition, we will attempt to consider a new domain, rather than focusing again on the biomedical domain and specifically COVID-19 (although this is a great example of a "surge" domain that immediately required that translation providers and MT engines adapt in order to handle translations of large volumes of text in this novel domain).

Acknowledgements

The organizers want to first thank the participants for their submissions and their constructive feedback during and after the Shared Task, which made the evaluation more robust. In addition, we are thankful to NAVER for creating the English-Korean translations and to Appen for quality assurance on them. The Czech-German track was organized with the support of the grant 1050119 of the Charles University Grant Agency. Last, we are thankful to Amazon, Tanya Badeka and Margo Lynch for the creation of the terminology-compliant translations of the evaluation datasets. Anastasopoulos is generously supported by NSF grant IIS-2125466.

References

Melissa Ailem, Jinghsu Liu, and Raheel Qader. 2021a. Lingua custodia's participation at the wmt 2021 machine translation using terminologies shared task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Melissa Ailem, Jingshu Liu, and Raheel Qader. 2021b. Encouraging neural machine translation to satisfy terminology constraints. *CoRR*, abs/2106.03730.

Md Mahfuz Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency. arXiv:2106.11891.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federman, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. Tico-19: the translation initiative for covid-19. In *NLP COVID-19 Workshop*, Online.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Yunju Bak, Jimin Sun, Jay Kim, Sungwon Lyu, and Changmin Lee. 2021. Kakao enterprise's wmt21 machine translation using terminologiestask submission. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Nicolas Ballier, Dahn Cho, Bilal Faye, Zong-You Ke, hanna Martikainen, Mojca Pecman, Jean-Baptiste Yunés, Guillaume Wisniewski, Lichao Zhu, and Maria Zimina-Poirot. 2021. The spectrans system description for the wmt21 terminology task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021a. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021b. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021c. Dynamic terminology integration for covid-19 and other emerging domains. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Miriam Exel, Bianka Buschbeck, Lauritz Brandt, and Simona Doneva. 2020. Terminology-constrained neural machine translation at SAP. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 271–280, Lisboa, Portugal. European Association for Machine Translation.

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. Association for Computational Linguistics.

Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021a. End-to-end lexically constrained machine translation for morphologically rich languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4019–4033, Online. Association for Computational Linguistics.

Josef Jon, Michal Novák, João Paulo Aires, Dušan Variš, and Ondrej Bojar. 2021b. Cuni systems for wmt21: Terminology translation shared task. In *Proceedings of the Sixth 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, et al. 2018. Marian: Fast neural machine translation in c++. arXiv:1804.00344.

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, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. arXiv:2107.10821.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Philipp Koehn, Franz J Och, and Daniel Marcu. 2003. Statistical phrase-based translation. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. *CoRR*, abs/1907.12484.

Elise Michon, Josep Crego, and Jean Senellart. 2020. Integrating domain terminology into neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3925–3937, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexander Molchanov, Vladislav Kovalenko, and Fedor Bykov. 2021. Promt systems for wmt21 terminology translation task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Jan Niehues. 2021. Continuous learning in neural machine translation using bilingual dictionaries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 830–840, Online. Association for Computational Linguistics.

MinhQuang Pham, Antoine Senellart, Dan Berrebbi, Josep Crego, and Jean Senellart. 2021. Systran @ wmt 2021: Terminology task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

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, New Orleans, Louisiana. 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*, volume 200. Citeseer.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Ke Wang, Shuqin Gu, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021a. Termmind: Alibaba's submission to the wmt21 machine translation using terminologies shared task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Weixuan Wang, Wei Peng, Xupeng Meng, and Qun Liu. 2021b. Huawei aarc's submissions to the wmt21 biomedical translation task: Domain adaption from a practical perspective. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Findings of the WMT 2021 Biomedical Translation Shared Task: Summaries of Animal Experiments as New Test Set

Lana Yeganova[1*] **Dina Wiemann**[2] **Mariana Neves**[3] **Federica Vezzani**[4]
Amy Siu[5] **Iñigo Jauregi Unanue**[6] **Maite Oronoz**[7] **Nancy Mah**[8]
Aurélie Névéol[9] **David Martinez**[10,11] **Rachel Bawden**[12] **Giorgio Maria Di Nunzio**[13]
Roland Roller[14] **Philippe Thomas**[14] **Cristian Grozea**[15] **Olatz Perez de Viñaspre**[7]
Maika Vicente Navarro[16] **Antonio Jimeno Yepes**[10]

[1]NCBI/NLM/NIH, Bethesda, USA
[2]Novartis AG, Basel, Switzerland
[3]German Centre for the Protection of Laboratory Animals (Bf3R),
German Federal Institute for Risk Assessment (BfR), Berlin, Germany
[4]Dept. of Linguistic and Literary Studies University of Padua, Italy
[5]Berliner Hochschule für Technik, Germany
[6]University of Technology Sydney, Sydney, Australia
[7]IXA NLP Group, University of the Basque Country, Donostia, Spain
[8]Fraunhofer Institute for Biomedical Engineering (IBMT), Berlin, Germany
[9]LISN, CNRS, Université Paris-Saclay, Orsay, France
[10]University of Melbourne, Australia
[11]Doctor Evidence, Santa Monica, CA, USA
[12]Inria, France
[13]Dept. of Information Engineering, University of Padua, Italy
[14]German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
[15]Fraunhofer Institute FOKUS, Berlin, Germany
[16]Maika Spanish Translator, Melbourne, Australia

Abstract

In the sixth edition of the WMT Biomedical Task, we addressed a total of eight language pairs, namely English/German, English/French, English/Spanish, English/Portuguese, English/Chinese, English/Russian, English/Italian, and English/Basque. Further, our tests were composed of three types of textual test sets. New to this year, we released a test set of summaries of animal experiments, in addition to the test sets of scientific abstracts and terminologies. We received a total of 107 submissions from 15 teams from 6 countries.

1 Introduction

Machine translation (MT) is the automatic translation of textual resources from one language to another. It is an important component in many applications and natural language processing (NLP) pipelines in the clinical and biomedical domains. On the one hand, some resources, such as specific biomedical terminologies, are only available for a limited number of languages. English is especially well covered in the Unified Medical Language System (UMLS) (Lindberg et al., 1993) while other languages are not (Wilde, 2021). On the other hand, there are many publications written in languages other than English and are therefore inaccessible to researchers who cannot read those languages.

This context has been the overarching goal for the organization of the WMT Biomedical task. The first edition took place in 2016 and addressed scientific abstracts for English/French (both directions), English/Spanish (both directions), and En-

[*] The organization of the biomedical task is complex and relies on varied essential contributions from many individuals. Authors are listed randomly because we could not do justice to the contributors using a single ranking. We would like to acknowledge MN for dataset preparation and general task organization, CG for creating baselines, AN for compiling information on participants methods, AJY for conducting the automatic evaluation, LY, DW, MN, FV, AS, AN, GMDN, RR, PT, MVN, AJY for evaluating the alignment of the test sets, and LY, DW, MN, FV, AS, MO, NM, AN, RB, GMDN, RR, PT, MVN, AJY for conducting the manual evaluation. All authors approved the final version of the manuscript. E-mail for contact: mariana.lara-neves@bfr.bund.de

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 664–683

glish/Portuguese (both directions) (Bojar et al., 2016). The subsequent shared task included six new language pairs, namely, English into Czech, English into German, English into Hungarian, English into Polish, English into Romanian, and English into Swedish, in addition to a new type of document, viz., health information texts (Jimeno Yepes et al., 2017). In 2018, we started using MEDLINE® as the source for our scientific abstracts and addressed a new language pair, namely English/Chinese (both directions), in addition to some of the languages already considered in the previous year (Neves et al., 2018). In the subsequent year, we introduced the translation of biomedical terminologies (from English into Spanish), in addition to the MEDLINE abstracts for the five language pairs from the 2018 task (Bawden et al., 2019). In 2020, we added three new language pairs, namely English/Russian (both directions), English/Italian (both directions), and English into Basque (en2eu) (Bawden et al., 2020).

For this year's shared task[1], we address the same eight language pairs as last year (Bawden et al., 2020) on the same translations tasks (scientific abstracts and terminologies). The main novel feature this year is a new test set composed of summaries of planned animal experiments to be translated from German into English. The list below summarizes the language pairs addressed this year:

- English to Basque (en2eu)
- English to Chinese (en2zh) and Chinese to English (zh2en)
- English to French (en2fr) and French to English (fr2en)
- English to German (en2de) and German to English (de2en)
- English to Italian (en2it) and Italian to English (it2en)
- English to Portuguese (en2pt) and Portuguese to English (pt2en)
- English to Russian (en2ru) and Russian to English (ru2en)
- English to Spanish (en2es) and Spanish to English (es2en)

Finally, we highlight the new aspect that we introduced in the 2021 edition of our shared task, namely, a novel test set for the automatic translation of summaries of animal experiments from German into English (see Section 2.4).

2 Training and test data

No additional training data was released for any of the language pairs, with the exception of en2eu, where we provide last year's test set as new training data for abstracts and terminology. As for the tests sets, we released test sets for scientific abstracts, terminologies, and summaries of animal experiments as follows:

- Scientific abstracts:
  - English to Basque
  - Chinese/English (both directions)
  - French/English (both directions)
  - German/English (both directions)
  - Italian/English (both directions)
  - Portuguese/English (both directions)
  - Russian/English (both directions)
  - Spanish/English (both directions)
- Terms from biomedical terminologies:
  - English to Basque
- Summaries of animal experiments:
  - German to English

Table 1 shows the number of documents, sentences and terms (if applicable) for each test set. In this section, we give details on the construction of the test sets.

2.1 MEDLINE test sets

Similar to previous years, we retrieved recent MEDLINE abstracts that were available in both English and one of the seven other languages we evaluate on (namely Chinese, French, German, Italian, Portuguese, Russian, and Spanish). The abstracts in both languages were processed as follows:

- language detection with the Python `langdetect` library;[2]
- sentence splitting using the Python `syntok` library;[3]

[1] http://www.statmt.org/wmt21/biomedical-translation-task.html

[2] https://pypi.org/project/langdetect/

[3] https://github.com/fnl/syntok

Language pairs	Abstracts		Terminology	Summaries	
	Documents	Sentences	Terms	Documents	Sentences
en2eu	76	450	2,736	-	-
de2en	50	480/481	-	30	648
en2de	50	516/501	-	-	-
es2en	50	445/444	-	-	-
en2es	50	486/501	-		
fr2en	50	365/351	-	-	-
en2fr	50	384/394	-		
it2en	43	432/407	-	-	-
en2it	44	460/448	-		
pt2en	50	468/484	-	-	-
en2pt	50	494/486	-		
ru2en	50	428/436	-	-	-
en2ru	50	354/373	-		
zh2en	50	341/393	-	-	-
en2zh	50	425/375	-		

Table 1: Number of documents, sentences and terms in the test sets released for this shared task. Some abstracts had to be removed from the it2en and en2it during the evaluation phase.

- sentence alignment using the `GMA` tool[4] for all language pairs except for English/Chinese, for which the `Champollion` tool[5] was used;
- random retrieval of 100 abstracts for each language pair;
- and manual validation of the selected abstracts using the "quality checking" task in the Appraise tool (Federmann, 2010), of which the results are shown in Table 2.

Table 2 shows that the highest quality was obtained for the zh/en test sets, with up to 94.5% perfectly aligned sentences. This is actually not a surprise, since these were the only test sets where an expert manually discarded abstracts that are clearly non-parallel, e.g. when the entire English abstract corresponds to only the first half of the Chinese abstract. A high quality of over 80% was also obtained for four language pairs, namely pt/en (90.4%), es/en (88.4%), fr/en (86.0%), and it/en (80.3%).

For en/fr, the automatic alignment was manually reviewed. In this process, the overall corpus size increased from 630 sentences to 775 sentences, mainly through the addition of article titles that had not been collected in English, and did not have any equivalent in French. In terms of alignment quality, it is important to note that the problematic categories *Source>Target* and *Target>Source* are significantly reduced in the revised corpus. The de/en test set obtained a slightly lower quality of 77.7%, while only 54.2% of ru/en sentences were perfectly aligned. Similar to previous years, the automatic evaluation was carried out for all sentences as well as only for the perfectly aligned (hereafter referred to as "OK") ones.

2.2 Basque abstracts

As we mentioned in (Bawden et al., 2020), the presence of Basque in MEDLINE is almost non-existent. In this edition we have again used the abstracts from the journal Osagaiz[6] as part of the test set, but due to the low production of this journal written in Basque, we have added abstracts from the journal *Gaceta Médica de Bilbao*[7], which contains abstracts written in Spanish, English, and Basque. From the 76 documents and 450 sentences mentioned in table 1, 18 documents and 119 sentences are from the Osagaiz journal, and 50 documents and 331 sentences from *Gaceta Médica de Bilbao*. The sentences were manually aligned by human annotators.

2.3 Terminologies

In the WMT20 edition, on behalf of Osakidetza (Basque Public Health System), we released 27,900 terms of the Basque ICD-10-CM edition, 2,000 of

[4] https://nlp.cs.nyu.edu/GMA/
[5] http://champollion.sourceforge.net/
[6] http://www.osagaiz.eus
[7] http://www.gacetamedicabilbao.eus/index.php/gacetamedicabilbao

Language	OK	Source>Target	Target>Source	Overlap	No Align.	Total
de/en	710 (77.7%)	38 (4.2%)	40 (4.4%)	34 (3.7%)	92 (10.0%)	914
es/en	792 (88.4%)	55 (6.1%)	15 (1.7%)	7 (0.8%)	27 (3.0%)	896
fr/en	540 (85.7%)	68 (10.8%)	10 (1.6%)	1 (0.2%)	11 (1.7%)	630
fr/en §	666 (86.0%)	9 (1.2%)	1 (0.1%)	8 (1.0%)	91 (11.7%)	775
it/en	666 (80.3%)	51 (6.2%)	26 (3.1%)	13 (1.6%)	73 (8.8%)	829
pt/en	838 (90.4%)	54 (5.8%)	18 (2.0%)	15 (1.6%)	2 (0.2%)	927
ru/en	371 (54.2%)	79 (11.5%)	63 (9.2%)	25 (3.6%)	147 (21.5%)	685
zh/en	658 (94.5%)	16 (2.3%)	9 (1.3%)	1 (0.2%)	12 (1.7%)	696

Table 2: Statistics (number of sentences and percentages) of the quality of the automatic alignment for the MEDLINE test sets. For each language pair, the total number of sentences corresponds to the 100 documents that constitute the two test sets (one for each language direction). § Results after manual correction of sentence segmentation and/or alignment.

which were used for evaluation. This year, we updated some of the Basque translations for correctness and cohesion. The full set from last year was released for training and a new set of 2,736 terms was used as a test set.

2.4 Summaries of planned animal experiments

We released a test set of 30 summaries of planned animal experiments that were retrieved from the AnimalTestInfo database[8], which is maintained by the German Federal Institute for Risk Assessment (BfR). The summaries describe planned and approved animal experiments to be carried out in Germany, which are anonymously stored in this online database in a bid to improve transparency (Bert et al., 2017). The aim of considering these summaries in this shared task is to assess the quality of MT of these documents, which is relevant for a couple of projects currently being carried out in the BfR, such as mining for alternative methods to animal experiments. A previous larger training set and test set from this database has been previously used in another shared task for the assessment of the automatic assignment of ICD-10 codes (Neves et al., 2019). The summaries contain following information (see Figure 1 for an example):

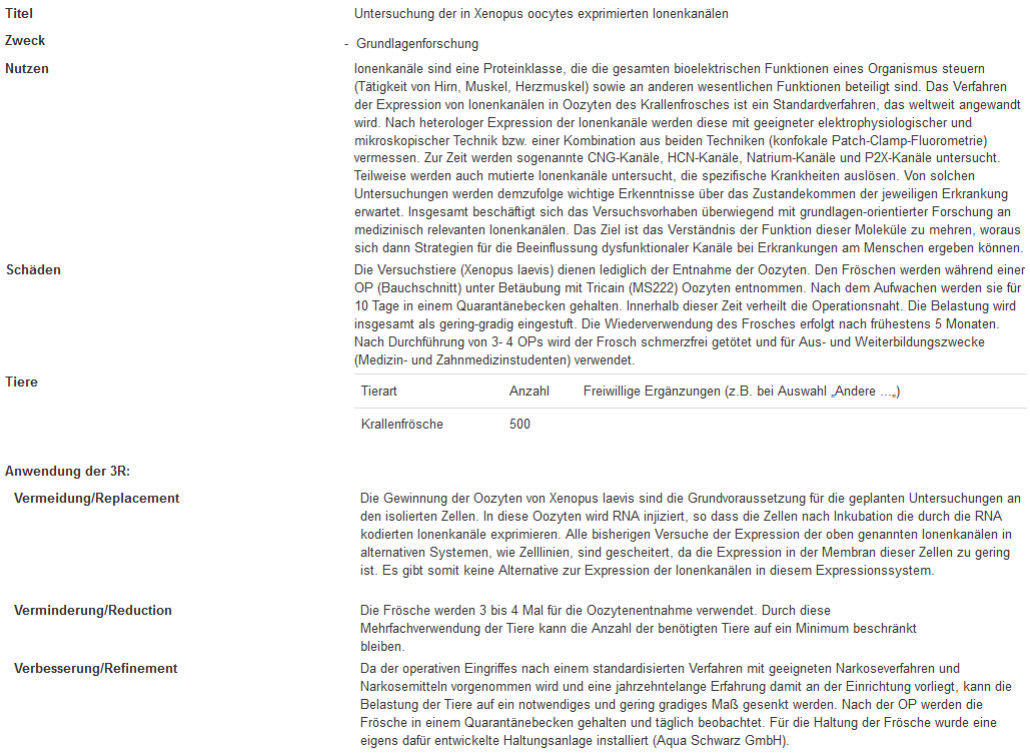
- title;
- aim of the study (e.g., basic research);
- benefits of the experiments;
- species and number of animals to be used;
- comments regarding the compliance to the so-called 3R principle (replacement, reduction, refinement of animal experiments).

The summaries were selected from the database in a way that addressed various animal species, and they were then manually translated by an English native speaker with a high knowledge of German. Before releasing the data, we converted the summaries into a format that is suitable for the WMT shared task.

3 Baselines

This year we had more choices for the baselines. As before, one option was to use our own models, trained with Marian NMT (Junczys-Dowmunt et al., 2018) on biomedical texts. A new option was to use pre-trained models, not specialized on biomedical texts. We used our own models as baselines for the following language directions: en2de, en2es, en2fr, en2pt, de2en, es2en, fr2en, pt2en. For en2zh, en2ru, en2it, en2eu, zh2en, ru2en, it2en, we used the pre-trained generic Marian NMT models available in the HuggingFace "Transformers" library.[9] An interesting question was whether the specialized models were still better than the newest out-of-the-box pretrained models. To this end, for en2de and en2fr we also tested the recent T5-large[10] model (Raffel et al., 2019). For en2fr, it outperformed our own model (trained on biomedical data) by almost 3 BLEU points, whereas for en2de the two systems were fairly comparable. The performance of the systems submitted starts from close to baseline for some language directions (e.g. for en2fr, en2es, de2en), whereas for other languages all systems were much better than the baseline (e.g. es2en, pt2en and especially ru2en).

[8] https://animaltestinfo.de/
[9] https://huggingface.co/Helsinki-NLP
[10] https://huggingface.co/t5-large



Titel Untersuchung der in Xenopus oocytes exprimierten Ionenkanälen

Zweck - Grundlagenforschung

Nutzen Ionenkanäle sind eine Proteinklasse, die die gesamten bioelektrischen Funktionen eines Organismus steuern (Tätigkeit von Hirn, Muskel, Herzmuskel) sowie an anderen wesentlichen Funktionen beteiligt sind. Das Verfahren der Expression von Ionenkanälen in Oozyten des Krallenfrosches ist ein Standardverfahren, das weltweit angewandt wird. Nach heterologer Expression der Ionenkanäle werden diese mit geeigneter elektrophysiologischer und mikroskopischer Technik bzw. einer Kombination aus beiden Techniken (konfokale Patch-Clamp-Fluorometrie) vermessen. Zur Zeit werden sogenannte CNG-Kanäle, HCN-Kanäle, Natrium-Kanäle und P2X-Kanäle untersucht. Teilweise werden auch mutierte Ionenkanäle untersucht, die spezifische Krankheiten auslösen. Von solchen Untersuchungen werden demzufolge wichtige Erkenntnisse über das Zustandekommen der jeweiligen Erkrankung erwartet. Insgesamt beschäftigt sich das Versuchsvorhaben überwiegend mit grundlagen-orientierter Forschung an medizinisch relevanten Ionenkanälen. Das Ziel ist das Verständnis der Funktion dieser Moleküle zu mehren, woraus sich dann Strategien für die Beeinflussung dysfunktionaler Kanäle bei Erkrankungen am Menschen ergeben können.

Schäden Die Versuchstiere (Xenopus laevis) dienen lediglich der Entnahme der Oozyten. Den Fröschen werden während einer OP (Bauchschnitt) unter Betäubung mit Tricain (MS222) Oozyten entnommen. Nach dem Aufwachen werden sie für 10 Tage in einem Quarantänebecken gehalten. Innerhalb dieser Zeit verheilt die Operationsnaht. Die Belastung wird insgesamt als gering-gradig eingestuft. Die Wiederverwendung des Frosches erfolgt nach frühestens 5 Monaten. Nach Durchführung von 3- 4 OPs wird der Frosch schmerzfrei getötet und für Aus- und Weiterbildungszwecke (Medizin- und Zahnmedizinstudenten) verwendet.

Tiere

Tierart	Anzahl	Freiwillige Ergänzungen (z.B. bei Auswahl „Andere ...„)
Krallenfrösche	500	

Anwendung der 3R:

Vermeidung/Replacement Die Gewinnung der Oozyten von Xenopus laevis sind die Grundvoraussetzung für die geplanten Untersuchungen an den isolierten Zellen. In diese Oozyten wird RNA injiziert, so dass die Zellen nach Inkubation die durch die RNA kodierten Ionenkanäle exprimieren. Alle bisherigen Versuche der Expression der oben genannten Ionenkanälen in alternativen Systemen, wie Zelllinien, sind gescheitert, da die Expression in der Membran dieser Zellen zu gering ist. Es gibt somit keine Alternative zur Expression der Ionenkanälen in diesem Expressionssystem.

Verminderung/Reduction Die Frösche werden 3 bis 4 Mal für die Oozytenentnahme verwendet. Durch diese Mehrfachverwendung der Tiere kann die Anzahl der benötigten Tiere auf ein Minimum beschränkt bleiben.

Verbesserung/Refinement Da der operativen Eingriffes nach einem standardisierten Verfahren mit geeigneten Narkoseverfahren und Narkosemitteln vorgenommen wird und eine jahrzehntelange Erfahrung damit an der Einrichtung vorliegt, kann die Belastung der Tiere auf ein notwendiges und gering gradiges Maß gesenkt werden. Nach der OP werden die Frösche in einem Quarantänebecken gehalten und täglich beobachtet. Für die Haltung der Frösche wurde eine eigens dafür entwickelte Haltungsanlage installiert (Aqua Schwarz GmbH).

Figure 1: Example of a summary for a planned animal experiment. Source: https://animaltestinfo.de/dsp_show_ntp.cfm?ntpID=19362

4 Teams and systems

This year, we received a total of 107 runs from 15 teams from the following countries: China (8), Spain (2), France (1), Japan (1), Pakistan (1), and USA (2). Table 3 presents the list of teams. We can note four returning teams: Huawei_AGI (most team members were part of the Huawei United team in 2020), LISN (LIMSI in 2020), nrpu-fjwu and TMT.

Table 4 presents an overview of the runs submitted by each team for language directions translating *from* English. Table 5 presents an overview of the runs submitted by each team for language directions translating *into* English.

We did not receive any submission for en2pt, even though we did receive submissions from one team for the opposite direction of this language pair, i.e., pt2en. Unfortunately, we did not receive any submission for the new test set that we released this year, i.e., for the summaries of planned animal experiments. Nevertheless, the test set (including the reference translation) is available for the research community for further experiments.

Similarly to the WMT 2020 biomedical task edition, we asked participants to fill in a survey with key information regarding the specific material and methods used in their self-identified primary runs that were used for manual evaluation. The survey comprised 14 questions covering the translation methods and corpora used.

On average, the time spent by participants to supply information for one language pair was 6 minutes and 35 seconds (Median: 3 minutes and 27 seconds). This is consistent with the 2020 survey statistics and suggests that the time commitment for supplying this information is limited, even for teams addressing more than one language pair.

All teams used transformer-based neural MT (NMT) and largely relied on existing implementations: 7 teams submitted runs using available libraries while 5 teams submitted runs using their own NMT implementations. Teams often used the same setup for a range of language pairs. Table 6 shows details of the teams' methods.

For in-domain data, teams used the training data distributed as part of the task as well as many of the sources described in (Névéol et al., 2018). Additional corpus used for Chinese have been prepared by the teams but are not always available or described in details. We can also notice that the use or pre-processing of resources supplied by the task organizers can differ between teams as the size reported for seemingly similar data can differ sig-

Team ID	Institution
ECNU_PAHT	Pingan Health Tech / ECNU, China
FJDMATH (Martínez, 2021)	Fujitsu DMATH, Japan
Haozhiweizi	Shanghai Jiaotong University, China
Huawei_AGI (Wang et al., 2021a)	Huawei Technologies, China
Huawei_TSC (Yang et al., 2021)	Huawei Translation Service Center, China
JinDong	unknown, China
LISN	LISN, CNRS, France
MT Learner	Microsoft Research, China
NVIDIA NeMo (Subramanian et al., 2021)	NVIDIA, USA
nrpu-fjwu (Naz et al., 2021)	Fatima Jinnah Women University, Pakistan
talp_upc (Rafieian and Costa-Jussà, 2021)	Universitat Politècnica de Catalunya, Spain
TMT (Wang et al., 2021b)	Tencent AI Lab, China
Transperfect	Transperfect Translations, Spain
Volctrans	ByteDance, China
ZengHuiMT	FORYOR HEALTH, USA

Table 3: List of the participating teams.

Teams	en2eu	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh	Total
ECNU_PAHT	-	-	-	-	-	-	-	A3	3
FJDMATH	T2A2	-	-	-	-	-	-	-	4
Haozhiweizi	-	-	-	-	-	-	-	A1	1
Huawei_AGI	-	A3	-	A3	A3	-	-	A3	12
Huawei_TSC	-	A3	-	-	-	-	-	A3	6
JingDong	-	-	-	-	-	-	-	A1	1
LISN	-	-	-	A3	-	-	-	-	3
NVIDIA NeMo	-	-	-	-	-	-	A2	-	2
talp_upc	-	-	A2	-	-	-	-	-	2
TMT	-	A1	A1	A1	-	-	A1	-	4
Transperfect	-	-	A3	-	-	-	A2	A2	7
Volctrans	-	-	-	-	-	-	-	A3	3
ZengHuiMT	-	-	-	-	-	-	-	A1	1
Total	4	7	6	7	3	0	5	17	49

Table 4: Overview of the submissions from all teams and test sets translating from English. We identify submissions to the abstracts testsets with an "A" and to the terminology test set with a "T". The value next to the letter indicates the number of runs for the corresponding test set, language pair, and team.

Teams	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en	Total
ECNU_PAHT	-	-	-	-	-	-	A3	3
Haozhiweizi	-	-	-	-	-	-	A1	1
Huawei_AGI	A3	-	A3	A3	-	-	A3	12
Huawei_TSC	A3	-	-	-	-	-	A3	6
JingDong	-	-	-	-	-	-	A1	1
LISN	-	-	A3	-	-	-	-	3
MT Learner	-	-	-	A2	A2	A2	-	6
NVIDIA NeMo	-	-	-	-	-	A1	-	1
nrpu-fjwu	A3	A3	A3	-	-	-	-	9
talp_upc	-	A2	-	-	-	-	-	2
TMT	A1	A1	A1	-	-	A1	-	4
Transperfect	-	A3	-	-	-	A2	A2	7
Volctrans	-	-	-	-	-	-	A2	2
ZengHuiMT	-	-	-	-	-	-	A1	1
Total	10	9	10	5	2	6	16	58

Table 5: Overview of the submissions from all teams and test sets translating into English. We identify submissions to the abstracts test sets with an "A" and to the terminology test set with a "T". The value next to the letter indicates the number of runs for the corresponding test set, language pair, and team.

Team ID	Language pair	NMT implementation	Trained	Fine-Tuned	BT	LM
Huawei_AGI	All	transformer (Own)	No	Yes	Yes (except zh2en)	No
Huawei_TSC	All	Marian,Fairseq	No	Yes	Yes (except en2de)	No
LISN	All	Fairseq	Yes	Yes	Yes	No
MT Learner	All	Marian-NMT	No	Yes	Yes	No
NVIDIA NeMo*	All	transformer (unspecified)	Yes	Yes	Yes	Yes
talp_upc	All	OpenNMT-transformer	Yes	No	Yes	No
nrpu-fjwu	All	Fairseq	Yes	Yes	No	No
TMT	All	Fairseq	Yes	No	Yes (except en2fr)	mBART (en2de,es,fr,ru)

Table 6: Overview of methods used by participating teams. Information is self-reported through the dedicated survey for each selected "best run" (information on the NVIDIA model is inferred from their system description (Subramanian et al., 2021)). BT indicates if backtranslation is used and LM if language models were used.

nificantly. Table 7 provide details of the in-domain data used by the teams.

For relevant language pairs, parallel data from other WMT tracks (e.g., News Task) was used. Out-of-domain data was also used in the form of pre-trained base models. Table 8 shows details of the out-of-domain data used by the teams.

We note that a number of the corpora used are referred to as "in house" corpus or data. This may indicate survey fatigue as this type of description is more frequently used for out of domain data, which appeared towards the end of the survey.

5 Automatic evaluation

For all the abstracts test sets, we evaluated system outputs using BLEU (Papineni et al., 2002) as provided by the Moses tool *mteval-v14.pl*[11]. We used this metric for the en2eu abstracts, summaries of animal experiments, and MEDLINE test sets. In en2zh, a modified version of the tool was used that removed white spaces and text is split in way that each character is a word.

The results for the en2eu abstract test sets are given in Table 9. There was a single team (Fujitsu DMATH) that submitted two runs, based on BPE dropout and sub-subword features with a Transformer (base) model. One of the runs (run2) included multilingual data from an English–Spanish terminology. The results are not as high as in the MEDLINE abstracts task, but they are above the baseline system, and they have improved from the best results from the 2020 challenge (0.1453 vs. 0.1279).

For the en2eu terminology test sets, we evaluated the translated concepts in terms of two metrics: (i) accuracy, by relying on strict matches (case insensitive) between the reference translation and predictions; and (ii) BLEU score, as measured by the Python NLTK module sentencebleu. The results are presented in Table 10. The same systems from FUJITSU DMATH participated in this task, and the BLEU score was higher than the score for abstracts, but there was a drop in performance from the results in 2020. This could have happened because the systems were tuned for abstracts and not terminologies. As is the case for abstracts, for the terminology set, run1 outperforms run2 again, showing that multilingual data seems to harm performance in this setting.

For the summaries of animal experiments, we only present the results obtained by our baseline system (Table 11).

Finally, for the Medline test sets, we performed evaluation based on all the sentences in the test set, including the poorly aligned ones, as well as an evaluation based on only the perfectly aligned ones (see Table 2). The results *from* English into the foreign languages are presented in Table 12, while the ones *into* English are presented in Table 13. The results calculated for all sentences, and not only the perfectly aligned ones, are published on the shared task's web site.[12]

For translation from English (cf. Table 12), the highest BLEU score of 0.5117 was obtained by the Transperfect team for en2es. Moreover, for all the language pairs for which the Huawei_TSC participated, i.e., en2de and en2zh, this team obtained the highest score, namely 0.3259 and 0.4650 respectively. For en2fr and en2it, the best performance was obtained by the Huawei_AGI team,

[11] https://github.com/moses-smt/mosesdecoder

[12] http://www.statmt.org/wmt21/results_biomedical.pdf

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
de/en	Huawei_AGI	MEDLINE corpus supplied by WMT biomedical task organizers	2.4 M	Yes	53 M (en)
	Huawei_TSC	MEDLINE corpus supplied by WMT biomedical task organizers	3.03M	Yes	21.43M (en)
	nrpu-fjwu	sources provided by WMT biomedical task organizers (UFAL, Medline Abstracts and EMEA)	3.71 M	No	-
	TMT	corpus provided by WMT biomedical task organizers and UFAL.	2.5 M	Yes	2.5 M
es/en	Talp_upc	UFAL, Pubmed, Medline, IBECS,UNcor-m and OPUS	6.86 M	No	-
	TMT	corpus provided by WMT biomedical task organizers and UFAL.	1.6 M	Yes	1.6 M
	Transperfect	corpus provided by WMT biomedical task organizers	618 K	No	-
fr/en	Huawei_AGI	MEDLINE corpus supplied by WMT biomedical task organizers	3.6 M	Yes (en)	53 M
	LISN	Bio-medical corpora provided by the task organiser along with Taus and Cochrane	6 M	Yes (fr)	0.81 M
	nrpu-fjwu	sources provided by WMT biomedical task organizers e.g. UFAL, Scielo Health, EDP, Medline Titles, Medline Abstracts and EMEA.	4.36 M	No	-
	TMT	corpus provided by WMT biomedical task organizers and UFAL.	3.5 M	Yes	3.5 M
it/en	Huawei_AGI	MEDLINE corpus supplied by WMT biomedical task organizers, TAUS	374 K	Yes (en)	55 M
	MT Learner	Corpus supplied by WMT biomedical task organizers, and in-domain data filtered from an in-house corpus.	364 K	Yes (en)	1.5 M
pt/en	MT Learner	Corpus supplied by WMT biomedical task organizers, and in-domain data filtered from an in-house corpus.	1.6 M	Yes (en)	6.2 M
en/ru	MT Learner	Corpus supplied by WMT biomedical task organizers, augmented with in-house corpus.	2.2 M	Yes (en)	2.1 M
	NVIDIA	Corpus supplied by organizers, augmented with automatically filtered news-task corpus.	256k	?	?
	TMT	Corpus supplied by organizers, augmented with in-house corpus.	1 M	WMT biomedical task and UFAL	?
	Transperfect	"internal data" (unspecified)	6.1 M	No	-
en/zh	ECNU_PAHT	In-house corpus (unspecified)	6 M	No	-
	Huawei_AGI	In-house data collected from a portion of abstracts of China Master's and Doctoral Dissertations.	847 K	No	-
	Huawei_TSC	In-house corpus (unspecified)	1.35M	Yes	36.11M (zh), 21.43M (en)
	Transperfect	"internal data" (unspecified)	6.8 M	No	-

Table 7: Overview of in-domain corpora used by participating teams. Information is self reported through our survey for each selected "best run" (information on the NVIDIA model is inferred from their task paper).

with 0.4531 and 0.0.4425 respectively. Finally, the NVIDIA NeMo team obtained the best score (0.4139) for en2ru.

For translation into English (cf. Table 13), the highest score over all teams and language pairs was 0.5685, which was obtained by the MT Learner team for pt2en. TMT obtained the best results for three of the language pairs, namely de2en, es2en, and fr2en, with the scores 0.4501, 0.5382, and 0.4928 respectively. For it2en and zh2en, slightly higher scores (0.4570 and 0.3943 respectively) were obtained by the Huawei_AGI team, when compared to the ones from the MT learner (0.4558) and Huawei_TSC (0.3904) respectively. Finally, the NVIDIA NeMo team obtained the top score (0.4918) for the only language pair (ru2en) and run that they submitted.

6 Manual evaluation

Similar to previous years, we manually validated a sample of the abstracts to compare the teams' primary submissions to each other and to the reference translation.

For the MEDLINE abstracts, we aimed for approximately 100 perfectly aligned sentences and retrieved the corresponding abstracts. The sentences were randomly retrieved, but we aimed to select abstracts with a higher percentage of perfectly aligned

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
en/de	Huawei_AGI	"in house data"	6 M	No	-
	Huawei_TSC	Corpus supplied by the WMT 2020 News task organizers	96.6 M	Yes	150M
	nrpu-fjwu	No	-	No	-
	TMT	Europarl-v10, Common Crawl corpus, ParaCrawl, News Commentary-v15 and Wiki Titles-v2	37.8 M	No	-
en/es	TALP	TED Talks	1.97 M	No	-
	TMT	Europarl-v7, Common Crawl corpus, News Commentary, ParaCrawl	30.3 M	No	-
	Transperfect	No	-	No	-
en/fr	Huawei_AGI	"in house data"	3 M	No	-
	LISN	WMT14 general domain corpus	35 M	No	-
	nrpu-fjwu	No	-	No	-
	TMT	Europarl-v7, Common Crawl corpus, News Commentary , English-French Giga Corpus	28 M	No	-
en/it	Huawei_AGI	"in house data"	6 M	No	-
	MT Learner	"in house data"	67 M	No	-
en/pt	MT Learner	"in house data"	67 M	No	-
en/ru	MT Learner	"in house data"	53 M	No	-
	NVIDIA	No?	-	No?	-
	TMT	Common Crawl corpus, News Commentary, ParaCrawl, Yandex Corpus, Wiki Titles-v2, Back-translated news	92 M	No	-
	Transperfect	No	-	No	-
en/zh	ECNU_PAHT	No	-	No	-
	Huawei_AGI	"in house data"	3 M	No	-
	Huawei_TSC	Corpus supplied by the WMT 2020 News task organizers	16.5M	Yes	150M
	Transperfect	No	-	No	-

Table 8: Overview of out-of-domain (OOD) corpora used by participating teams. Information is self reported through our survey for each selected "best run". (information on the NVIDIA model is inferred from their task paper).

Teams	Runs	BLEU
FJDMATH	run1	0.1453
	run2*	0.1403
Baseline	-	0.1091

Table 9: BLEU scores for the Abstract test set (en2eu). *Indicates the primary run as indicated by the participants.

Teams	Runs	Accuracy	BLEU
FJDMATH	run1	0.16	0.2783
	run2*	0.15	0.2674

Table 10: Scores for the Terminology test set (en2eu). *Indicates the primary run as indicated by the participants.

Teams	Runs	BLEU
Baseline	-	0.3800

Table 11: Performance scores for the test set of summaries of animal experiments (de2en).

sentences. This is the same strategy described in last year's publication (Bawden et al., 2020).

We only considered those teams which either submitted a publication to the workshop or filled in our survey with information about theirs runs. In some few cases, we could not considered some teams for the manual validation, e.g., MT learner for it2en, because the team filled in the survey when the manual validation was already been carried out.

For all teams, we considered the primary run (as indicated by the participants). The only exception was made for the Volctrans team, for which we considered the run with the highest BLEU score, according to the automatic evaluation. The primary runs that we considered in the manual validation are listed below:

- en2de (3 teams): Huawei_AGI (run3), Huawei_TSC (run3), TMT (run1)
- en2es (3 teams): talp_upc (run2), TMT (run1), Transperfect (run2)
- en2fr (3 teams): Huawei_AGI (run3), LISN (run1), TMT (run1)

Teams	Runs	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh
ECNU_PAHT	run1	-	-	-	-	-	-	0.4197
	run2	-	-	-	-	-	-	0.4364
	run3	-	-	-	-	-	-	0.4504*
Haozhiweizi	run1	-	-	-	-	-	-	0.4381*
Huawei_AGI	run1	0.3172	-	0.4531	0.4301	-	-	0.4342
	run2	0.3198	-	0.4424	0.4334	-	-	0.4440
	run3	0.3172*	-	0.4489*	0.4425*	-	-	0.4293*
Huawei_TSC	run1	0.3259	-	-	-	-	-	0.4639
	run2	0.3329	-	-	-	-	-	0.4640*
	run3	0.3259*	-	-	-	-	-	0.4650
JingDong	run1	-	-	-	-	-	-	0.3970*
LISN	run1	-	-	0.3912*	-	-	-	-
	run2	-	-	0.3913	-	-	-	-
	run3	-	-	0.4293	-	-	-	-
NVIDIA NeMo	run1	-	-	-	-	-	0.4139	-
	run2	-	-	-	-	-	0.4112*	-
talp_upc	run1	-	0.4084	-	-	-	-	-
	run2	-	0.4142*	-	-	-	-	-
TMT	run1	0.2765	0.4354	0.4456	-	-	0.3289	-
Transperfect	run1	-	0.5117	-	-	-	0.3686	0.4029
	run2	-	0.5012*	-	-	-	0.3492*	0.4025*
	run3	-	0.4917	-	-	-	-	-
Volctrans	run1	-	-	-	-	-	-	0.4406
	run2	-	-	-	-	-	-	0.4433
	run3	-	-	-	-	-	-	0.4361*
ZengHuiMT	run1	-	-	-	-	-	-	0.4126
Baseline	-	0.2536	0.4027	0.3924	0.4147	0.4304	0.2451	0.3096

Table 12: BLEU scores for "OK" aligned test sentences, from English. For the Volctrans team, we renamed the runs: run1=run1, run2=nnmt, run3=nnmtne. *Indicates the primary run as indicated by the participants.

- en2it (1 team): Huawei_AGI (run3)
- en2ru (3 teams): NVIDIA NeMo (run2), TMT (run1), Transperfect (run2)
- en2zh (6 teams): ECNU_PAHT (run3), Haozhiweizi (run1), Huawei_AGI (run3), Huawei_TSC (run2), Transperfect (run2), Volctrans (run2)
- de2en (4 teams): Huawei_AGI (run3), Huawei_TSC (run3), nrpu-fjwu (run1), TMT (run1)
- es2en (4 teams): nrpu-fjwu (run1), talp_upc (run2), TMT (run1), Transperfect (run2)
- fr2en (4 teams): Huawei_AGI (run3), LISN (run3), nrpu-fjwu (run1), TMT (run1)
- it2en (2 teams): Huawei_AGI (run3)
- pt2en (1 team): MT Learner (run1)
- ru2en (4 teams): NVIDIA NeMo (run1), TMT (run1), Transperfect (run2)
- zh2en (6 teams): ECNU_PAHT (run3), Haozhiweizi (run1), Huawei_AGI (run3), Huawei_TSC (run3), Transperfect (run2), Volctrans (run2)

For each language pair, we generated pairwise combinations of either two teams' primary runs or one primary run and the reference translation. The evaluator first compared pairs of sentences, followed by whole abstracts; the exception was en2zh and zh2en, where only whole abstracts were compared due to the otherwise infeasible large amount of evaluation required. These pairs of translations were manually validated in the Appraise tool (Federmann, 2010) following the same procedure carried out in previous years. For each pair of sentence or abstracts, the aim of the evaluation was to decide whether the translations were of equivalent quality or whether one was better than the other. The results of the manual validation are presented in

Teams	Runs	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en
ECNU_PAHT	run1	-	-	-	-	-	-	0.3232
	run2	-	-	-	-	-	-	0.3232
	run3	-	-	-	-	-	-	0.3546*
Haozhiweizi	run1	-	-	-	-	-	-	0.3713*
Huawei_AGI	run1	0.3956	-	0.4860	0.4570	-	-	0.3943
	run2	0.4132	-	0.4871	0.4569	-	-	0.3785
	run3	0.4048*	-	0.4871*	0.4550*	-	-	0.3934*
Huawei_TSC	run1	0.4230	-	-	-	-	-	0.3828
	run2	0.4258	-	-	-	-	-	0.3921
	run3	0.4310*	-	-	-	-	-	0.3904*
JingDong	run1	-	-	-	-	-	-	0.3041*
LISN	run1	-	-	0.4322	-	-	-	-
	run2	-	-	0.4112	-	-	-	-
	run3	-	-	0.4325*	-	-	-	-
MT Learner	run1	-	-	-	0.4558*	0.5584*	0.4871*	-
	run2	-	-	-	0.4548	0.5685	0.4751	-
NVIDIA NeMo	run1	-	-	-	-	-	0.4918	-
nrpu-fjwu	run1	0.3524*	0.4590*	0.3840*	-	-	-	-
	run2	0.3495	0.4598	0.3921	-	-	-	-
	run3	0.3367	0.4600	0.3772	-	-	-	-
talp_upc	run1	-	0.4194	-	-	-	-	-
	run2	-	0.4194*	-	-	-	-	-
TMT	run1	0.4501	0.5382	0.4928	-	-	0.4061	-
Transperfect	run1	-	0.5237	-	-	-	0.4794	0.3291
	run2	-	0.4991*	-	-	-	0.4769*	0.3212*
	run3	-	0.4969	-	-	-	-	-
Volctrans	run1	-	-	-	-	-	-	0.2911
	run2	-	-	-	-	-	-	0.3796
ZengHuiMT	run1	-	-	-	-	-	-	0.2832
Baseline	-	0.3392	0.3959	0.3796	0.4075	0.4506	0.3115	0.2237

Table 13: BLEU scores for "OK" aligned test sentences, into English. For the Volctrans team, we renamed the runs: run1=base, run2=nnmt. *Indicates the primary run as indicated by the participants.

various tables as summarized below:

- pt2en: Table 14
- en2es and es2en: Table 15
- en2de and de2en: Table 16
- en2fr and fr2en: Table 17
- en2it and it2en: Table 18
- en2zh and zh2en: Table 19
- en2ru and ru2en: Table 20

We identified the item (a system or the reference translation) of each pairwise comparison that performed better (see respective tables) and ran a Wilcoxon Signed-Rank Test from the Python `scipy` library. We consider all comparisons for two particular items over all validated abstracts and sentences, except for skipped ones. The test was calculated for the abstracts and the sentences. We mark in bold in the respective tables the ones that were found to be significant (i.e., p-value< 0.05) and otherwise the systems are considered to be similar. We considered one item superior than the other when either the validation of the abstract of the sentences was statistically significant. For the language pairs validated by two experts (i.e., es2en and pt2en), we only considered one item to be superior than the other when at least two of the four comparisons (2x for the abstracts, 2x for the sentences) were statistically significant.

We ranked the system by assigning points to each item: 3 points if superior to the opponent, 1 point when they have similar quality, and no points if inferior to the opponent. Based on the sum of these points over all comparisons, we ranked the systems and the reference translations as shown be-

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
pt2en	reference-MT Learner	14	3	9	2	112	6	92	14

Table 14: Manual validation for the pt2en MEDLINE abstracts test set. The test set could only be validated with regard to the content of the translation, but not regarding the quality of the English translations.

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2es	TMT-**reference**	8	0	0	**8**	103	9	17	**77**
	TMT-**talp_upc**	8	0	0	**8**	103	6	23	**74**
	TMT-**Transperfect**	8	0	0	**8**	103	3	21	**79**
	reference-talp_upc	8	3	4	1	103	17	77	9
	reference-Transperfect	8	1	7	0	103	8	90	5
	talp_upc-**Transperfect**	8	1	3	4	103	8	75	**20**
es2en	reference-nrpu-fjwu	13/4	7/2	3/1	3/1	107/31	23/**14**	53/13	31/4
	reference-**TMT**	13/4	0/2	4/1	**9**/1	107/31	6/13	57/10	**44**/8
	reference-**Transperfect**	13/4	1/3	4/0	**8**/1	107/31	10/**15**	60/11	**37**/5
	reference-talp_upc	13/4	**9**/3	2/0	2/1	107/31	33/12	49/14	25/5
	nrpu-fjwu-**TMT**	13/4	1/0	2/2	**10**/2	107/31	5/3	67/24	**35**/4
	nrpu-fjwu-**Transperfect**	13/4	0/1	3/1	**10**/2	107/31	4/6	67/22	**36**/3
	nrpu-fjwu-talp_upc	13/4	9/2	1/2	3/0	107/31	31/9	53/17	23/5
	TMT-Transperfect	13/4	3/2	9/2	1/0	107/31	14/**8**	84/22	9/1
	TMT-talp_upc	13/4	**10**/3	3/0	0/1	107/31	**42**/8	61/17	4/6
	Transperfect-talp_upc	13/4	**10**/3	2/0	1/1	107/31	**36**/6	63/19	8/6

Table 15: Manual validation for the en2es and es2en MEDLINE abstracts test set. The better performing MT system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior. For the es2en test set, the values on the left are the validation with regard to the content of the translations, while the ones on the right are regarding the quality of the English translations.

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2de	reference-TMT	9	5	2	1	114	40	38	35
	reference-Huawei_AGI	9	**6**	2	0	114	**51**	40	21
	reference-Huawei_TSC	9	**4**	4	0	114	13	57	**43**
	TMT-Huawei_AGI	9	2	4	2	114	**36**	60	16
	TMT-**Huawei_TSC**	9	0	4	**4**	114	3	59	**51**
	Huawei_AGI-**Huawei_TSC**	9	0	1	**7**	114	5	33	**74**
de2en	**Huawei_TSC**-reference	11	**9**	1	1	93	**31**	44	17
	Huawei_TSC-Huawei_AGI	11	**9**	1	1	93	**38**	50	5
	Huawei_TSC-nrpu-fjwu	11	**11**	0	0	93	**58**	33	2
	Huawei_TSC-TMT	11	6	2	3	93	14	69	10
	reference-Huawei_AGI	11	7	1	3	93	**40**	30	23
	reference-nrpu-fjwu	11	**9**	1	1	93	**47**	28	18
	reference-TMT	11	5	1	5	93	22	47	24
	Huawei_AGI-nrpu-fjwu	11	**10**	1	0	93	**44**	35	14
	Huawei_AGI-**TMT**	11	1	6	4	93	9	56	**28**
	nrpu-fjwu-**TMT**	11	0	2	**9**	93	5	43	**45**

Table 16: Manual validation for the en2de and de2en MEDLINE abstracts test set. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

low (the points obtained are shown in parentheses):

- en2de: Huawei_AGI (0) < TMT (4) = reference (5) < Huawei_TSC (7)
- en2es: TMT (0) < reference (4) = talp_upc (4) < Transperfect (7)
- en2fr: Huawei_AGI (2) = TMT (2) < LISN (5) < reference (6)
- en2it: Huawei_AGI (1) = reference (1)

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2fr	**reference**-LISN	16	10	1	3	100	**58**	13	18
	reference-Huawei_AGI	16	**14**	0	2	100	**65**	18	17
	reference-TMT	16	**14**	1	1	100	**65**	18	17
	LISN-Huawei_AGI	16	6	1	7	100	37	29	23
	LISN-TMT	16	4	4	6	100	30	30	29
	Huawei_AGI-TMT	16	7	7	2	100	29	43	28
fr2en	nrpu-fjwu-**Huawei_AGI**	12	2	0	**10**	79	15	19	**42**
	nrpu-fjwu-LISN	12	4	1	7	79	19	26	33
	nrpu-fjwu-reference	12	3	1	8	79	28	13	37
	nrpu-fjwu-**TMT**	12	1	0	**11**	79	9	24	**45**
	Huawei_AGI-LISN	12	7	0	5	79	27	30	21
	Huawei_AGI-reference	12	7	1	4	79	**37**	20	21
	Huawei_AGI-TMT	12	3	3	6	79	17	36	25
	LISN-reference	12	6	2	4	79	31	26	21
	LISN-TMT	12	3	0	9	79	16	36	26
	reference-**TMT**	12	3	2	7	79	19	18	**41**

Table 17: Manual validation for the en2fr and fr2en MEDLINE abstracts test set. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2it	Huawei_AGI-reference	10	6	0	4	100	38	35	27
it2en	Huawei_AGI-reference	11	4	0	7	102	32	40	30

Table 18: Manual validation for the en2it and it2en MEDLINE abstracts test sets. For the it2en test set, only the translation from Italian into English was assessed, but not the quality of the English text.

- en2ru: NVIDIA Nemo (3) = reference (3) = TMT (3) = Transperfect (3)
- en2zh: ECNU_PAHT (3) < Huawei_AGI (4) = Haozhiweizi (4) = Transperfect (4) < Huawei_TSC (9) < Volctrans (12) < reference (16)
- de2en: nrpu-fjwu (0) < Huawei_AGI (3) < reference (7) < TMT (8) < Huawei_TSC (10)
- es2en: nrpu-fjwu (2) = reference (2) = talp_upc (2) < TMT (10) = Transperfect (10)
- fr2en: nrpu-fjwu (2) = reference (2) < LISN (4) < Huawei_AGI (8) = TMT (8)
- it2en: reference (1) = Huawei_AGI (1)
- pt2en: reference (1) = MT Learner (1)
- ru2en: TMT (0) < Transperfect (4) < reference (5) < NVIDIA NeMo (7)
- zh2en: Huawei_AGI (5) < Haozhiweizi (6) = Volctrans (6) = Huawei_TSC (6) = ECNU_PAHT (6) = reference (6) < Transperfect (8)

Abstracts for en2eu (Osagaiz + Gaceta) were manually validated following the same approach, but only at the sentence level. As there was one submission for this language pair, we only generated a pairwise combination of the participant's run and the reference. The run with the highest BLEU score was selected for validation:

- en2eu (1 team): FJDMATH (run1)

The translations were evaluated by three annotators using the Appraise tool, and the averaged results are presented in Table 21. The ranking based on the points is as follows:

- en2eu: FJDMATH (0) < reference (3)

7 Discussion

7.1 Quality of the MT evaluation process.

Marie et al. (2021) introduced guidelines for the evaluation of MT quality, based on four criteria:

Language	Pair	Abstracts			
		Total	**A>B**	**A=B**	**A<B**
en2zh	ECNU_PAHT-Huawei_AGI	13	3	1	9
	ECNU_PAHT-Transperfect	13	5	0	8
	ECNU_PAHT-**Volctrans**	13	1	0	**12**
	ECNU_PAHT-Haozhiweizi	13	4	3	6
	ECNU_PAHT-**Huawei_TSC**	13	1	2	**10**
	ECNU_PAHT-**reference**	13	0	1	**12**
	Huawei_AGI-Transperfect	13	8	1	4
	Huawei_AGI-Volctrans	13	2	3	8
	Huawei_AGI-Haozhiweizi	13	7	1	5
	Huawei_AGI-**Huawei_TSC**	13	2	1	**10**
	Huawei_AGI-**reference**	13	2	2	**9**
	Transperfect-**Volctrans**	13	2	1	**10**
	Transperfect-Haozhiweizi	13	7	1	5
	Transperfect-Huawei_TSC	13	3	0	10
	Transperfect-**reference**	13	2	1	**10**
	Volctrans-Haozhiweizi	13	**10**	1	2
	Volctrans-Huawei_TSC	13	3	6	4
	Volctrans-reference	13	3	2	8
	Haozhiweizi-Huawei_TSC	13	4	1	8
	Haozhiweizi-**reference**	13	2	0	**11**
	Huawei_TSC-**reference**	13	2	1	**10**
zh2en	Haozhiweizi-Volctrans	19	11	1	7
	Haozhiweizi-Huawei_TSC	19	10	3	6
	Haozhiweizi-reference	19	9	4	5
	Haozhiweizi-ECNU_PAHT	19	10	2	7
	Haozhiweizi-Huawei_AGI	19	10	5	4
	Haozhiweizi-Transperfect	19	4	6	9
	Volctrans-Huawei_TSC	19	3	8	8
	Volctrans-reference	19	7	3	8
	Volctrans-ECNU_PAHT	19	8	4	7
	Volctrans-Huawei_AGI	19	9	3	7
	Volctrans-Transperfect	19	5	3	11
	Huawei_TSC-reference	19	6	5	7
	Huawei_TSC-ECNU_PAHT	19	9	2	8
	Huawei_TSC-Huawei_AGI	19	10	1	8
	Huawei_TSC-Transperfect	19	7	4	8
	reference-ECNU_PAHT	19	12	0	6
	reference-Huawei_AGI	19	9	2	7
	reference-Transperfect	19	7	4	7
	ECNU_PAHT-Huawei_AGI	19	8	4	7
	ECNU_PAHT-Transperfect	19	3	6	10
	Huawei_AGI-**Transperfect**	19	3	3	**13**

Table 19: Manual validation for the en2zh and zh2en MEDLINE abstracts test set. Only the abstracts were validated. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

(1) use of an evaluation method in addition to/in lieu of BLEU, (2) use of statistical significance testing to compare systems, (3) direct computation of scores instead of copying from previous experiments and (4) comparison of systems only if the same training, validation and test sets have been used, as well as the same pre-processing steps.

The evaluation carried out in this task is compliant with criteria (1-3). However, participants are free to use their choice of training corpus, validation corpus and pre-processing methods. This approach was selected to foster participant creativity and set a lower entry cost to the task. It is a limitation in the comparability of the systems submitted for this task. As a mitigation strategy, we encourage participants to also submit detailed descriptions of system particulars to provide transparency on the material and methods used.

A future edition of the task could introduce a "constrained" track where pre-processed training/validation sets would be supplied to be used exclusively (as in the WMT news translation task (Barrault et al., 2020)).

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2ru	TMT-Transperfect	16	6	1	9	95	15	61	19
	TMT-NVIDIA NeMo	16	4	1	11	95	20	47	25
	TMT-reference	16	6	0	10	95	27	42	25
	Transperfect-NVIDIA NeMo	16	6	5	5	95	26	52	15
	Transperfect-reference	16	4	6	6	95	22	52	21
	NVIDIA NeMo-reference	16	2	9	5	95	13	64	15
ru2en	**Transperfect**-TMT	16	**10**	6	0	109	**42**	56	9
	Transperfect-reference	16	2	9	5	109	25	65	19
	Transperfect-**NVIDIA NeMo**	16	0	10	**6**	109	7	87	15
	TMT-**reference**	16	0	3	**13**	109	10	56	**41**
	TMT-**NVIDA NeMo**	16	0	1	**15**	109	4	61	**42**
	reference-NVIDIA NeMo	16	2	9	5	109	16	71	22

Table 20: Manual validation for the en2ru and ru2en MEDLINE abstracts test sets. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

Language	Pair	Sentences			
		Total	**A>B**	**A=B**	**A<B**
en2eu	reference-FJDMATH	100	**61**	16	23

Table 21: Manual validation for the en2eu (Osagaiz and Gaceta) abstracts test set. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior. The values show the validation performed by the Basque native speakers (averaged over three annotators).

7.2 Quality of the system translations

We report below some of the major observations collected throughout the manual validation of the selected runs and the reference translations.

7.2.1 MEDLINE test sets

de (from en) The perceived quality of translations was high, with a high proportion of perfect or near perfect translations. The translation quality between participating systems differ only by small nuances. For example, translations differ only by word order or use different synonyms for a specific medical term. All participating systems had problems translating abbreviations. For instance, "image quality (IQ)" becomes "Bildqualität (IQ)" instead of BQ. One participant could not generate umlauts (e.g., ä,ö,ß, ...) and another participant produced only lowercased text. Both problems lead to slightly reduced quality of translations.

en (from de) Overall, the translation quality was high. Most translated sentences were understandable, except in cases where the original German sentences were too long to translate correctly. In some cases, the translations captured the intended meaning, but took information from different sentences, or even used synonyms, which were not direct literal translations of words, such as "neurosensory retina" for "macular region". There were also some translations from the first person point of view, rather than the impersonal, for a more personal touch. If such examples represent MT outputs rather than the reference translations, the quality of translation is approaching native speaker level.

Some texts contained small errors, which should be easy to avoid, such as capitalization of the first word after "e.g.", the use of lower case letters for well-known abbreviations like "AR" for augmented reality, proper nouns ("Marburg heart score"), and gene names (PD-1). Also a repetition of words in common expressions like the German *wie z. B.* should not have been translated as "e.g. For example". Using the same word twice in a sentence could have been avoided: "Relapse is defined as the recurrence" instead of "Recurrence is defined as the recurrence". Interestingly, a translation actually corrected a capitalization error in the original German text, from *l. reuteri* to *L. reuteri*, for the genus *Lactobacillus* in the scientific name of the bacteria.

The correct translation of medical terms also

proved to be difficult in cases such as "centrum" instead of "ventrum", "neurology" instead of "urology", or "endourolgy" instead of "endourology". Medical terms pertaining to a specific field, were also difficult to translate properly, such as the German *Pluszeichen* to the English "plus disease" in the context of retinal disease and "fusion biopsy" to describe the method of using both magnetic resonance imaging and ultrasound to take prostate biopsies.

es (from en) Quality of translations is improving every year and in many cases it is difficult to identify which translations are machine made.

There have been cases in which abbreviations were not translated correctly (e.g. *HGS* vs *FPM* for *fuerza de prensión manual*) and some times specific terms were not translated, e.g. *receiver operating characteristic curve*. Only in a few cases word gender was different to the article one.

There are examples of words that are not translated properly for instance *adnexal* has been translated as *adnexiales* instead of *anexas* by one of the teams.

In addition to individual sentences, the manual evaluation included abstracts as well. Since translations were sentence based, there were cases in which there were misaligned information between the content of the sentences in the abstract, even if the translated sentences were perfectly fine in their own.

We identified that a team might have been missing accents on vowels and special letters such as *ñ*, which seemed to indicate that the translation was machine made.

fr (from en) The overall quality of translation was high. We noted that many of the sentences compared were identical or nearly identical. In many cases, the translations selected as superior were chosen based on small nuances, such as capitalization, typography, ordering of words or sentence structure that appeared more adequate to a native speaker, while causing no difference in the understanding of the text. A few terms or acronyms were sometimes untranslated but the resulting text could be understood (for example, *incidentaloma* was used instead of *incidentalome*). Erroneous disambiguation was observed in some translations for one abstract discussing pressure at the fingertips, where *numérique* was used instead of *digital*. At the abstract level, some vocabulary consistency issues could be evidenced. For example, one translation used the synonyms *sclérodermie systémique* and *sclérose systémique* alternatively as translation for "systemic sclerosis". While individual sentences were correctly translated, it created confusion at the abstract level, compared to the "reference" translation, which used the term *sclérodermie* throughout. Consistently with the 2020 edition, arbitration between sentences exhibiting a fluency or grammatical flaw vs. a semantic or clinical flaw was conducted as favoring the semantic or clinical correctness. However, the nature of the "reference" translation (which is often not produced by professional translators and does not necessarily provide straight forward sentence-by-sentence translations (Névéol et al., 2020)) introduces bias and difficulty in the evaluation: highly fluent text with some semantic distance with the "original" sentence to be translated can sometimes be easily identified as the reference text. It is difficult to arbitrate between this high quality text and the machine translation that will attempt to be semantically closer while exhibiting language flaws.

en (from fr) Translation quality was generally very high, with some variation in the quality depending on the topic of the abstract being translated (most systems struggled with the more literary text from a sociology abstract). This meant that many of the decisions were, as with the other language pairs, based on preferences and formatting rather than differences in meaning (punctuation, capitalisation, minor grammar mistakes).

The most serious errors observed were with the translation of specific terms, such as illnesses and drugs. They were particularly prevalent for acronyms, which were sometimes not translated and sometimes poorly translated (another more common acronym being used instead, e.g. *ADHD* instead of *ADPKD*). A few tricky sentences revealed the risk of major semantic errors resulting from seemingly small and localised errors. We give two such examples here. Firstly, concerning temporality, several systems translated French *puis* 'then' as English *and* in *un anticoagulant puis l'aspirine* 'an anti-coagulant then aspirine', a sentence for which the order in which drugs are given may be fundamental. Secondly, many systems stumbled on the translation of French *cela ne s'accompagne pas d'une attention égale au rôle de l'écoute* 'this is not accompanied by equal attention to the role of listening', inverting the order of the two underlined

words, resulted in the opposite meaning (i.e. listening receiving more rather than less attention).

As noted above for en2fr, despite very high MT quality, reference translations are often still easily identifiable due to them being less literal. This means that they are often characterized by better word choice and more natural syntax, but it can also mean that they are less adequate because of missing information or even additional details not present in the French text.

it (from en) The quality of the translation was on average high, probably higher than 2020. Some of the sentences compared were almost identical.

From a terminological viewpoint, it is possible to identify some inaccuracies in the choice of translating terms in the target language. For example, the term 'malignancies' was translated by one system with *tumori* (corresponding to the English 'tumors') having a broader meaning than *neoplasie maligne* ('malignant neoplasms'). Furthermore, cases of erroneous choices of the translating terms can be identified. The adjective 'unpreventable' was translated as *non prevedibile* ('unpredictable') instead of *non evitabile*, thus causing the transmission of an incorrect information in the target text.

Another error can be identified in the choice of the generic verb 'consider'. In the case of the sentence 'total laryngectomy should be considered [...]', the construction was wrongly translated as *la laringectomia dovrebbe essere considerata il trattamento di scelta*, that is the 'Laryngectomy should be considered the treatment of choice'. It is also possible to identify cases of non-translation in the Italian text (for example the name of the city 'Zurich' remained untranslated) and the presence of anglicisms as *imaging diagnostico* chosen as the translation of 'diagnostic imaging', although the Italian equivalent *diagnostica per immagini* is commonly used in the target language.

Moreover, the term 'livestock' was translated with *mandria* 'herd', *bestiame* 'livestock' or *allevamento* 'farm'. One interesting case was the term 'blacks', which was translated with *non bianchi* (non-whites) instead of the more frequently used *neri*.

Finally, from a syntactic point of view, there were a couple of examples where the syntactic tree was built erroneously: for example, the phrase 'incidental thyroid cancer rates' was translated as *i tassi di carcinoma tiroideo incidentale* ('rates of incidental thyroid cancer'); another example is '4 cm lobule contoured mass' translated as *massa sagomata di 4 cm del lobulo* ('4 cm contoured mass of the lobule').

zh (from en) The quality of translation was high overall. The primary reason for awkward translations was word order, since English and Chinese employ different word orders not only at the word level, but also at the phrase level. Consider this example, where the source text was *surveillance and early warning of infectious diseases in China*. A good translation first needed to adjust word order within *infectious diseases in China* to yield 中国传染病 (the order is *China* then *infectious diseases*). Then phrase order also needed to be adjusted to yield 中国传染病监测预警能力 (the order is *China infectious diseases* then *surveillance and early warning*). Some translations failed to make these necessary adjustments, such that the Chinese translation in the original English word order rendered the translation awkward or even unintelligible.

Another source of deficiency was translations that were too literal. For instance, *under-reporting* was most often translated as 报告不足 (*insufficient reporting*), though a more native, conventional wording would actually be 漏报 (*omitted reporting*). Consider another example, *appraised persons* in the context of a study of familial relationships. While the reference translation 被鉴定人 was the most fitting, some teams' translation 被评估者 (a person to be evaluated) was also a good fit. However, another translation such as 评估人员 (evaluation personnel) was outright incorrect, as the meaning went from "a passive person being evaluated" to "an active person evaluating someone else".

en (from zh) The quality of translation was also high and noticeably better than last year. Where some translations last year were unintelligible, such cases have disappeared this year. In addition, there was a range of translation qualities last year, but this year every team's translation quality was high.

This year, the aspects that distinguish a better translation from a worse one are more subtle. Firstly, Chinese sentences as delimited by the Chinese full stop "。" are often equivalent to short paragraphs in English. A good translation should therefore split a Chinese sentence where necessary into multiple English sentences. Secondly, a technical term may have synonyms (e.g. *acetabulum*

labrum and *acetabular lip*), and a better translation should use one synonym consistently within the same abstract instead of mixing different ones. Thirdly, a good translation should use the most fitting wording, a task that requires good understanding of sentence context as well as domain knowledge. Consider this example, where 夫妻婚姻关系 (*marital relationship of married couples*) and 双向关联性 (*bi-directional correlation*) occur within the same sentence. A good translation can tease out *relationship* and *correlation*, but a worse translation simply uses *relationship* for both occurrences. This year, the better translations achieved the above three aspects, though rarely all three at once in the same translation.

en (from pt) While there was no significant difference between the automatic translations from the MT Learner team and the reference translation, we highlight some situations in which one was considered better than the other. On one hand, some mistakes were very subtle, such as a typo in a word (e.g., "verage" instead of "average"), or an inappropriate capitalization of a word. On the other hand, there were some semantic mistakes related to the translation of the sentences. For instance, the passage "insuficiência do glúteo médio esteve presente em todos os sujeitos" was translated as "the gluteus medius was insufficient in all patients".

Overall Based on these comments, many language pairs would benefit from a visual feature highlighting differences in the translations in the interface to focus the analysts' attention on often small differences. It could also be relevant to focus manual evaluation on a number of targeted linguistic features that seem to remain difficult (based on 2020 and 2021 observations), such as (a) translation of acronyms (b) vocabulary/grammatical consistency throughout a document (c) translation of numerical data. This might help make the manual evaluation more comparable between language pairs. However, it raises the question of the method to use for the selection of sentences/passages exhibiting the desired phenomena.

7.2.2 Osagaiz/Gaceta abstract test sets (en2eu)

In general, despite the fact that in the manual evaluation FJDMATH ranked below the references, the translations generated by the system were good, containing sentences with high-level of fluency and high adequacy with respect to the source. Similar to what has been observed in other language pairs, the system sometimes struggled with the translation of acronyms. For example, "non-motor symptoms (NMS)" should be translated to "sintoma ez motor (SEM)" but the participant's system translated it as "sintoma ez-motor (NMS)"; or "amiotrophic lateral sclerosis (ALS)" should be "alboko esklerosi amiotrofikoa (AEA)".

On the other hand, sometimes the reference translation in Basque contained extra information that was not present in the source English sentence. In these cases the additional information is contained in the context (i.e. surrounding sentences in the abstract). This can penalize the BLEU score of a correct sentence-level translation. For example, the source sentence "*Important hormonal changes happen during pregnancy and lactation*" was correctly translated by the system to "*Hormona aldaketa garrantzitsuak gertatzen dira haurdunaldian eta laktazioan.*". However, the reference translation also mentions "physiological changes in the body" (i.e. "*Haurdunaldi eta edoskitzaroan zehar,* ***gorputzeko maila askotan aldaketa fisiologikoak eragingo dituzten*** *gorabehera hormonal nabariak gertatuko dira*"). Document or abstract level translation systems could potentially alleviate this problem by leveraging contextual information from surrounding sentences.

7.2.3 Terminology test sets (en2eu)

The participating team (FJDMATH) had more difficulty in the translation of ICD-10 code descriptions (16% accuracy), particularly if we compare them with the results obtained by many teams last year (∼70% accuracy). Note that ICD-10 codes included in the test sets every year are different, but the performance difference is big considering there was more in-domain training data available this year. Some of the common mistakes observed in the system are word repetition (e.g. *hortz-posizioko anomaliak, hortz edo hortz guztiz eruptatuen posizioa* - "tooth position anomalies, tooth and tooth"), not translating an English word (e.g. *tidal-wave* instead of *olatu erraldoi*) or low adequacy (i.e. *huts egin du jaioberrian irabaztean* / (en) *missed when winning the newborn* where it should be *jaioberriaren garapeneko atzerapen* / (en) *Failure to thrive in newborn*). It is possible that the system was not sufficiently fine-tuned for the technical and specific language employed in ICD-10 code descriptions.

8 Conclusions

Our sixth edition of the WMT Biomedical Translation addressed a total of eight language pairs and three types of documents. One more time, we could assess the performance of current MT technology for the translation of biomedical textual resources. Further, we could attract the attention of many teams and received submissions for most of our test sets.

Similar as in the more recent editions of the shared task, participating system could perform better than the reference translation for many of the language pairs. However, this is still a challenge for en2zh. In future editions of this challenge, we aim at releasing more resources, especially additional training data, adding new language pairs, and considering a variety of test sets.

Acknowledgements

We would like the thank the participants Virginia Adams (NVIDIA NeMo), Cao Jun (Volctrans/ByteDance), Wei Peng (Huawei_AGI), and multiple individuals in the Huawei_TSC team for supporting us in the manual validation. The Academy of Medical Sciences of Bilbao and the Gaceta Médica de Bilbao have collaborated in WMT 2021 by providing us with their documents.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 biomedical translation shared task: Basque, Italian and Russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation*, pages 660–687, Online. Association for Computational Linguistics.

Bettina Bert, Antje Dörendahl, Nora Leich, Julia Vietze, Matthias Steinfath, Justyna Chmielewska, Andreas Hensel, Barbara Grune, and Gilbert Schönfelder. 2017. Rethinking 3R strategies: Digging deeper into AnimalTestInfo promotes transparency in in vivo biomedical research. *PLOS Biology*, 15(12):1–20.

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.

Christian Federmann. 2010. Appraise: An Open-Source Toolkit for Manual Phrase-Based Evaluation of Translations. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 1731–1734, Valletta, Malta.

Antonio Jimeno Yepes, Aurelie Neveol, 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.

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.

DA Lindberg, BL Humphreys, and AT McCray. 1993. The unified medical language system. *Yearb Med Inform*, 1:41–51.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online. Association for Computational Linguistics.

Ander Martínez. 2021. The Fujitsu DMATH submissions for WMT21 News Translation and Biomedical Translation Tasks. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Sumbal Naz, Sadaf Abdul Rauf, and Sami Ul Haq. 2021. FJWU participation for the WMT21 Biomedical Translation Task. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Aurélie Névéol, Antonio Jimeno Yepes, and Mariana Neves. 2020. MEDLINE as a parallel corpus: a survey to gain insight on French-, Spanish- and Portuguese-speaking authors' abstract writing practice. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3676–3682, Marseille, France. European Language Resources Association.

Mariana Neves, Daniel Butzke, Antje Dörendahl, Nora Leich, Benedikt Hummel, Gilbert Schönfelder, and Barbara Grune. 2019. Overview of the CLEF eHealth 2019 Multilingual Information Extraction. In *Tenth International Conference of the CLEF Association (CLEF 2019)*.

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: Shared Task Papers*, pages 324–339. Association for Computational Linguistics.

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)*, 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Bardia Rafieian and Marta Ruiz Costa-Jussà. 2021. High frequent in-domain words segmentation and forward translation for the WMT21 Biomedical task. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams, and Oleksii Kuchaiev. 2021. NVIDIA NeMo's Neural Machine Translation Systems for English ↔ German and English ↔ Russian News and Biomedical Tasks at WMT21. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Weixuan Wang, Wei Peng, Xupeng Meng, and Qun Liu. 2021a. Huawei AARC's submissions to the WMT21 Biomedical Translation Task: Domain Adaption from a Practical Perspective. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Xing Wang, Zhaopeng Tu, and Shuming Shi. 2021b. Tencent AI Lab Machine Translation Systems for the WMT21 Biomedical Translation Task. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Sarah Wilde. 2021. African languages to get more bespoke scientific terms. *Nature (news)*, pages 469–470.

Hao Yang, ZhanglinWu, Zhengzhe Yu, Xiaoyu Chen, Daimeng Wei, Zongyao Li, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Chuanfei Xu, Min Zhang, and Ying Qin. 2021. HW-TSC's submissions to the WMT21 Biomedical Translation Task. In *Proceedings of the Sixth Conference on Machine Translation: Shared Task Papers*.

Findings of the WMT 2021 Shared Task on Quality Estimation

Lucia Specia,[1,2] Frédéric Blain,[2,3] Marina Fomicheva,[2] Chrysoula Zerva,[4,5]
Zhenhao Li,[1] Vishrav Chaudhary,[6] André F. T. Martins[4,5,7]

[1]Imperial College London, [2]University of Sheffield, [3]University of Wolverhampton,
[4]Instituto de Telecomunicações, [5]Instituto Superior Técnico, [6]Facebook AI, [7]Unbabel
{l.specia,m.fomicheva}@sheffield.ac.uk, f.blain@wlv.ac.uk
chrysoula.zerva@tecnico.ulisboa.pt, zhenhao.li18@imperial.ac.uk,
vishrav@fb.com, andre.t.martins@tecnico.ulisboa.pt

Abstract

We report the results of the WMT 2021 shared task on Quality Estimation, where the challenge is to predict the quality of the output of neural machine translation systems at the word and sentence levels. This edition focused on two main novel additions: (i) prediction for unseen languages, i.e. zero-shot settings, and (ii) prediction of sentences with catastrophic errors. In addition, new data was released for a number of languages, especially post-edited data. Participating teams from 19 institutions submitted altogether 1263 systems to different task variants and language pairs.

1 Introduction

The 10th edition of the shared task on Quality Estimation (QE) builds on its previous editions to further benchmark methods for estimating the quality of neural machine translation (MT) output at runtime, without the use of reference translations. It includes the (sub)tasks of word-level and sentence-level estimation. The document-level task was removed from this edition, since it has not attracted many participants in previous editions. Important elements introduced this year are: a new sentence-level task where sentences are annotated with a binary label reflecting whether or not they contain a critical error that could lead to catastrophic consequences; new test data for languages that are not covered by any training set for zero-shot prediction (both direct assessment and post-editing based labels, at sentence and word levels. scores instead of labels based on post-editing; a new multilingual sentence-level dataset mainly from Wikipedia articles, where the source articles can be retrieved for document-wide context; the availability of NMT models to explore system-internal information for the task.

In addition to advancing the state-of-the-art at all prediction levels, our main goals are:

- To extend the MLQE-PE public benchmark datasets;
- To investigate new language independent approaches especially for zero-shot prediction;
- To study the feasibility of unsupervised approaches especially for zero-shot prediction; and
- To create a new task focusing on critical error detection.

We have three subtasks: Task 1 aims at predicting a variant of DA scores at sentence level (Section 2.1); Task 2 aims at predicting post-editing effort scores at both sentence and word levels, i.e. words that need editing, as well as missing words and incorrect source words (Section 2.2); Task 3 aims at predicting a binary label at sentence level to indicate whether the sentence contains one or more critical errors (Section 2.3).

Tasks make use of large datasets annotated by professional translators with either 0-100 DA scoring, post-edition, or critical error flagging. The text domains and languages vary for each subtask. Neural MT systems were built on freely available data using an open-source toolkits to produce translations. We provide new training and test datasets for Tasks 2 and 3, new test sets for Task 1, as well as new *zero-shot* test sets for Tasks 1 and 2. The datasets and models released are publicly available. Participants are also allowed to explore any additional data and resources deemed relevant.

Baseline systems were entered in the platform by the task organisers (Section 3). The shared task uses CodaLab as submission platform, where participants (Section 4) could submit up to 30 systems for each task and language pair, except for the multilingual track of Tasks 1 and 2 (up to 10 systems). Results for all tasks evaluated according to standard metrics are given in Section 5, which this year also included model size. A discussion on the main

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 684–725

goals and findings from this year's task is presented in Section 6.

2 Subtasks

In what follows, we give a brief description for each subtask, including the datasets provided for them.

2.1 Task 1: Predicting sentence-level DA

This task consists in scoring translation sentences according to their perceived quality score – which we refer to as direct assessment (DA). For that, we use the same training sets as last year's Task 1 (Specia et al., 2020), and provided new test sets for all low, medium and high-resource languages:

- English→German (En-De),
- English→Chinese (En-Zh),
- Russian→English (Ru-En),
- Romanian→English (Ro-En),
- Estonian→English (Et-En),
- Sinhala→English (Si-En), and
- Nepali→English (Ne-En).

This data was produced in the same way as the data for last year, with sentences sample from Wikipedia (or Wikipedia and Reddit for Ru-En) and translated by a fairseq Transformer (Ott et al., 2019) bilingual model.

In addition, we provide new test sets for four other languages, for which training data was not provided. The goal was to test the performance of QE models under zero-shot settings. The new test sets contain source Wikipedia sentences sampled in the same way as the previous data, but translated by the ML50 fairseq multilingual Transformer model (Tang et al., 2020),[1] which had been found to perform well especially for low-resource languages. The following languages were used:

- English→Czech (En-Cs),
- English→Japanese (En-Ja),
- Pashto→English (Ps-En), and
- Khmer→English (Km-En),

All translations were manually annotated for perceived quality, with a quality label ranging from 0 to 100, following the FLORES guidelines (Guzmán et al., 2019). According to the guidelines given to annotators, the 0-10 range represents an incorrect translation; 11-29, a translation with few correct keywords, but the overall meaning is different from the source; 30-50, a translation with major mistakes; 51-69, a translation which is understandable and conveys the overall meaning of the source but contains typos or grammatical errors; 70-90, a translation that closely preserves the semantics of the source sentence; and 91-100, a perfect translation. DA scores were standardised using the z-score by rater. Participating systems are required to score sentences according to z-standardised DA scores. Statistics on the dataset are shown in Table 1. This dataset part of the MLQE-PE dataset and more details are given in Fomicheva et al. (2020). The complete data can be downloaded from the public repository.[2]

Participation was encouraged for each language pair and also for the **multilingual variant** of the task, where submissions had to include predictions for all six Wikipedia-based language pairs (all except Ru-En). The latter aimed at fostering work on language-independent models, as well as models that can leverage data from multiple languages.

2.2 Task 2: Predicting post-editing effort

This task concerns scoring translations according to the proportion of the words that need to be edited to obtain a correct translation. The scores are generated using Human-mediated Translation Edit Rate (HTER) (Snover et al., 2006), i.e. calculating the minimum edit distance between the machine translation and its manually post-edited version, as well as detecting where errors are in the translation of source sentences. It comprises two sub-tasks, a sentence level one where the targets are the HTER scores per segment and a word level task where the targets are word level OK/BAD tags to signify the correctness of words and gaps in the source and translation sentences. Both sub-tasks use the same languages pairs and splits described for Task 1 in Table 1. Details on the data, such as label distributions, can be found in Fomicheva et al. (2020).

Sentence-level post-editing effort The label for this task is the percentage of edits that need to be fixed (HTER). The data used for this task is the PE annotations and corresponding HTER scores from the MLQE-PE dataset (Fomicheva et al., 2020). HTER labels are computed using TERCOM,[3] with

[1] https://github.com/pytorch/fairseq/tree/master/examples/multilingual

[2] https://github.com/sheffieldnlp/mlqe-pe

[3] https://github.com/jhclark/tercom

Language Pairs	Sentences Train / Dev / Test21	Tokens Train / Dev / Test21	DA	PE	CE
En-De	7,000 / 1,000 / 1,000	114,980 / 16,519 / 16,545	✓	✓	
En-Zh	7,000 / 1,000 / 1,000	115,585 / 16,307 / 16,637	✓	✓	
Ru-En	7,000 / 1,000 / 1,000	82,229 / 11,992 / 11,650	✓	✓	
Ro-En	7,000 / 1,000 / 1,000	120,198 / 17,268 / 17,359	✓	✓	
Et-En	7,000 / 1,000 / 1,000	98,080 / 14,423 / 14,044	✓	✓	
Ne-En	7,000 / 1,000 / 1,000	104,934 / 15,144 / 15,017	✓	✓	
Si-En	7,000 / 1,000 / 1,000	109,515 / 15,708 / 15,709	✓	✓	
Ps-En	– / – / 1,000	– / – / 27,045	✓	✓	
Km-En	– / – / 1,000	– / – / 21,981	✓	✓	
En-Ja	– / – / 1,000	– / – / 20,626	✓	✓	
En-Cs	– / – / 1,000	– / – / 20,394	✓	✓	
En-Cs	7,476 / 1,000 / 1,000	122,275 / 16,270 / 16,106			✓
En-De	7,878 / 1,000 / 1,000	127,778 / 16,114 / 16,371			✓
En-Ja	7,658 / 1,000 / 1,000	126,307 / 16,400 / 16,412			✓
En-Zh	6,859 / 1,000 / 1,000	110,717 / 16,283 / 15,989			✓

Table 1: Statistics of the data used for Task 1 (DA), Task 2 (PE) and Task 3 (CE) (last four rows). The number of tokens is computed based on the source sentences.

default settings (tokenised, case insensitive, exact matching only) with scores capped to 1.

Word-level errors This sub-task focuses on detecting word-level errors in the MT output. The goal in this case is to annotate each token with binary correctness (OK/BAD) tags. The token-level annotations include the annotation of gaps, which allows us to account for omission errors. All annotations are produced with respect to a post-edited sentence, which is treated as the ground truth reference. Similarly to the sentence-level tasks, the MLQE-PE data is used for all language pairs (see Table 1). The following types of labels are used:

- **Source side:** Each word in the source side is labelled as OK (correctly translated) or BAD (caused a translation error).
- **Target side:** Each word in the target side is labelled as OK (a correct translation) or BAD (should be replaced or deleted). Additionally, we consider gap 'tokens' at the beginning of the sentence, at the end and between each two words. They are labelled OK if no word should be inserted in that position (according to the post-edited version), and BAD otherwise.

2.3 Task 3: Predicting Catastrophic Errors

This is a new task introduced this year. It aims to predict a sentence-level binary score indicating whether a translation contains (at least one) critical error (CE). Translations with such errors are defined as translations that deviate in meaning as compared to the source sentence in such a way that they are misleading and may carry health, safety, legal, reputation, religious or financial implications. Meaning deviations from the source sentence can happen in three ways:

- **Mistranslation**: critical content is translated incorrectly into a different meaning, or not translated (i.e. it remains in the source language) or translated into gibberish.
- **Hallucination**: critical content that is not in the source is introduced in the translation, for example, profanity words are introduced that were not in the source.
- **Deletion**: critical content that is in the source sentence is not present in the translation. For example, the source sentence may contain a negation or hateful word that is removed in the translation.

We focus on the following set of critical error categories:

- **TOX.** Deviation in toxicity (hate, violence or profanity) be against an individual or a group (a religion, race, gender, etc.). This error can happen because toxicity is introduced in the translation when it is not in the source, deleted in the translation when it was in the source, or mistranslated into different (toxic or not) words, or not translated at all (i.e. the toxicity remains in the source language or it is transliterated).

- **SAF.** Deviation in health or safety risks, i.e. the translation contains errors that may bring a risk to the reader. This issue can happen because content is introduced in the translation when it is not in the source, deleted in the translation when it was in the source, or mistranslated into different words, or not translated at all (i.e. it remains in the source language).

- **NAM.** Deviation in named entities. A named entity (people, organisation, location, etc.) is deleted, mistranslated by either another incorrect named entity or a common word or gibberish, or left untranslated when it should be translated, or transliterated where the transliteration makes no sense in the target language (i.e. the reader cannot recover the actual named entity from it), or introduced when it was not in the source text. If the named entity is translated partially correctly but one can still understand that it refers to the same entity, it should not be an error.

- **SEN.** Deviation in sentiment polarity or negation. The MT either introduces or removes a negation (with or without an explicit negation word), or reverses the sentiment of the sentence (e.g. a negative sentence becomes positive or vice-versa). We note that SEN errors do not always involve a full negation, for example, replacing "possibly" with "with certainty" constitutes a SEN error.

- **NUM.** Deviation in units/time/date/numbers. The MT translated a number/date/time or unit incorrectly (or translated it as gibberish), or removed it, which could lead someone to miss an appointment, get lost, etc.

Data for this task was annotated at the word/span level by professional translators not only for the presence of an error, but also with the error category. Each instance was annotated by three professional translators using a modified version of MT-EQuAl.[4] We instructed the translators to ignore other translation errors, be them critical (there may be other types of critical errors outside these five categories) or non-critical, e.g. minor grammatical or typographical errors. We also instructed them to indicate source sentences that were unintelligible, or translation sentences that contained too many errors to be annotated. Figure 1 shows three examples of different error annotations for the translations.

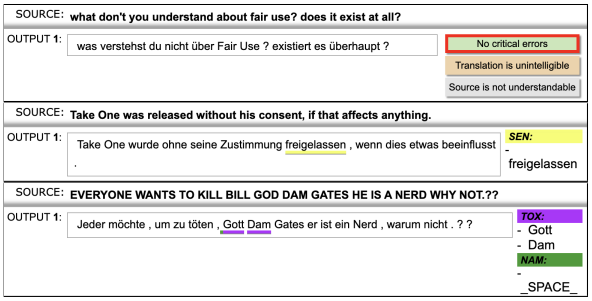


Figure 1: Example of fine-grained sentence annotation. Spans in the same colour belong to the same catastrophic error type. In the first case, the translation contains no critical error; in the second case, the translation contains only one SEN error; in the last case, the translation contains two errors: one TOX and one NAM (the space is annotated to indicate a missing named entity).

For this the first edition of this task, we aggregated these labels in two ways: First, for each of the three annotated versions of a sentence, we extrapolated the word-level labels into a sentence-level label: if the sentence contained at least one critical error, it was annotated as critical. Second: we took the majority sentence-level label from the three annotators to create a single sentence-level label for each sentence, resulting in the following binary labels:[5]

- ERR: the translated sentence contains at least one (any) token or whitespace (for deletion errors) annotated with a critical error in any categories, according to at least 2 out of 3 annotators, or otherwise

- NOT: the sentence does not contain any token with a critical error.

[4] https://mt4cat.fbk.eu/software/mt-equal

[5] We removed from the dataset sentences that had been annotated by the majority as having an unintelligible source or a translation with too many errors.

Thus, the task does not expect the errors to be categorised or to have their spans identified in the sentence, but rather to have a binary prediction on a sentence basis. For example, the first sample in Figure 1 would have resulted in the sentence labelled as NOT by that annotator, while the last two samples would have resulted in the sentence labelled as ERR.

An initial set of 10K English samples for training, development and test data was created from Wikipedia comments, extracted from two sources: the Jigsaw Toxic Comment Classification Challenge[6] and the Wikipedia Comments Corpus.[7] Machine translations were generated by the ML50 fairseq multilingual translation model for the following languages:

- English-Czech,
- English-Japanese,
- English-Chinese, and
- English-German.

After filtering for unintelligible source sentences and translations with too many errors, the statistics for the resulting splits are presented in Table 1. As expected, critical errors are rare. Give the nature of this dataset (user generated content with high chances of toxicity, named entities, etc.), we observed a fairly large proportion of sentences with such errors. Nevertheless, the distribution of labels is skewed towards the NOT class. The proportion of instances with NOT labels in the training set (similar for dev and test sets) is as follows: 83% for En-Cs, 72% for En-De, 91% or En-Ja, and 84% for En-Zh.

3 Baseline systems

Sentence-level baseline systems: For Tasks 1 and 2, both word and sentence-level, we used a multilingual transformer-based Predictor-Estimator approach (Kim et al., 2017), which is described in detail in (Fomicheva et al., 2020). Both baselines are implemented in OpenKiwi (Kepler et al., 2019) and trained using the concatenated train portions of the data for training (combining all 7 language pairs) and the concatenated development portions of the data for validation/early-stopping. In all cases, the XLM-RoBERTa transformer was used for the encoding (predictor) part of the architecture, using `xlm-roberta-base` for all experiments. The XLM-RoBERTa encoder is initially trained on the concatenated train and development segments using ULM fine-tuning (Howard and Ruder, 2018) and then this fine-tuned encoder is used in the full predictor-estimator model which is fine-tuned separately for each task scores (DA or HTER).

Word-level baseline systems: For Task 2, we used the same architecture and encoder as above, but it was trained to predict jointly word-level OK/BAD tags and sentence-level HTER scores.

Catastrophic error baseline system: The baseline model for Task 3 follows the MonoTransQuest architecture proposed by Ranasinghe et al. (2020) for sentence-level classification. As input, the model takes a sequence of tokens including the `[CLS]` token, and the source and translated sentence tokens, separated by a `[SEP]` token. This string is fed into a transformer encoder and the output of the encoder is given to a classification head where cross-entropy is adopted as the loss function. We use the pre-trained XLM-RoBERTa-base released by HuggingFace's model repository (**?**) for the implementation.

4 Participants

Table 2 lists all participating teams submitting systems to any of the tasks, and Table 3 report the number of successful submissions to each of the sub-tasks and language pairs. 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).

Bergamot (T1): Bergamot explores the use of a teacher-student knowledge distillation framework to transfer knowledge from a strong QE teacher model to a much smaller student model with a different, shallower architecture. Namely, the system distill a large and powerful QE model TransQuest [1] based on XLM-Roberta into a small BiRNN-based DeepQuest model [2]. The predictions from a teacher QE model trained on MLQE data [3] is used to train the lightweight student. Additionally, the system employs data augmentation through teacher predictions on monolingual data sampled from Wikipedia following

[6] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

[7] https://meta.wikimedia.org/wiki/Research:Detox/Data_Release#Wikipedia_Comments_Corpus

ID	Participating team	
Bergamot	University of Sheffield & Imperial College London & University of Wolverhampton, UK	(Gajbhiye et al., 2021)
Bergamot-UTartu	University of Tartu, Estonia	(Yankovskaya and Fishel, 2021)
ENSBRT	University of Illinois at Chicago & IQVIA, USA	(Chowdhury et al., 2021)
HW-TSC	Huawei Translation Services Center & Nanjing University, China	(Chen et al., 2021)
IST-Unbabel	Instituto de Telecomunicações Lisbon & Instituto Superior Técnico Lisbon & Unbabel, Portugal	(Zerva et al., 2021)
JHU-Microsoft	Johns Hopkins University & Microsoft	(Ding et al., 2021)
LAMA-ICL	LAMA - Imperial College London, UK	(Jiang et al., 2021)
NICT Kyoto	National Institute of ICT, Japan	(Rubino et al., 2021)
Papago	Naver, Republic of Korea	(Lim et al., 2021)
POSTECH	Pohang University of Science and Technology, Republic of Korea	(Heo et al., 2021)
QEMind	Alibaba, China	(Wang et al., 2021)
RTM	Boğaziçi University, Turkey	(Biçici, 2021)
SMOB-ECEIIT	Technion - Israel Institute of Technology, Israel	–
TUDa	Technische Universität Darmstadt, Germany	(Geigle et al., 2021)

Table 2: Participants to the WMT21 Quality Estimation shared task.

Task/LP	# submission
Task 1 – Sent-level Direct Assessment	**725**
Multilingual	32
English-German	99
English-Chinese	78
Romanian-English	58
Estonian-English	56
Nepalese-English	52
Sinhala-English	65
Russian-English	54
English-Czech	54
English-Japanese	62
Pashto-English	50
Khmer-English	65
Task 2 – Sent-/Word-level PE Effort	**163/178**
Multilingual	7/–
English-German	37/33
English-Chinese	22/32
Romanian-English	13/14
Estonian-English	13/19
Nepalese-English	6/11
Sinhala-English	7/10
Russian-English	11/11
English-Czech	10/14
English-Japanese	13/14
Pashto-English	6/8
Khmer-English	18/12
Task 3 – Sent-Level Critical Error Det.	**197**
English-German	56
English-Chinese	30
English-Czech	36
English-Japanese	75
Total	**1263**

Table 3: Number of submissions to each sub-task and language-pair at the WMT21 Quality Estimation shared task.

the procedure described in [3]. Further details about the distillation framework used for submission can be found in [4].

Bergamot-UTartu (T1, T2): Bergamot-UTartu proposes CNN-models based on attention weights extracted from NMT systems. For Task 1, they explored three QE models: i) CNN-DA trained on human-labelled data; ii) CNN-BLEURT a "zero-shot" system that requires only synthetic data, for which they used BLEURT scores (Sellam et al., 2020) as training data; iii) CNN-BLEURT+ a fine-tuned version of CNN-BLEURT. For Task 2, CNN-HTER is a model similar to CNN-DA, but trained on the post-editing scores.

ENSBRT (T2): ENSBRT propose a system that is an ensemble of multilingual BERT (mBERT)-based regression models, which are generated by fine-tuning on different input settings. They adapted their system for the zero-shot setting by exploiting target language-relevant language pairs and pseudo-reference translations.

HW-TSC (T1, T2, T3): HW-TSC's submissions in the three sub tasks follow the framework of Predictor-Estimator (Kim et al., 2017), with a pre-trained XLM-Roberta as Predictor and task-specific classifier or regressor as Estimator. They further explore to incorporate additional high-quality translation sentences in the way of multitask learning or encoding it with the Predictor directly. For Task1, they enable the model to jointly learn to score translations with a regression task and to distinguish between translations and additional better translations (i.e. post-edits from Task2 dataset) with a classification task. They also exploit

a data augmentation strategy based on MC dropout to improve zero-shot performance. They ensemble multi results with MC dropout to keep a relatively small number of parameters and model size. For Task 2, they leverage additional translation sentence generated by a NMT system trained for WMT21 News shared task in the way of directly concatenating it with source and original translation. A unified model is trained under multi-task learning framework where losses of source word, translation token and gap, additional translation token and gap, HTER scores are all added up to train the model. For Task 3, they translate source sentences with Google and Baidu translation API. Each new translation is then concatenated to the corresponding source and translation pair, to get a sentence feature. They ensemble the results of three different models and take their majority voting as final result.

IST-Unbabel (T1, T2): For Task 1, IST-Unbabel's system is an ensemble of an XLM-R with stacked adapter layers and an mBART that incorporates different types of uncertainty (annotation uncertainty and MT uncertainty). For Task 2, the submitted system is an ensemble of two XLM-R with adapters (the difference being the XLM-R checkpoint, while one uses the xlm-roberta-large normal checkpoint the other uses an XLM-R checkpoint pertained on data from the metrics shared task). The ensembled checkpoints learn to predict both word level tags and sentence level HTER scores in a multi-task setting.

JHU-Microsoft (T2): The JHU-Microsoft submission focuses on the word-level subtask of Task 2, for which they adopt Levenshtein Transformer (Gu et al. 2019) as their model architecture. The training procedure starts with training a non-autoregressive translation model using a Levenshtein Transformer, with its encoder and decoder initialized with those from the M2M multilingual translation model (Fan et al. 2020). They then fine-tune the model to perform the word-level QE task on the human-annotated training set, or optionally also on automatically generated pseudo-post-editing translation triplets. The final submission is an ensemble of 4-8 best models on the 2020 test set for each language pair, and the ensemble is performed by linear interpolation of scores from each model, with the interpolation weights tuned by the Nelder-Meade method (Nelder and Meade, 1965).

LAMA-ICL (T3): LAMA-ICL's approach builds on cross-lingual pre-trained representations in a sequence classification model. We further improve the base classifier by (i) adding a weighted sampler to deal with unbalanced data and (ii) introducing feature engineering, where features related to toxicity, named-entities and sentiment, which are potentially indicative of critical errors, are extracted using existing tools and integrated to the model in different ways. We train models with one type of feature at a time and ensemble those models that improve over the base classifier on the development (dev) set.

NICT Kyoto (T3): NICT Kyoto submission for the Critical Error Detection task consists in large scale QE pretraining with synthetic data in a multilingual and multimetric setting. A total of six sentence- and word-level quality indicators were involved in continued training of an XLM-R checkpoint using QE oriented training objectives in a multi-task fashion, based on a corpus of 70 million sentence pairs including twelve languages. Fine-tuning on the official dataset was then performed and resulting models from different initializations were ensembled to constitute the final submission.

Papago (T1): Papago's submission is a multilingual Quality Estimation system that explores the combination of pre-trained language models and multi-task Learning architectures. They propose an iterative training pipeline based on pretraining with large amounts of in-domain synthetic data and fine-tuning with gold (labeled) data. They then compress our system via knowledge distillation in order to reduce parameters yet maintain strong performance.

POSTECH (T2): POSTECH's model uses two pre-trained monolingual encoders to first produce monolingual representations of the two input data separately and then exchanges the information of these representations through

two additional cross attention networks. The two pre-trained monolingual encoders are an English ELECTRA and a German ELECTRA, respectively. They fine-tuned their system in two stages: the QE pre-training stage and the QE fine-tuning stage. In the former, they used a large quantity of artificial training data, and the loss value equals to the sum of the losses for the estimated HTER (sentence-level QE), OK-BAD for the source tokens, OK-BAD for the MT tokens, and OK-BAD for the gap tokens in between two MT tokens. In the latter, they only used human labelled training data, and the loss value is one of the four above mentioned loss values, depending on the targeted subtask.

QEMind (T1, T3): QEMind propose novel glass-box QE features to estimate the uncertainty of machine translations and incorporate them into the transfer learning from the large-scale pre-trained model, XLM-Roberta. In addition, three important strategies are particularly utilized for improving the QE system's performance such as multilingual training, data augmentation and model ensemble.

RTM (T1, T2): Referential Translation Machines (RTMs) Superlearner results combine individual machine learning model results via cross-validation on the training set. The combined models guarantee lower error on the validation set than the model that minimises the overall error. A superlearner model improves the results over non-mixture results.

SMOB-ECEIIT (T1): SMOB-ECEIIT's participation is fully unsupervised, as created without using any annotated data or even parallel bilingual data. The system is composed of two novel different methods. The first method is based on soft alignment of multilingual contextual embeddings, generated by pre-trained mBert or XLM-R (depending on the specific language). The soft alignment is calculated by the Sinkhorn distance (Cuturi, 2013), which is an optimal transportation distance with an entropic regularization term. The second method is based on the assumption that word embedding spaces are approximately isometric (Vulić et al., 2020), and on an isometric-invariant method known as Persistent Homology (Edelsbrunner, 2013). Each sentence is represented by the distances between its own word embeddings (either static or contextual). The created distance matrices are compared using the Wasserstein distance between their persistence barcodes (the output of persistent homology computation). Finally, the two methods are linearly combined.

TUDa (T1): TUDa's submissions are produced with pre-trained multilingual language models which they extended to new languages and unseen scripts using recent adapter-based methods.

5 Results

5.1 Task 1

Submissions for Task 1 are **evaluated** against the true z-normalised direct assessment label using Pearson's r correlation score as primary metric. This is what was used for ranking system submissions. Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) were also computed as secondary metrics. Statistical significance on Pearson r was computed using William's test.[8]

Table 4 summarises the results for all language pairs, as well as the multilingual variant, in terms of Pearson's r correlation with direct assessments, ranking systems by their average performance for all language pairs (using 0 as Pearson score for other languages). In the Appendix, Tables 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 and 20 provide the detailed results for all language pairs and the multilingual variant, ranking participants by their performance for each of these cases.

Best performers The best system varies slightly across language pairs, with QEMind winning the multilingual task, i.e. the average performance for all language pairs (including zero-shot). Overall, the three top performing systems, QEMind, HW-TSC and IST-Unbabel, perform very closely on average, and also for each given language. The three make use of the XML-R large pre-trained representations in a predictor-estimator fashion, and model ensembling. Another recurring theme is to explore data augmentation (QEMind and HW-TSC) and model uncertainty (QEMind and IST-Unbabel). While the baseline system also uses XLM-R as predictor, it uses its 'base' version, and only the provided 'train' part of the data to train the estimator. In addition, it does not resort to ensembling.

[8] https://github.com/ygraham/mt-qe-eval

Model	Multi	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	En-Cs	En-Ja	Ps-En	Km-En
QEMind	0.675	0.567	**0.603**	**0.908**	**0.812**	**0.867**	**0.596**	**0.806**	**0.582**	**0.359**	**0.647**	**0.679**
HW-TSC	0.665	**0.584**	0.583	0.901	**0.808**	0.858	0.581	0.878	**0.573**	**0.364**	0.622	0.659
IST-Unbabel	0.665	**0.579**	0.586	0.899	0.796	0.856	**0.605**	0.792	**0.577**	**0.355**	0.628	0.650
papago (IKT)	0.658	**0.568**	0.567	0.901	0.759	0.853	**0.595**	0.793	**0.572**	0.332	**0.637**	0.662
TUDa	0.631	0.473	0.558	0.886	0.792	0.834	0.571	0.764	0.545	0.330	0.609	0.639
Inmon‡	0.623	–	–	–	–	–	–	–	0.547	0.297	0.592	0.630
papago (KD)	0.613	0.551	0.553	0.879	0.794	0.823	0.582	0.744	0.497	0.276	0.582	0.625
BASELINE	0.541	0.403	0.525	0.818	0.660	0.738	0.513	0.677	0.352	0.230	0.476	0.562
SMOB-ECEIIT	0.348	0.226	0.131	0.650	0.329	0.544	0.347	0.420	0.195	0.153	0.424	0.409
Bergamot	–	–	0.687	0.544	0.626	0.425	–	–	–	–	–	–
Bergamot-UTartu	–	0.369	–	–	0.547	–	–	–	0.300	–	–	–
RTM	–	0.143	0.248	0.287	0.099	0.127	0.061	0.356	0.104	0.082	–	–

Table 4: Pearson correlation with direct assessments for the submissions to WMT21 Quality Estimation Task 1. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; ‡ indicates Codalab username of participants from whom we have not received further information.

To gain a better understanding in the performance of different QE approaches for different language pairs, Figure 2 shows the scatter plots for the baseline and the best performing system for each language pair.

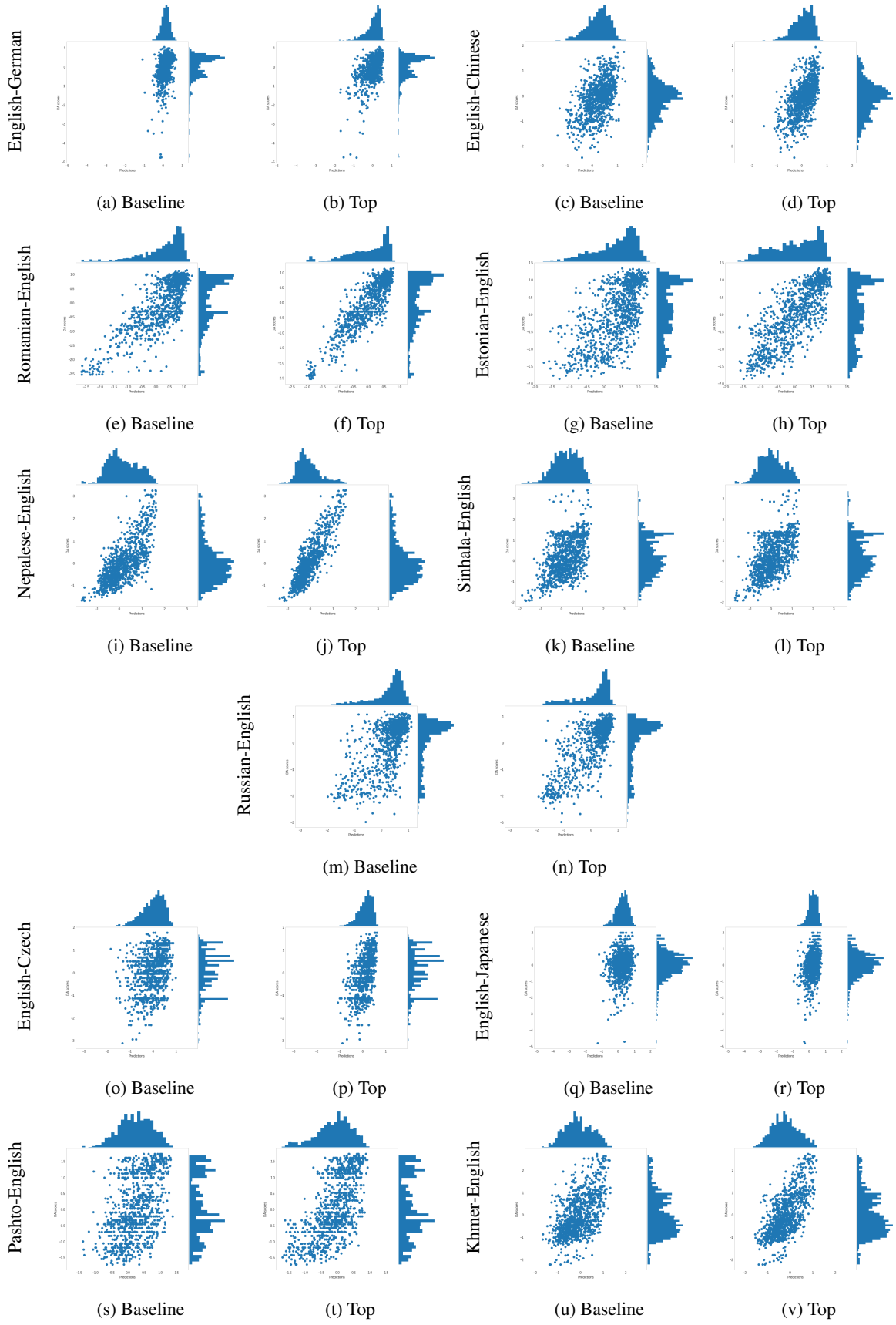
The performance of all except four systems is substantially better than that of the baseline system for all languages. The systems below the baseline correspond to unsupervised systems (Bergamot, Bergamot-UTartu - using NMT glass-box features; and SMOB-ECEIIT, using alignment over XLM-R representations), as well as RTM, which does not rely on pre-trained representations altogether.

Zero-shot languages On the zero-shot languages, the performance was comparable to those of the average non-zero-shot language pairs, except for the En-Ja language pair, where it was substantially lower. Most systems achieved such good performance by relying on multilingual prediction models trained on cross-lingual representations from XLM-R. With En-Ja, we believe there may have been an issue with the segmentation of the Japanese data after translation, which led to annotation issues and/or issues of mapping of characters against the vocabulary of pre-trained language models. We will investigate this further.

High vs low-resource performance Similar to last year, MT quality for the high-resource language pairs, in particular En-De but also En-Zh, proved to be more challenging to predict. This could be an indication of less variability in the MT outputs for these language pairs, given that the NMT models are likely to perform overall well for these languages. This would lead to little variability in perceived MT quality by humans, and thus a harder data to learn from. Interestingly, this also seemed to be the case in the zero-shot QE setting for En-Cs, which is relatively higher resource than Ps-En and Km-En. We observe that these differences in correlation also happen with the HTER predictions for these language pairs (see the analysis of the sentence level task in §5.2 and Table 5).

All medium and low and medium-resource language pairs achieve high correlations, in particular in the supervised settings with Ro-En and Ne-En. This again is an indication of the potential of multilingual or cross-lingual pre-trained representations. It could also indicate that the models (and human annotators to some extent) rely heavily on the target language (English), which is well represented in the pre-trained representations.

High correlations Just like in WMT2020, the very high correlation for some language pairs, particularly for Ro-En (r = 0.91) but also for Ne-En (r = 0.87) could be explained by the fact that there is a number of very low-quality sentences that the QE systems are able to successfully detect. Esp. for Ro-En, we find that they correspond to 'hallucinated' outputs that do not have anything to do with the original sentences. Detecting such cases should be trivial for QE systems, which explains the particularly high correlation values for this language pair.



(a) Baseline (b) Top (c) Baseline (d) Top

(e) Baseline (f) Top (g) Baseline (h) Top

(i) Baseline (j) Top (k) Baseline (l) Top

(m) Baseline (n) Top

(o) Baseline (p) Top (q) Baseline (r) Top

(s) Baseline (t) Top (u) Baseline (v) Top

Figure 2: Scatter plots for the predictions against true direct assessment scores for the baseline and top-performing system for each language pair. The histograms show the corresponding marginal distributions of predicted and true scores.

5.2 Task 2

Sentence-level post-editing effort For this task variant, **evaluation** was performed against the true HTER label using the same metrics as in Task 1, with Pearson's r correlation score as the primary metric. Table 5 summarises the results for all language pairs, including the multilingual performance. Systems are ranked by their averaged performance over all language pairs based on the Pearson r coefficient. Statistical significance on Pearson r was computed using the William's test. In the Appendix, Tables 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 and 32 provide the detailed results for all language pairs and the multilingual variant, ranking participants by their performance for each of these cases. Note that for the *multilingual* track (Table 21 and 1st column of table 5) we present only the performance of systems that submitted multilingual models (HW-TSC and IST-Unbabel) for that specific track.

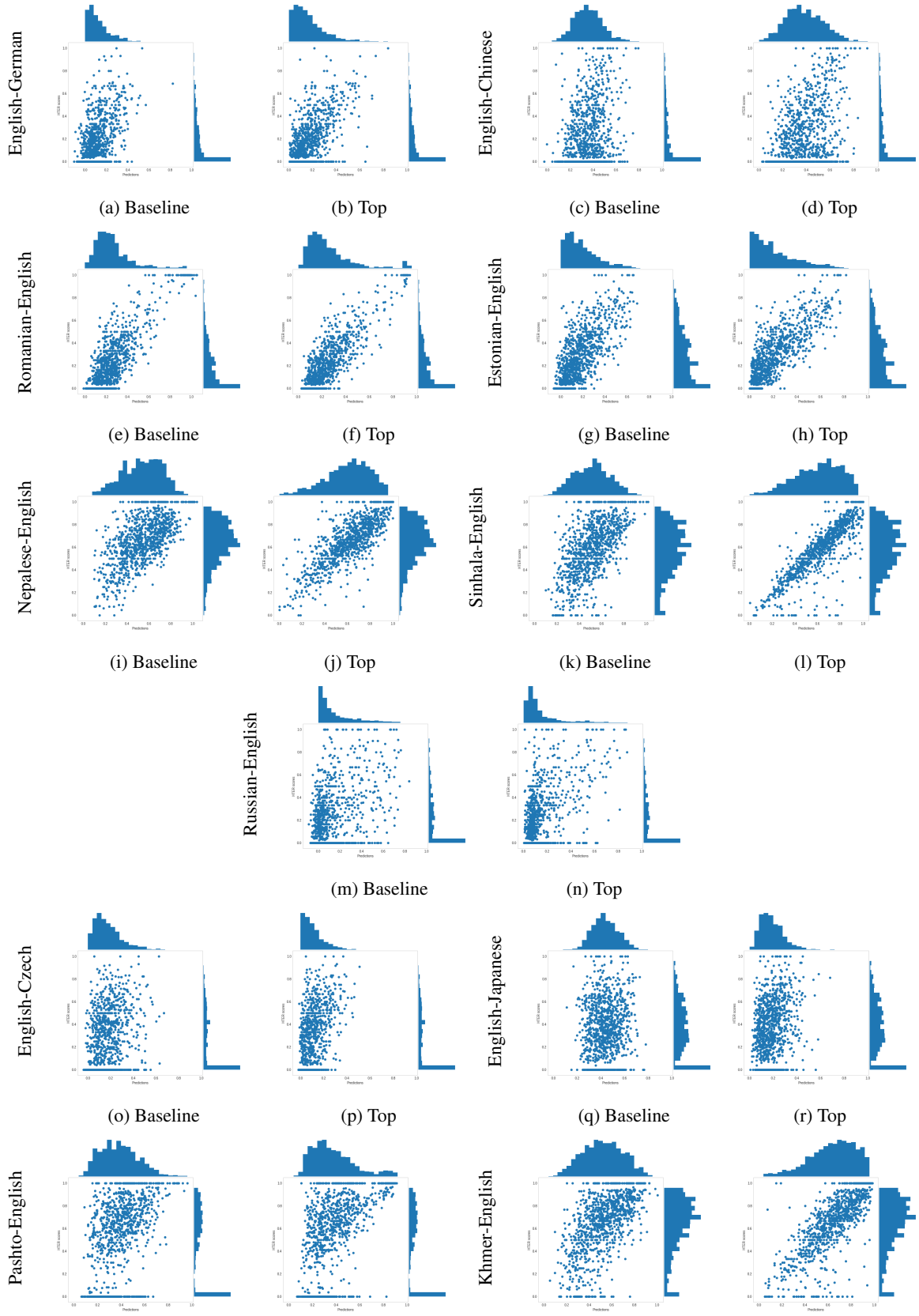
Best performers Both multilingual system submissions outperform the monolingual approaches in the individual language pairs, with HW-TSC ranking first for the majority of the 'supervised' language pairs (En-De, En-Zh, Ne-En, Si-En and Ru-En) as well as the multilingual track and IST-Unbabel leading the majority of the zero-shot language pairs (En-Cs, En-Ja, Ps-En). Apart from the multilingual aspect, the two top systems have more in common: they both use the Predictor-Estimator framework with XLM-Roberta encoders for the predictor and task-specific classifiers for the estimator. Additionally, they both address the sentence- and word-level task using a multi-task approach. HW-TSC enhances their approach using additional pseudo-references as input (generated by another NMT system), while IST-Unbabel system uses additional external data from the WMT Metrics shared task and incorporate adapters in their architecture.

Overall, submitted systems used a variety of approaches to improve performance and address the zero-shot tasks, which revolved around augmenting the training data either by including synthetic data (Bergamot-UTartu, POSTECH) and/or external data (IST-Unbabel) or by using pseudo-references generated by other MT systems (ENSBRT, HW-TSC, JHU-Microsoft). Additionally, ensembling approaches were used to boost performance (HW-TSC, ENSBRT, IST-Unbabel).

Figure 3 shows the scatter plots for the baseline and the best performing system for each language pair. We can see that in most language pairs (perhaps with the exception of En-Zh) the scatter plots for the 'Top' system are much narrower and closer to the identity line, compared to these for the corresponding baseline. More importantly, language pairs with a high proportion of HTER score values close to 0 (many segments without post edits) prove to be more challenging for the submitted models. For example, comparing En-Zh, Ru-En against En-De to Si-En, Ne-En and Et-En, we can see that the latter have narrower, better correlated scatter plots in Figure 3, which is reflected in higher performance in Table 5. This observation seemingly extends to the zero-shot languages, where we observe that performance for the Km-En language pair is consistently higher for all systems compared to the other zero-shot pairs.

Word-level errors For this task, the primary **evaluation** metric is Matthews correlation coefficient (MCC, Matthews, 1975). We also report the F_1-scores for the OK and BAD classes. Similarly to the previous editions, we evaluate separately the source and target side, and this year we also provide a separate evaluation for the target gap tag predictions. We also calculate the performance for combined gaps and words in MT, although it was not considered in the overall ranking process. Systems are primarily ranked by their MCC performance for the word tags on the target side (denoted as 'Words in MT' in the tables). The word-level results for Task 2 are summarised in Tables 6, ordered by the MCC metric, while in the Appendix, Tables 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43 and 44 provide the detailed results for all language pairs and the multilingual variant, ranking participants by their performance for each of these cases. Statistical significance was calculated based on the MCC metric for each language pair using randomization tests with Bonferroni correction (Yeh, 2000a).

Best performers For the multilingual track the picture is similar to the sentence level sub-task, with the HW-TSC system ranking first across all performance indicators, and also leading most of the individual language pairs. Apart from the two multilingual approaches, most of the systems participating in the sentence level sub-task did not submit predictions for the word-level task with the exception of POSTECH which submitted predictions for En-De. However, the JHU-Microsoft which



(a) Baseline (b) Top (c) Baseline (d) Top

(e) Baseline (f) Top (g) Baseline (h) Top

(i) Baseline (j) Top (k) Baseline (l) Top

(m) Baseline (n) Top

(o) Baseline (p) Top (q) Baseline (r) Top

(s) Baseline (t) Top (u) Baseline (v) Top

Figure 3: Scatter plots for the predictions against true HTER scores for the baseline and top-performing system for each language pair. The histograms show the corresponding marginal distributions of predicted and true scores.

Model	Multi	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	En-Cs	En-Ja	Ps-En	Km-En
HW-TSC	0.631	**0.653**	**0.368**	0.862	**0.809**	**0.798**	**0.869**	**0.562**	0.475	**0.262**	**0.534**	**0.753**
IST-Unbabel	0.597	0.617	0.290	**0.879**	**0.811**	0.718	0.710	**0.539**	**0.529**	**0.275**	**0.555**	0.655
BASELINE	0.502	0.529	0.282	0.831	0.714	0.626	0.607	0.448	0.306	0.098	0.503	0.576
ENSBRT	–	0.520	–	0.795	0.666	0.572	0.522	0.376	–	–	–	0.530
Abulice‡	–	0.577	0.312	–	–	–	–	–	–	–	–	–
POSTECH	–	0.546	–	–	–	–	–	–	–	–	–	–
Bergamot-UTartu	–	0.531	–	–	0.562	–	–	–	–	–	–	–
RTM	–	–	0.087	–	–	–	–	–	–	–	–	–

Table 5: Pearson correlation with post-editing effort for the submissions to WMT21 Quality Estimation Task 2 (sentence-level). For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; ‡ indicates Codalab username of participants from whom we have not received further information.

Model	Multi	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	En-Cs	En-Ja	Ps-En	Km-En
Words in MT												
HW-TSC	0.530	**0.510**	**0.354**	**0.666**	**0.606**	**0.674**	**0.847**	**0.451**	**0.380**	**0.258**	**0.450**	**0.636**
IST-Unbabel	0.430	0.466	0.310	**0.649**	0.570	0.508	0.528	0.332	**0.376**	0.169	0.370	0.448
BASELINE	0.346	0.370	0.247	0.536	0.461	0.440	0.425	0.256	0.273	0.131	0.313	0.351
JHU-Microsoft	–	**0.523**	0.149	0.634	0.572	0.329	–	0.303	–	–	0.191	–
Abulice‡	–	0.437	0.033	–	–	–	–	–	–	–	–	–
POSTECH	–	0.413	–	–	–	–	–	–	–	–	–	–
GAPs in MT												
HW-TSC	0.337	**0.300**	**0.172**	**0.446**	**0.312**	**0.403**	**0.639**	**0.388**	**0.213**	**0.152**	**0.260**	**0.419**
IST-Unbabel	0.196	0.183	0.068	0.357	0.254	0.268	0.258	0.165	0.125	0.025	0.177	0.259
BASELINE	0.126	0.116	0.065	0.205	0.136	0.215	0.208	0.073	0.039	0.036	0.134	0.175
JHU-Microsoft	–	**0.256**	0.035	0.208	0.218	0.207	–	0.167	–	–	0.118	–
Abulice‡	–	–	–	–	–	–	–	–	–	–	–	–
POSTECH	–	0.110	–	–	–	–	–	–	–	–	–	–
Words in SRC												
HW-TSC	0.432	**0.450**	**0.310**	**0.614**	**0.549**	**0.545**	**0.616**	**0.426**	**0.313**	**0.217**	**0.304**	**0.410**
IST-Unbabel	0.378	0.404	0.286	**0.603**	0.522	**0.445**	0.406	0.351	**0.294**	**0.210**	**0.294**	**0.345**
BASELINE	0.307	0.322	0.241	0.511	0.405	0.390	0.335	0.215	0.224	0.175	0.249	0.279
JHU-Microsoft	–	–	–	–	–	–	–	–	–	–	–	–
Abulice‡	–	0.392	0.011	–	–	–	–	–	–	–	–	–
POSTECH	–	0.320	–	–	–	–	–	–	–	–	–	–
Combined Words and Gaps in MT												
HW-TSC	n/a	**0.496**	**0.359**	**0.656**	**0.584**	**0.749**	**0.868**	**0.456**	0.336	0.180	**0.533**	**0.677**
IST-Unbabel	n/a	**0.468**	**0.369**	**0.640**	**0.582**	0.705	0.690	0.339	**0.400**	0.217	**0.538**	0.631
BASELINE	n/a	0.378	0.320	0.543	0.482	0.672	0.642	0.319	0.339	0.250	0.517	0.587
JHU-Microsoft	n/a	**0.500**	0.240	0.642	**0.572**	0.657	–	0.329	–	–	**0.523**	–
Abulice‡	n/a	0.442	0.118	–	–	–	–	–	–	–	–	–
POSTECH	n/a	0.403	–	–	–	–	–	–	–	–	–	–

Table 6: Matthews correlation coefficient with the OK and BAD classes labels for the submissions to WMT21 Quality Estimation Task 2 (word-level). For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000a). Baseline systems are highlighted in grey; ‡ indicates Codalab username of participants from whom we have not received further information.

Model	En-De	En-Zh	En-Cs	En-Ja
NICT Kyoto	**0.546**	**0.311**	**0.511**	0.252
HW-TSC	0.490	**0.353**	**0.448**	0.318
LAMA-ICL	0.498	0.305	0.473	0.314
QEMind	0.480	0.278	0.454	0.260
BASELINE	0.397	0.187	0.388	0.214
silence1024‡	0.449	**0.343**	–	0.277
Jason_pogba‡	–	–	–	0.278
serkan‡	–	0.141	–	–

Table 7: Matthews correlation coefficient with the binary critical error labels for the submissions to WMT21 Quality Estimation Task 3. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system based on William's test. Baseline systems are highlighted in grey; ‡ indicates Codalab username of participants from whom we have not received further information.

was trained only for the target predictions of the word-level task, obtained the best performance in En-De. JHU-Microsoft also seemed to obtain competitive performance for the Et-En and Ro-En tasks, indicating a strength in languages closer to English.

Overall, the performance of the top two systems is closer for the high – and some of the medium – resource languages, both in the supervised and zero-shot tracks (En-De, En-Zh, En-Cs). Much like the WMT20 shared task, the performance for the target word tags is considerably higher compared to the source tags. This phenomenon is observed across language pairs with the exception of Ru-En where predictions for source and target words are close for all systems. This year we can also observe the performance on target gaps separately, which is consistently lower, even when compared to the source tags, across all language pairs and submitted systems.

It is important to note that when focusing on the combined target performance, i.e., the combination of word and gap quality predictions for the MT, the order and performance differences between the top scoring teams can vary compared to the MT word prediction ones. Overall, there are fewer language pairs where we have a clear winner (Ne-En, Si-En and Km-En for HW-TSC and Cs-En for IST-Unbabel) while for the rest there is no statistically significant difference between the top pairs. Still, HW-TSC is consistently among the top systems, with the exception of Cs-En.

5.3 Task 3

Table 7 summarises the results for all language pairs, ranked by their performance in terms of Matthews correlation coefficient (MCC, Matthews, 1975). In the Appendix, Tables 45, 46, 47 and 48 provide the detailed results for all language pairs, ranking participants by their performance for each of these cases. Statistical significance is calculated using the William's test.

This task attracted fewer participants than the others, most likely because it is new. All described systems perform better than the baseline for all language pairs. Across languages, the order of MCC scores roughly corresponds to the skewness of data distribution obtained for languages: For En-De, which achieved the highest MCC score, the NOT (no error class) accounted for 72% of the training instances, while for En-Ja, with the lowest MCC score, the NOT class accounted for 91% of the training instances.

Best performers NICT Kyoto ranked among the top systems for all language pairs. However, only for En-De it did significantly outperform all other systems. For the rest of the language pairs there could not be a clear winner based on statistical significance testing; HW-TSC was in the top-ranked systems for En-Zh and En-Cs, while for En-Ja no system managed to significantly outperform the others, but they all performed significantly better than the baseline.

In terms of the approaches applied by the best performing systems, they all use the baseline architecture as starting point, but HW-TSC also uses machine translations for the source sentences by top online systems. These are concatenated to the provided source and translation pair. NICT Kyoto also added synthetic data for multiple language pairs with multitask learning and model ensembling. LAMA-ICL used additional features to detect the presence/deviation of toxicity, sentiment and named entities, also followed by ensenbling of models with different individual features.

6 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.

General progress. Participating systems achieved very promising results for most languages, with the best performing submissions showing moderate to strong correlation for sentence-level DA and HTER prediction tasks. One reason for high correlation levels is likely

to be that top performing systems are based on pre-trained representations. Even for zero-shot languages (see below), relatively high (above 0.5) correlation was achieved for most languages for sentence-level tasks. The same applies for the word-level tasks, where the performance was behind that of supervised prediction, but still high for Km-En and Ps-En.

A comparison to previous years submissions is possible for Task 1 on the non-zero shot languages. The training data is the same and the test sets (test20 and test21) were created at the same time, with data sampled and annotated in the same way. Comparing Pearson correlation scores from the 2020 official results to this year's official results, as we can see from Table 8, for the languages which had already achieved very strong correlations, it remained the same (Ro-En, Et-En, Ru-En) or improved (Ne-En), whereas for the languages with average correlation, it mostly improved substantially (En-De, En-Zh). The exception was Si-En, where the correlation was lower in 2021, which needs further investigation. Overall, we believe the numbers show steady progress over previous models, even though the core of most of the winning systems is the same for the 2020 and 2021 editions.

Model size. When interpreting the results for all tasks, it should be noted that most of the participants use extremely resource-heavy systems, ensembles of multiple models with more than 500M parameters, which could make them difficult to use in practice. In this year's edition of the Shared Task on QE we asked the participants to provide information on the size of their models. Figures 4 and 5 illustrate the performance-efficiency trade-off for the submitted systems. On the x-axis we plot the Pearson correlation with sentence-level DA judgements (Task 1), while the y-axis shows the number of model parameters, as reported by the participants. Pareto-optimal submissions are marked in blue. These plots give us a different view of the performance of the submitted systems. Thus, for the higher quality models, the best results are achieved by QEMind and HW-TSC, whereas Bergamot, Bergamot-UTartu and BASELINE are optimal in terms of model size.

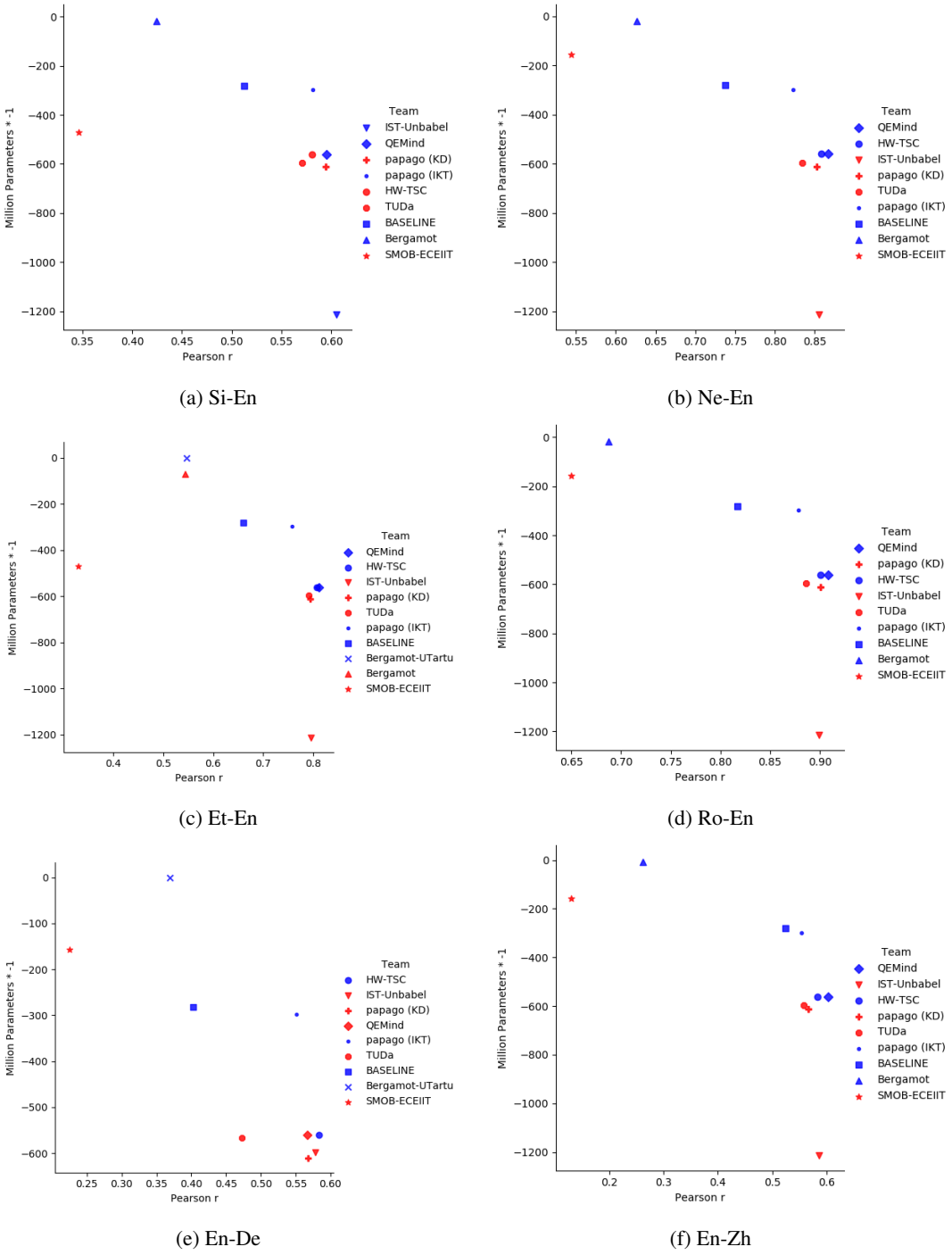
Extending publicly available benchmarks. This year counted with substantial new data. One the one hand, we extended the MLQE-PE dataset with more DA test sets (for all seven previous language pairs and four new zero-shot language pairs), as well as post-editing training and test sets for five additional language pairs (which only had DA scores before), as well as the four zero-shot language pairs. On the other hand, we created sizeable data for the new Task 3, a unique set focusing on critical errors, based on three annotations by professional translators. We hope that others will also contribute by adding new languages to this dataset in the future.

Zero-shot prediction. For the first time, we introduced language pairs for which no training data was available. This challenge was addressed mainly in two ways: synthetic data creation with using parallel data for the relevant languages, and use of indicators coming from the NMT system for unsupervised prediction. Overall, the performance for these languages was surprisingly good (except for En-Ja, potentially for data segmentation issues), comparable to non-zero-shot languages in the dataset. We attribute this high performance mostly to the use of fine-tuning on synthetic data for the relevant languages. In future editions, we may consider blinder zero-shot settings where participants will not be informed of the actual languages the models will require to predict the quality for, to encourage the development of truly multilingual or language-agnostic models.

Critical error detection. We posit that the detection of critical errors is a very important problem for two main reasons: (i) high-quality NMT models may produce fluent translations that may appear very good, but contain localised errors which are not always obvious and may go unnoticed, even by human translators post-editing the translation; and (ii) certain types of content are particularly challenging for MT models, such as social media data posts containing named entities, and could lead to critical errors especially if translations are to be used without human editing. While in the past we have provided word-level QE tasks where errors were annotated not only with error categories, but also error severity (e.g. MQM data in last year's WMT QE Task 3), this was the first attempt to predict specifically (and only) critical errors. This seems a much harder problem, as we expect the QE model to be able not only to find errors, but to distinguish minor (and even major) errors from critical errors. That was the reasoning for our "simplification" of the task this year, i.e. for making

Shared task	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En
WMT 2021	**0.58**	**0.60**	0.91	0.81	**0.87**	0.61	0.81
WMT 2020	0.55	0.54	0.91	0.82	0.82	**0.68**	0.81

Table 8: Pearson correlation with direct assessments - comparison between top submission in 2020 and 2021. While the test set is different, it was taken from the same distribution. The training set is the same.



(a) Si-En

(b) Ne-En

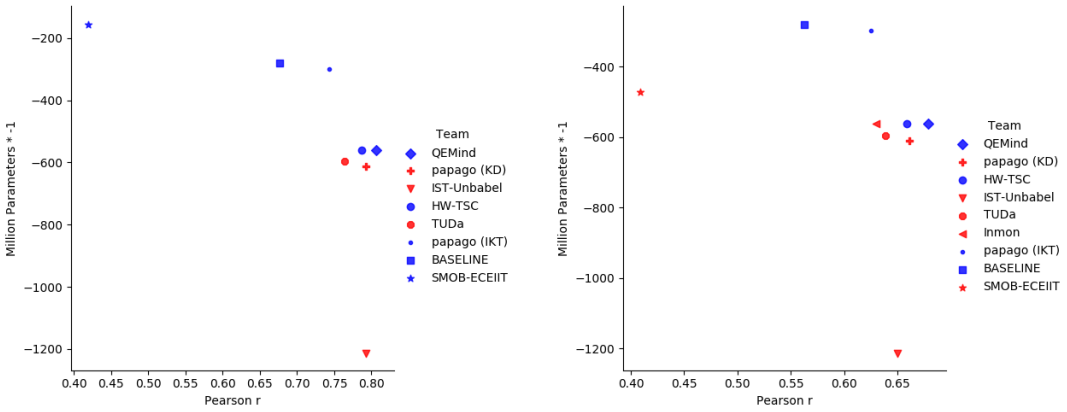
(c) Et-En

(d) Ro-En

(e) En-De

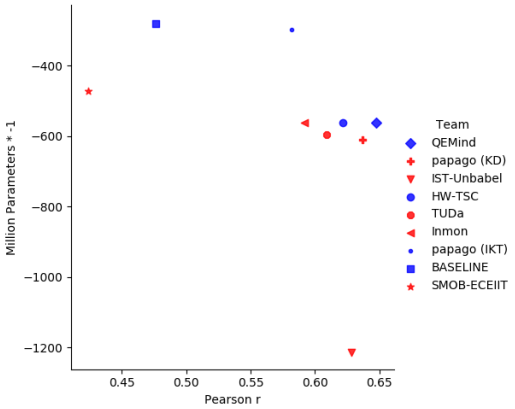
(f) En-Zh

Figure 4: Performance of the submitted systems on Task 1 for Si-En, Ne-En, Et-En, Ro-En, En-De and En-Zh. The x-axis shows Pearson correlation with human judgements and the y-axis corresponds to the number of model parameters multiplied by -1. Pareto optimal submissions are marked in blue, while the rest are shown in red.

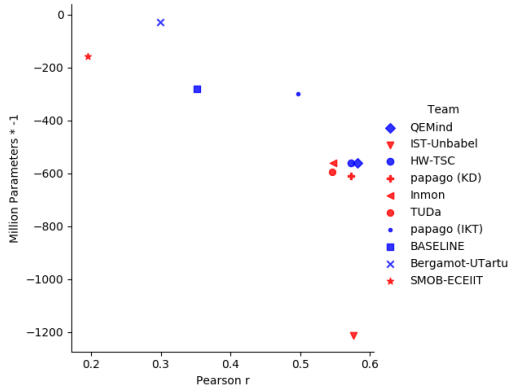


(a) Ru-En

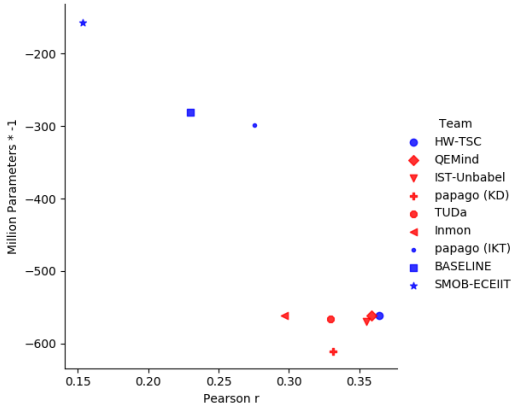
(b) Km-En



(c) Ps-En



(d) En-Cs



(e) En-Ja

Figure 5: Performance of the submitted systems on Task 1 for Ru-En, Km-En, Ps-En, En-Cs and En-Ja. The x-axis shows Pearson correlation with human judgements and the y-axis corresponds to the number of model parameters multiplied by -1. Pareto optimal submissions are marked in blue, while the rest are shown in red.

it a sentence-level binary classification problem. This might be enough for filtering purposes, i.e. to avoid offering/using automatic translations that may contain critical errors. If the goal is to support human translators in the task of post-editing, more fine-grained prediction may be needed.

The overall results for this task in terms of MCC are promising, especially for En-Cs and En-De. Considering the detailed results for this task in Tables 45, 46, 47 and 48, we see that despite the skewed distribution between the two classes, the models achieve a high F1 score at detecting errors, around 0.9 or higher for all language pairs.

7 Conclusions

This year's edition of the QE Shared Task introduced a number of new elements: new data covering five more language pairs with post-edits for sentence and word-level prediction, new test sets for all tasks, including four new zero-shot language pairs, and a new task focusing on critical error detection. Our analysis also paid close attention to model size, an important aspect for deploying QE systems in realistic applications, such as real-time inference and devices with limited resources. The tasks attracted a steady number of participating teams and systems and we believe the overall results are a great reflection of the SotA in QE. Continuing from the effort we set forward last year, this edition the tasks in this edition, with its zero-shot variant, cover a broad range of challenges in QE, such as improving performance for languages with skewed distributions, addressing low (or zero) resource languages, predicting source words that lead to errors, multilingual models, etc.

We are making the gold labels and all submissions to all tasks available for those interested in further analysing the results, investigating approaches for prediction ensembling, among others.

Acknowledgments

Marina Fomicheva and Lucia Specia were supported by funding from the Bergamot project (EU H2020 Grant No. 825303). André Martins and Chrysoula Zerva were funded by the P2020 programs Unbabel4EU (contract 042671) and MAIA contract 045909), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020. We would like to thank Erick Fonseca for answering questions about data preprocessing, and Marina Sánchez-Torrón and Camila Pohlman for monitoring the post-editing process for English-German, English-Chinese, and Russian-English for Task 2. We thank IQT Labs for providing the Russian-English dataset for Task 1. We thank Genze Jiang for helping with the baseline model for Task 3.

References

Ergun Biçici. 2021. Rtm superlearner results at quality estimation task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, Yujia Liu, and Shujian Huang. 2021. Hw-tsc's participation at wmt 2021 quality estimation shared tasks. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Shaika Chowdhury, Naouel Baili, and Brian Vannah. 2021. Ensemble fine-tuned mbert for wmt21 translation quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transportation distances. *arXiv preprint arXiv:1306.0895*.

Shuoyang Ding, Marcin Junczys-Dowmunt, Matt Post, Christian Federmann, and Philipp Koehn. 2021. The jhu-microsoft submission for wmt21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Herbert Edelsbrunner. 2013. Persistent homology: theory and practice.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020. MLQE-PE: A multilingual quality estimation and post-editing dataset. *arXiv preprint arXiv:2010.04480*.

Amit Gajbhiye, Marina Fomicheva, Fernando Alva-Manchego, Frédéric Blain, Abiola Obamuyide, Nikolaos Aletras, and Lucia Specia. 2021. Knowledge distillation for quality estimation.

Gregor Geigle, Jonas Elias Stadtmüller, Wei Zhao, Jonas Pfeiffer, and Steffen Eger. 2021. Tuda at wmt21: Sentence-level direct assessment with adapters. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Dam Heo, WonKee Lee, Baikjin Jung, and Jong-Hyeok Lee. 2021. Quality estimation using dual encoders with transfer learning. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Genze Jiang, Zhenhao Li, and Lucia Specia. 2021. Icl's submission to the wmt21 critical error detection shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of ACL 2019 System Demonstrations*.

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*, pages 562–568.

Seunghyun Shaun Lim, Hantae Kim, and Hyunjoong Kim. 2021. Papago submission for the wmt21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Raphaël Rubino, Atsushi Fujita, and Benjamin Marie. 2021. Nict kyoto submission for the wmt'21 quality estimation task: Multimetric multilingual pretraining for critical error detection. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

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: Technical Papers*, pages 223–231.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Y. Tang, C. Tran, Xian Li, P. Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are all good word vector spaces isomorphic? *arXiv preprint arXiv:2004.04070*.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. Qemind: Alibaba's submission to the wmt21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Evan J. Williams. 1959. *Regression Analysis*, volume 14. Wiley, New York, USA.

Lisa Yankovskaya and Mark Fishel. 2021. Bergamot-utartu participation in the wmt'2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

Alexander Yeh. 2000a. More Accurate Tests for the Statistical Significance of Result Differences. In *Coling-2000: the 18th Conference on Computational Linguistics*, pages 947–953, Saarbrücken, Germany.

Alexander Yeh. 2000b. More accurate tests for the statistical significance of result differences. *arXiv preprint cs/0008005*.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro G. Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and

André F. T. Martins. 2021. Ist-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation Shared Tasks Papers*, Online. Association for Computational Linguistics.

A Official Results of the WMT21 Quality Estimation Task 1

Tables 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 and 20 show the results for all language pairs and the multilingual variant, ranking participating systems best to worst using Pearson's r correlation as primary key for each of these cases.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
QEMind	3	0.675	0.627	0.486	2,244,030,744	560,981,507
HW-TSC	2.4	0.665	0.627	0.482	2,243,941,083	560,941,057
IST-Unbabel	5	0.665	0.642	0.495	4,872,322,439	1,214,683,792
papago (IKT)	5.2	0.658	0.645	0.496	2,503,797,760	611,278,859
TUDa	6.2	0.631	0.688	0.526	2,382,759,964	595,689,991
Inmon‡	5.2	0.623	0.687	0.526	2,243,941,083	560,941,057
papago (KD)	4.2	0.613	0.687	0.524	1,249,902,592	297,974,795
BASELINE	5.2	0.541	0.729	0.562	1,142,413,043	281,291,535
SMOB-ECEIIT	6.6	0.348	1.057	0.821	1,886,937,088	471,716,864

Table 9: Official results of the WMT21 Quality Estimation Task 1 for the **Multilingual** variant. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	2.6	0.584	0.544	0.390	2,243,941,083	560,941,057
• IST-Unbabel	4.4	0.579	0.567	0.393	2,409,244,995	598,943,476
• papago (IKT)	5.8	0.568	0.580	0.430	2,445,115,000	611,278,859
QEMind	4.8	0.567	0.579	0.432	2,244,030,744	560,981,507
papago (KD)	4.2	0.551	0.587	0.426	1,249,902,592	297,974,795
TUDa	6.6	0.473	0.626	0.440	2,264,844,300	566,211,075
BASELINE	5.2	0.403	0.629	0.433	1,142,413,043	281,291,535
Bergamot-UTartu	5.2	0.369	0.854	0.605	6,985,478	421,537
SMOB-ECEIIT	6.2	0.226	1.070	0.834	626,401,280	156,589,824
RTM	n/a	0.143	1.150	0.538	61,203,283,968	380

Table 10: Official results of the WMT21 Quality Estimation Task 1 for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	3	0.603	0.580	0.450	2,244,030,744	560,981,507
IST-Unbabel	5.6	0.586	0.631	0.499	4,872,322,439	1,214,683,792
HW-TSC	3.6	0.583	0.627	0.487	2,243,941,083	560,941,057
papago (IKT)	5	0.567	0.623	0.490	2,503,797,760	611,278,859
TUDa	6.6	0.558	0.687	0.541	2,382,759,964	595,689,991
papago (KD)	4.8	0.553	0.643	0.500	1,249,902,592	297,974,795
BASELINE	5	0.525	0.683	0.534	1,142,413,043	281,291,535
Bergamot	5.4	0.262	1.088	0.914	28,949,742	6,941,751
RTM	n/a	0.248	1.924	1.772	61,203,283,968	380
SMOB-ECEIIT	6	0.131	1.149	0.838	626,401,280	156,589,824

Table 11: Official results of the WMT21 Quality Estimation Task 1 for the **English-Chinese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	4	0.908	0.393	0.316	2,244,030,744	560,981,507
papago (IKT)	4.6	0.901	0.393	0.288	2,503,797,760	611,278,859
HW-TSC	3	0.901	0.384	0.286	2,243,941,083	560,941,057
IST-Unbabel	5.8	0.899	0.393	0.289	4,872,322,439	1,214,683,792
TUDa	6.2	0.886	0.453	0.335	2,382,759,964	595,689,991
papago (KD)	4.6	0.879	0.427	0.316	1,249,902,592	297,974,795
BASELINE	5.4	0.818	0.556	0.408	1,142,413,043	281,291,535
Bergamot	5.6	0.687	1.024	0.748	70,044,344	16,772,151
SMOB-ECEIIT	5.8	0.650	0.794	0.628	626,401,280	156,589,824
RTM	n/a	0.287	3.749	3.607	61,203,283,968	380

Table 12: Official results of the WMT21 Quality Estimation Task 1 for the **Romanian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	3.4	0.812	0.488	0.393	2,244,030,744	560,981,507
• HW-TSC	4.4	0.808	0.520	0.409	2,243,941,083	560,941,057
IST-Unbabel	5.8	0.796	0.519	0.404	4,872,322,439	1,214,683,792
papago (KD)	5.2	0.794	0.510	0.397	2,503,797,760	611,278,859
TUDa	6.4	0.792	0.563	0.424	2,382,759,964	595,689,991
papago (IKT)	5	0.759	0.550	0.434	1,249,902,592	297,974,795
BASELINE	5.4	0.660	0.700	0.543	1,142,413,043	281,291,535
Bergamot-UTartu	6	0.547	1.840	1.701	1,705,478	421,537
Bergamot	5.8	0.544	0.966	0.761	284,339,184	70,969,501
SMOB-ECEIIT	7.6	0.329	1.072	0.862	1,886,937,088	471,716,864
RTM	n/a	0.099	2.520	2.346	61,203,283,968	380

Table 13: Official results of the WMT21 Quality Estimation Task 1 for the **Estonian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	4.8	0.867	0.570	0.426	2,244,030,744	560,981,507
HW-TSC	2.8	0.858	0.504	0.384	2,243,941,083	560,941,057
IST-Unbabel	5.2	0.856	0.515	0.401	4,872,322,439	1,214,683,792
papago (IKT)	5	0.853	0.522	0.399	2,503,797,760	611,278,859
TUDa	5.4	0.834	0.540	0.426	2,382,759,964	595,689,991
papago (KD)	5	0.823	0.562	0.441	1,249,902,592	297,974,795
BASELINE	5.4	0.738	0.657	0.524	1,142,413,043	281,291,535
Bergamot	5.6	0.626	0.977	0.818	83,907,600	19,220,401
SMOB-ECEIIT	5.8	0.544	0.931	0.717	626,401,280	156,589,824
RTM	n/a	0.127	2.286	2.017	61,203,283,968	380

Table 14: Official results of the WMT21 Quality Estimation Task 1 for the **Nepalese-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• IST-Unbabel	4.2	0.605	0.742	0.583	4,872,322,439	1,214,683,792
• QEMind	5	0.596	0.783	0.609	2,244,030,744	560,981,507
• papago (IKT)	4.6	0.595	0.745	0.585	2,503,797,760	611,278,859
papago (KD)	3.2	0.582	0.768	0.597	1,249,902,592	297,974,795
HW-TSC	4.8	0.581	0.776	0.602	2,243,941,083	560,941,057
TUDa	6	0.571	0.774	0.609	2,382,759,964	595,689,991
BASELINE	5	0.513	0.797	0.626	1,142,413,043	281,291,535
Bergamot	5.2	0.425	0.920	0.773	74,490,910	17,079,701
SMOB-ECEIIT	7	0.347	1.115	0.864	1,886,937,088	471,716,864
RTM	n/a	0.061	2.822	2.485	61,203,283,968	380

Table 15: Official results of the WMT21 Quality Estimation Task 1 for the **Sinhala-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" column indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	2.6	0.806	0.534	0.388	2,244,030,744	560,981,507
papago (IKT)	4.2	0.793	0.572	0.392	2,503,797,760	611,278,859
IST-Unbabel	5.4	0.792	0.583	0.412	4,872,322,439	1,214,683,792
HW-TSC	3.4	0.787	0.554	0.397	2,243,941,083	560,941,057
TUDa	5.8	0.764	0.629	0.437	2,382,759,964	595,689,991
papago (KD)	4.4	0.744	0.615	0.421	1,249,902,592	297,974,795
BASELINE	5	0.677	0.702	0.492	1,142,413,043	281,291,535
SMOB-ECEIIT	5.2	0.420	1.026	0.795	626,401,280	156,589,824
RTM	n/a	0.356	1.126	0.841	61,203,283,968	380

Table 16: Official results of the WMT21 Quality Estimation Task 1 for the **Russian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	3.4	0.582	0.746	0.599	2,244,030,744	560,981,507
• IST-Unbabel	5	0.577	0.751	0.583	4,872,322,439	1,214,683,792
• HW-TSC	3.8	0.573	0.747	0.602	2,243,941,083	560,941,057
• papago (IKT)	5	0.572	0.748	0.585	2,503,797,760	611,278,859
Inmon ‡	5.8	0.547	0.809	0.624	2,243,941,083	560,941,057
TUDa	6.2	0.545	0.808	0.619	2,382,759,964	595,689,991
papago (KD)	5.2	0.497	0.765	0.621	1,249,902,592	297,974,795
BASELINE	6	0.352	0.845	0.686	1,142,413,043	281,291,535
Bergamot-UTartu	6.2	0.300	1.420	1.166	111,300,550	27,815,809
SMOB-ECEIIT	6.4	0.195	1.199	0.967	626,401,280	156,589,824
RTM	n/a	-0.104	2.159	1.902	61,203,283,968	380

Table 17: Official results of the WMT21 Quality Estimation Task 1 for the **English-Czech** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	2.2	0.364	0.755	0.556	2,243,941,083	560,941,057
• QEMind	3.2	0.359	0.757	0.560	2,244,030,744	560,981,507
• IST-Unbabel	4.6	0.355	0.764	0.566	2,277,509,716	569,330,715
papago (IKT)	6	0.332	0.853	0.648	2,503,797,760	611,278,859
TUDa	6.6	0.330	0.917	0.705	2,264,844,300	566,211,075
Inmon ‡	5.6	0.297	0.882	0.665	2,243,941,083	560,941,057
papago (KD)	5	0.276	0.865	0.649	1,249,902,592	297,974,795
BASELINE	4	0.230	0.816	0.617	1,142,413,043	281,291,535
SMOB-ECEIIT	5.8	0.153	1.174	0.870	626,401,280	156,589,824
RTM	n/a	-0.082	2.694	2.576	61,203,283,968	380

Table 18: Official results of the WMT21 Quality Estimation Task 1 for the **English-Japanese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	2.8	0.647	0.736	0.605	2,244,030,744	560,981,507
• papago (IKT)	4	0.637	0.738	0.605	2,503,797,760	611,278,859
IST-Unbabel	5.8	0.628	0.780	0.658	4,872,322,439	1,214,683,792
HW-TSC	3.4	0.622	0.737	0.616	2,243,941,083	560,941,057
TUDa	6.2	0.609	0.824	0.674	2,382,759,964	595,689,991
Inmon ‡	5.2	0.592	0.795	0.665	2,243,941,083	560,941,057
papago (KD)	3.8	0.582	0.771	0.632	1,249,902,592	297,974,795
BASELINE	5.2	0.476	0.852	0.711	1,142,413,043	281,291,535
SMOB-ECEIIT	6.6	0.424	1.044	0.832	1,886,937,088	471,716,864

Table 19: Official results of the WMT21 Quality Estimation Task 1 for the **Pashto-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• QEMind	2.8	0.679	0.729	0.564	2,244,030,744	560,981,507
papago (IKT)	6	0.662	0.815	0.641	2,503,797,760	611,278,859
HW-TSC	3.6	0.659	0.744	0.578	2,243,941,083	560,941,057
IST-Unbabel	4.6	0.650	0.721	0.568	4,872,322,439	1,214,683,792
TUDa	4.8	0.639	0.740	0.585	2,382,759,964	595,689,991
Inmon ‡	4.8	0.630	0.765	0.599	2,243,941,083	560,941,057
papago (KD)	5.4	0.625	0.879	0.693	1,249,902,592	297,974,795
BASELINE	4.4	0.562	0.788	0.614	1,142,413,043	281,291,535
SMOB-ECEIIT	6.6	0.409	1.057	0.830	1,886,937,088	471,716,864

Table 20: Official results of the WMT21 Quality Estimation Task 1 for the **Khmer-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

B Official Results of the WMT21 Quality Estimation Task 2 (Sentence-level)

Tables 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 and 32 show the results for all language pairs and the multilingual variant, ranking participating systems best to worst using Pearson's r correlation as primary key for each of these cases.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
HW-TSC	1.4	0.631	0.202	0.153	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.597	0.219	0.171	2,294,887,576	569,368,609
BASELINE	2.2	0.502	0.235	0.188	1,142,441,796	281,297,685

Table 21: Official results of the WMT21 Quality Estimation Task 2 for the **Multilingual** variant. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	3	0.653	0.151	0.108	2,243,954,093	560,944,640
IST-Unbabel	4.6	0.617	0.172	0.116	2,294,887,576	569,368,609
Abulice ‡	4.2	0.577	0.174	0.115	2,243,439,613	560,814,661
POSTECH	4.6	0.546	0.172	0.139	1,561,188,430	390,210,052
Bergamot-UTartu	3.2	0.531	0.171	0.135	55,632,317	48
BASELINE	4.4	0.529	0.183	0.129	1,142,441,796	281,297,685
ENSBRT	4	0.520	0.171	0.129	1,363,652,116	502,000,000

Table 22: Official results of the WMT21 Quality Estimation Task 2 for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	2.6	0.368	0.297	0.248	2,243,954,093	560,944,640
Abulice ‡	2.4	0.312	0.340	0.280	100,000	9,501,148
IST-Unbabel	2.6	0.290	0.266	0.220	2,294,887,576	569,368,609
BASELINE	2.4	0.282	0.287	0.246	1,142,441,796	281,297,685
RTM	n/a	0.087	0.668	0.621	61,203,283,968	380

Table 23: Official results of the WMT21 Quality Estimation Task 2 for the **English-Chinese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• IST-Unbabel	2.2	0.879	0.122	0.098	2,294,887,576	569,368,609
HW-TSC	2.6	0.862	0.144	0.111	2,243,954,093	560,944,640
BASELINE	2	0.831	0.142	0.115	1,142,441,796	281,297,685
ENSBRT	3.2	0.795	0.171	0.141	1,363,652,116	502,000,000

Table 24: Official results of the WMT21 Quality Estimation Task 2 for the **Romanian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• IST-Unbabel	2.8	0.811	0.153	0.112	2,294,887,576	569,368,609
• HW-TSC	2.6	0.809	0.154	0.110	2,243,954,093	560,944,640
BASELINE	3.4	0.714	0.195	0.149	1,142,441,796	281,297,685
ENSBRT	3.2	0.666	0.171	0.132	1,363,652,116	502,000,000
Bergamot-UTartu	3	0.562	0.191	0.149	65,310,657	48

Table 25: Official results of the WMT21 Quality Estimation Task 2 for the **Estonian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	1.8	0.798	0.136	0.099	2,243,954,093	560,944,640
IST-Unbabel	2.8	0.718	0.161	0.126	2,294,887,576	569,368,609
BASELINE	2.6	0.626	0.205	0.160	1,142,441,796	281,297,685
ENSBRT	2.8	0.572	0.176	0.139	1,363,652,116	502,000,000

Table 26: Official results of the WMT21 Quality Estimation Task 2 for the **Nepalese-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	1.8	0.869	0.126	0.075	2,243,954,093	560,944,640
IST-Unbabel	2.8	0.710	0.178	0.136	2,294,887,576	569,368,609
BASELINE	2.2	0.607	0.204	0.159	1,142,441,796	281,297,685
ENSBRT	3.2	0.522	0.206	0.162	1,363,652,116	502,000,000

Table 27: Official results of the WMT21 Quality Estimation Task 2 for the **Sinhala-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	2	0.562	0.225	0.160	2,243,954,093	560,944,640
• IST-Unbabel	2.6	0.539	0.224	0.165	2,294,845,131	569,360,411
BASELINE	2.4	0.448	0.255	0.188	1,142,441,796	281,297,685
ENSBRT	3	0.376	0.251	0.189	1,363,652,116	502,000,000

Table 28: Official results of the WMT21 Quality Estimation Task 2 for the **Russian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• IST-Unbabel	2.4	0.529	0.271	0.200	2,294,887,576	569,368,609
HW-TSC	1.6	0.475	0.249	0.196	2,243,954,093	560,944,640
BASELINE	2	0.306	0.262	0.206	1,142,441,796	281,297,685

Table 29: Official results of the WMT21 Quality Estimation Task 2 for the **English-Czech** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• IST-Unbabel	2	0.275	0.279	0.224	2,294,887,576	569,368,609
• HW-TSC	1.8	0.262	0.278	0.228	2,243,954,093	560,944,640
BASELINE	2.2	0.098	0.279	0.232	1,142,441,796	281,297,685

Table 30: Official results of the WMT21 Quality Estimation Task 2 for the **English-Japanese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• IST-Unbabel	2.2	0.555	0.328	0.284	2,294,887,576	569,368,609
• HW-TSC	1.6	0.534	0.298	0.232	2,243,954,093	560,944,640
BASELINE	2.2	0.503	0.333	0.290	1,142,441,796	281,297,685

Table 31: Official results of the WMT21 Quality Estimation Task 2 for the **Pashto-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**Pearson** r	MAE	RMSE	Disk footprint (B)	# Model params
• HW-TSC	1.8	0.753	0.165	0.111	2,243,954,093	560,944,640
IST-Unbabel	3.4	0.655	0.243	0.199	2,294,887,576	569,368,609
BASELINE	2	0.576	0.241	0.196	1,142,441,796	281,297,685
ENSBRT	2.8	0.530	0.262	0.197	1,363,652,116	167,357,185

Table 32: Official results of the WMT21 Quality Estimation Task 2 for the **Khmer-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

C Official Results of the WMT21 Quality Estimation Task 2 (Word-level)

Tables 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43 and 44 show the results for all language pairs and the multilingual variant, ranking participating systems best to worst using Matthews correlation coefficient (MCC) as primary key for each of these cases.

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
HW-TSC	n/a	0.530	0.679	0.828	0.565	n/a	n/a
IST-Unbabel	n/a	0.430	0.628	0.787	0.486	n/a	n/a
BASELINE	n/a	0.346	0.579	0.717	0.402	n/a	n/a
				GAPs in MT			
HW-TSC	n/a	0.337	0.343	0.939	0.326	n/a	n/a
IST-Unbabel	n/a	0.196	0.209	0.975	0.203	n/a	n/a
BASELINE	n/a	0.126	0.137	0.973	0.133	n/a	n/a
				Words in SRC			
HW-TSC	n/a	0.432	0.592	0.799	0.473	n/a	n/a
IST-Unbabel	n/a	0.378	0.561	0.795	0.437	n/a	n/a
BASELINE	n/a	0.307	0.511	0.751	0.370	n/a	n/a

Table 33: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Multilingual** task. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• JHU-Microsoft	3	0.523	0.599	0.907	0.543	6,863,178,235	484,431,872
• HW-TSC	3.6	0.510	0.587	0.900	0.528	2,243,954,093	560,944,640
IST-Unbabel	3.8	0.466	0.551	0.914	0.504	2,294,887,576	569,368,609
Abulice‡	4.2	0.437	0.530	0.884	0.468	2,243,439,613	560,814,661
POSTECH	3	0.413	0.497	0.915	0.454	1,561,188,430	390,210,052
BASELINE	3.4	0.370	0.455	0.911	0.415	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	3.2	0.300	0.294	0.969	0.285	2,243,954,093	560,944,640
• JHU-Microsoft	3.4	0.256	0.266	0.985	0.262	6,863,178,235	484,431,872
IST-Unbabel	3.8	0.183	0.178	0.986	0.176	2,294,887,576	569,368,609
BASELINE	2.8	0.116	0.098	0.986	0.097	1,142,441,796	281,297,685
POSTECH	3.8	0.110	0.124	0.982	0.122	1,561,188,430	390,210,052
Abulice‡	–	–	–	–	–	–	–
				Words in SRC			
• HW-TSC	3.2	0.450	0.516	0.894	0.461	2,243,954,093	560,944,640
IST-Unbabel	3.8	0.404	0.483	0.921	0.445	2,294,887,576	569,368,609
Abulice‡	3.8	0.392	0.468	0.875	0.409	2,243,439,613	560,814,661
BASELINE	2.8	0.322	0.393	0.924	0.363	1,142,441,796	281,297,685
POSTECH	3.4	0.320	0.395	0.922	0.364	1,561,188,430	390,210,052
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• JHU-Microsoft	n/a	0.500	0.546	0.947	0.517	6,863,178,235	484,431,872
• HW-TSC	n/a	0.496	0.533	0.939	0.5	2,243,954,093	560,944,640
• IST-Unbabel	n/a	0.468	0.514	0.954	0.49	2,294,887,576	569,368,609
Abulice‡	n/a	0.442	0.488	0.934	0.456	2,243,439,613	560,814,661
BASELINE	n/a	0.378	0.42	0.952	0.4	1,142,441,796	281,297,685
POSTECH	n/a	0.403	0.45	0.952	0.428	1,561,188,430	390,210,052

Table 34: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	2	0.354	0.497	0.806	0.401	2,243,954,093	560,944,640
IST-Unbabel	3	0.310	0.467	0.792	0.370	2,294,887,576	569,368,609
BASELINE	3	0.247	0.426	0.723	0.308	1,142,441,796	281,297,685
JHU-Microsoft	4	0.149	0.357	0.751	0.268	6,863,178,235	484,431,872
Abulice‡	3	0.033	0.254	0.770	0.196	100,000	9,501,148
				GAPs in MT			
• HW-TSC	2.6	0.172	0.160	0.934	0.149	2,243,954,093	560,944,640
IST-Unbabel	3	0.068	0.083	0.982	0.082	2,294,887,576	569,368,609
BASELINE	2.4	0.065	0.092	0.969	0.089	1,142,441,796	281,297,685
JHU-Microsoft	3.6	0.035	0.051	0.981	0.050	6,863,178,235	484,431,872
Abulice‡	–	–	–	–	–	–	–
				Words in SRC			
• HW-TSC	2.2	0.310	0.443	0.813	0.360	2,243,954,093	560,944,640
IST-Unbabel	3.2	0.286	0.427	0.803	0.343	2,294,887,576	569,368,609
BASELINE	3.2	0.241	0.394	0.751	0.295	1,142,441,796	281,297,685
Abulice‡	3	0.011	0.222	0.769	0.171	100,000	9,501,148
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• IST-Unbabel	n/a	0.369	0.441	0.904	0.398	2,294,887,576	569,368,609
• HW-TSC	n/a	0.359	0.424	0.88	0.373	2,243,954,093	560,944,640
BASELINE	n/a	0.32	0.393	0.871	0.342	1,142,441,796	281,297,685
JHU-Microsoft	n/a	0.24	0.337	0.884	0.298	6,863,178,235	484,431,872
Abulice‡	n/a	0.118	0.228	0.884	0.201	100,000	9,501,148

Table 35: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **English-Chinese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	2	0.666	0.740	0.910	0.673	2,243,954,093	560,944,640
• IST-Unbabel	2.6	0.649	0.729	0.915	0.667	2,294,881,977	569,368,609
• JHU-Microsoft	2.6	0.634	0.713	0.922	0.657	6,863,178,235	484,431,872
BASELINE	2.8	0.536	0.642	0.862	0.553	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	2.2	0.446	0.449	0.974	0.437	2,243,954,093	560,944,640
IST-Unbabel	2.6	0.357	0.377	0.980	0.370	2,294,881,977	569,368,609
JHU-Microsoft	2.8	0.208	0.162	0.983	0.159	6,863,178,235	484,431,872
BASELINE	2.4	0.205	0.229	0.976	0.223	1,142,441,796	281,297,685
				Words in SRC			
• HW-TSC	2	0.614	0.694	0.898	0.623	2,243,954,093	560,944,640
• IST-Unbabel	2.6	0.603	0.689	0.910	0.627	2,294,881,977	569,368,609
BASELINE	2.6	0.511	0.618	0.871	0.539	1,142,441,796	281,297,685
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• HW-TSC	n/a	0.656	0.694	0.947	0.657	2,243,954,093	560,944,640
• IST-Unbabel	n/a	0.64	0.686	0.952	0.653	2,294,881,977	569,368,609
JHU-Microsoft	n/a	0.612	0.656	0.954	0.626	6,863,178,235	484,431,872
BASELINE	n/a	0.543	0.598	0.929	0.556	1,142,441,796	281,297,685

Table 36: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Romanian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.6	0.606	0.703	0.902	0.634	2,243,954,093	560,944,640
JHU-Microsoft	2.4	0.572	0.688	0.882	0.607	6,863,178,235	484,431,872
IST-Unbabel	3.2	0.570	0.687	0.880	0.605	2,294,887,576	569,368,609
BASELINE	2.8	0.461	0.589	0.869	0.512	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	2.2	0.312	0.334	0.969	0.324	2,243,954,093	560,944,640
IST-Unbabel	2.8	0.254	0.271	0.977	0.265	2,294,887,576	569,368,609
JHU-Microsoft	2.6	0.218	0.213	0.980	0.209	6,863,178,235	484,431,872
BASELINE	2.4	0.136	0.135	0.979	0.132	1,142,441,796	281,297,685
				Words in SRC			
• HW-TSC	1.8	0.549	0.650	0.899	0.584	2,243,954,093	560,944,640
IST-Unbabel	2.8	0.522	0.633	0.885	0.561	2,294,887,576	569,368,609
BASELINE	2.6	0.405	0.522	0.879	0.459	1,142,441,796	281,297,685
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• HW-TSC	n/a	0.584	0.644	0.94	0.605	2,243,954,093	560,944,640
• IST-Unbabel	n/a	0.582	0.644	0.937	0.604	2,294,881,977	569,368,609
• JHU-Microsoft	n/a	0.572	0.636	0.936	0.595	6,863,178,235	484,431,872
BASELINE	n/a	0.482	0.545	0.932	0.508	1,142,441,796	281,297,685

Table 37: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Estonian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.6	0.674	0.876	0.795	0.696	2,243,954,093	560,944,640
IST-Unbabel	2.6	0.508	0.842	0.652	0.549	2,294,881,977	569,368,609
BASELINE	2.2	0.440	0.828	0.583	0.483	1,142,441,796	281,297,685
JHU-Microsoft	3.6	0.329	0.813	0.299	0.243	6,863,178,235	484,431,872
				GAPs in MT			
• HW-TSC	2	0.403	0.435	0.961	0.418	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.268	0.284	0.969	0.276	2,294,881,977	569,368,609
BASELINE	2.2	0.215	0.249	0.963	0.240	1,142,441,796	281,297,685
JHU-Microsoft	3.4	0.207	0.253	0.953	0.241	6,863,178,235	484,431,872
				Words in SRC			
• HW-TSC	1.8	0.545	0.787	0.754	0.594	2,243,954,093	560,944,640
• IST-Unbabel	2.8	0.445	0.782	0.631	0.493	2,294,881,977	569,368,609
BASELINE	2.6	0.390	0.768	0.570	0.438	1,142,441,796	281,297,685
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• HW-TSC	n/a	0.749	0.833	0.915	0.763	2,243,954,093	560,944,640
IST-Unbabel	n/a	0.705	0.809	0.894	0.723	2,294,881,977	569,368,609
BASELINE	n/a	0.672	0.79	0.877	0.693	1,142,441,796	281,297,685
JHU-Microsoft	n/a	0.637	0.77	0.83	0.639	6,863,178,235	484,431,872

Table 38: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Nepalese-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.4	0.847	0.937	0.910	0.853	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.528	0.822	0.683	0.561	2,294,887,576	569,368,609
BASELINE	2.2	0.425	0.793	0.574	0.456	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	1.4	0.639	0.651	0.979	0.638	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.258	0.271	0.972	0.263	2,294,887,576	569,368,609
BASELINE	2.2	0.208	0.239	0.966	0.231	1,142,441,796	281,297,685
				Words in SRC			
• HW-TSC	1.4	0.616	0.804	0.810	0.651	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.406	0.722	0.627	0.452	2,294,887,576	569,368,609
BASELINE	2.2	0.335	0.698	0.544	0.379	1,142,441,796	281,297,685
				Combined MT Words & Gaps			
• HW-TSC	n/a	0.868	0.909	0.958	0.872	2,243,954,093	560,944,640
IST-Unbabel	n/a	0.69	0.79	0.896	0.708	2,294,881,977	569,368,609
BASELINE	n/a	0.642	0.758	0.87	0.660	1,142,441,796	281,297,685

Table 39: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Sinhala-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.8	0.451	0.553	0.892	0.493	2,243,954,093	560,944,640
IST-Unbabel	2.6	0.332	0.430	0.896	0.386	2,294,887,576	569,368,609
JHU-Microsoft	3	0.303	0.439	0.847	0.372	6,863,178,235	484,431,872
BASELINE	2.6	0.256	0.360	0.889	0.319	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	2.2	0.388	0.393	0.962	0.378	2,243,954,093	560,944,640
JHU-Microsoft	2.6	0.167	0.159	0.978	0.156	6,863,178,235	484,431,872
IST-Unbabel	3	0.165	0.160	0.978	0.156	2,294,887,576	569,368,609
BASELINE	2.2	0.073	0.051	0.979	0.050	1,142,441,796	281,297,685
				Words in SRC			
• HW-TSC	2.2	0.426	0.540	0.876	0.473	2,243,954,093	560,944,640
IST-Unbabel	2.6	0.351	0.438	0.899	0.394	2,294,887,576	569,368,609
BASELINE	2.4	0.251	0.326	0.893	0.292	1,142,441,796	281,297,685
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• HW-TSC	n/a	0.456	0.514	0.931	0.479	2,243,954,093	560,944,640
IST-Unbabel	n/a	0.339	0.39	0.941	0.367	2,294,881,977	569,368,609
JHU-Microsoft	n/a	0.329	0.406	0.919	0.373	6,863,178,235	484,431,872
BASELINE	n/a	0.319	0.939	0.299	0.139	1,142,441,796	281,297,685

Table 40: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Russian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.6	0.380	0.502	0.864	0.433	2,243,954,093	560,944,640
• IST-Unbabel	2.2	0.376	0.493	0.865	0.426	2,294,887,576	569,368,609
BASELINE	2.2	0.273	0.454	0.819	0.372	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	1.8	0.213	0.188	0.945	0.178	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.125	0.143	0.981	0.141	2,294,887,576	569,368,609
BASELINE	1.8	0.039	0.054	0.983	0.053	1,142,441,796	281,297,685
				Words in SRC			
• HW-TSC	1.4	0.313	0.426	0.886	0.377	2,243,954,093	560,944,640
• IST-Unbabel	2.4	0.294	0.410	0.883	0.362	2,294,887,576	569,368,609
BASELINE	2.2	0.224	0.362	0.862	0.312	1,142,441,796	281,297,685
				Combined MT Words & Gaps			
• IST-Unbabel	n/a	0.4	0.459	0.931	0.427	2,294,881,977	569,368,609
BASELINE	n/a	0.339	0.425	0.914	0.389	1,142,441,796	281,297,685
HW-TSC	n/a	0.336	0.427	0.909	0.388	2,243,954,093	560,944,640

Table 41: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **English-Czech** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.6	0.258	0.495	0.625	0.309	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.169	0.416	0.742	0.309	2,294,887,576	569,368,609
BASELINE	2	0.131	0.437	0.497	0.217	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	1.8	0.152	0.180	0.763	0.137	2,243,954,093	560,944,640
BASELINE	1.6	0.036	0.060	0.962	0.057	1,142,441,796	281,297,685
IST-Unbabel	2.6	0.025	0.016	0.969	0.015	2,294,887,576	569,368,609
				Words in SRC			
• HW-TSC	1.8	0.217	0.416	0.602	0.250	2,243,954,093	560,944,640
• IST-Unbabel	2.2	0.210	0.394	0.808	0.318	2,294,887,576	569,368,609
BASELINE	2	0.175	0.393	0.693	0.272	1,142,441,796	281,297,685
				Combined MT Words & Gaps			
BASELINE	n/a	0.25	0.403	0.79	0.319	1,142,441,796	281,297,685
IST-Unbabel	n/a	0.217	0.352	0.865	0.304	2,294,881,977	569,368,609
HW-TSC	n/a	0.186	0.361	0.677	0.244	2,243,954,093	560,944,640

Table 42: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **English-Japanese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.6	0.450	0.723	0.727	0.525	2,243,954,093	560,944,640
IST-Unbabel	2.6	0.370	0.685	0.684	0.469	2,294,887,576	569,368,609
BASELINE	2.4	0.313	0.674	0.631	0.425	1,142,441,796	281,297,685
JHU-Microsoft	3.4	0.191	0.677	0.170	0.115	6,863,178,235	484,431,872
				GAPs in MT			
• HW-TSC	2.2	0.260	0.262	0.942	0.246	2,243,954,093	560,944,640
IST-Unbabel	2.6	0.177	0.193	0.976	0.188	2,294,887,576	569,368,609
BASELINE	2	0.134	0.145	0.977	0.142	1,142,441,796	281,297,685
JHU-Microsoft	3.2	0.118	0.153	0.951	0.146	6,863,178,235	484,431,872
				Words in SRC			
• HW-TSC	2	0.304	0.538	0.723	0.389	2,243,954,093	560,944,640
• IST-Unbabel	2.6	0.294	0.522	0.758	0.396	2,294,887,576	569,368,609
BASELINE	2.6	0.249	0.501	0.720	0.361	1,142,441,796	281,297,685
JHU-Microsoft	–	–	–	–	–	–	–
				Combined MT Words & Gaps			
• IST-Unbabel	n/a	0.538	0.658	0.88	0.579	2,294,881,977	569,368,609
• HW-TSC	n/a	0.533	0.661	0.868	0.574	2,243,954,093	560,944,640
• JHU-Microsoft	n/a	0.523	0.648	0.782	0.507	6,863,178,235	484,431,872
BASELINE	n/a	0.517	0.648	0.867	0.562	1,142,441,796	281,297,685

Table 43: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Pashto-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-BAD	F1-OK	F1-Multi	Disk footprint (B)	# Model params
				Words in MT			
• HW-TSC	1.4	0.636	0.853	0.779	0.664	2,243,954,093	560,944,640
IST-Unbabel	2.4	0.448	0.790	0.638	0.503	2,294,887,576	569,368,609
BASELINE	.2	0.351	0.766	0.534	0.409	1,142,441,796	281,297,685
				GAPs in MT			
• HW-TSC	1.8	0.419	0.426	0.928	0.395	2,243,954,093	560,944,640
IST-Unbabel	2.2	0.259	0.274	0.964	0.264	2,294,887,576	569,368,609
BASELINE	2	0.175	0.204	0.959	0.195	1,142,441,796	281,297,685
				Words in SRC			
• HW-TSC	1.4	0.410	0.698	0.634	0.443	2,243,954,093	560,944,640
• IST-Unbabel	2.4	0.345	0.668	0.618	0.413	2,294,887,576	569,368,609
BASELINE	2.2	0.279	0.644	0.552	0.355	1,142,441,796	281,297,685
				Combined MT Words & Gaps			
• HW-TSC	n/a	0.677	0.783	0.883	0.692	2,243,954,093	560,944,640
IST-Unbabel	n/a	0.631	0.751	0.877	0.659	2,294,881,977	569,368,609
BASELINE	n/a	0.587	0.725	0.853	0.618	1,142,441,796	281,297,685

Table 44: Official results of the WMT21 Quality Estimation Task 2 (word-level) for the **Khmer-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000b). Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

D Official Results of the WMT21 Quality Estimation Task 3 (Sentence-level)

Tables 45, 46, 47 and 48 show the results for all language pairs, ranking participating systems best to worst using Matthews correlation coefficient (MCC) as primary key for each of these cases.

Model	Rank	**MCC**	F1-ERR	F1-NOT	F1-Multi	Disk footprint (B)	# Model params
• NICT Kyoto	1.5	0.546	0.877	0.667	0.585	2,239,774,281	559,892,482
LAMA-ICL	2.67	0.498	0.868	0.623	0.541	2,239,830,893	559,908,866
HW-TSC	4.17	0.490	0.867	0.613	0.532	2,241,232,523	561,947,562
QEMind	4	0.480	0.854	0.625	0.534	2,244,034,844	560,982,532
silence1024‡	4.33	0.449	0.850	0.597	0.507	2,239,747,529	560,365,209
BASELINE	4.33	0.397	0.848	0.532	0.451	1,114,634,523	278,635,778

Table 45: Official results of the WMT21 Quality Estimation Task 3 for the **English-German** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on William's test. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**MCC**	F1-ERR	F1-NOT	F1-Multi	Disk footprint (B)	# Model params
• HW-TSC	2.83	0.353	0.889	0.462	0.411	2,241,232,523	531,947,562
• silence1024‡	3.5	0.343	0.888	0.453	0.402	2,239,747,529	560,365,209
• NICT Kyoto	4	0.311	0.883	0.426	0.376	2,239,774,281	559,892,482
LAMA-ICL	4.33	0.305	0.892	0.413	0.368	2,239,830,893	559,908,866
QEMind	5.33	0.278	0.893	0.384	0.343	2,244,034,844	560,982,532
BASELINE	4	0.187	0.898	0.269	0.242	1,114,634,523	278,635,778
serkan‡	4	0.141	0.913	0.131	0.120	1,112,236,548	1,024

Table 46: Official results of the WMT21 Quality Estimation Task 3 for the **English-Chinese** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on William's test. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	Rank	**MCC**	F1-ERR	F1-NOT	F1-Multi	Disk footprint (B)	# Model params
• NICT Kyoto	1.83	0.511	0.913	0.595	0.543	2,239,774,281	559,892,482
• LAMA-ICL	2.17	0.473	0.911	0.555	0.506	2,239,765,357	559,892,482
QEMind	3.8	0.454	0.909	0.534	0.485	2,244,034,844	560,982,532
HW-TSC	3.33	0.448	0.906	0.537	0.486	2,234,153,425	560,365,209
BASELINE	3.67	0.388	0.899	0.477	0.429	1,114,634,523	278,635,778

Table 47: Official results of the WMT21 Quality Estimation Task 3 for the **English-Czech** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on William's test. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters).

Model	Rank	**MCC**	F1-ERR	F1-NOT	F1-Multi	Disk footprint (B)	# Model params
HW-TSC	2.5	0.318	0.937	0.378	0.354	2,239,747,529	560,365,209
LAMA-ICL	2.83	0.314	0.956	0.336	0.321	2,239,769,453	559,893,506
Jason_pogba‡	3.83	0.278	0.936	0.341	0.319	2,213,468,431	564,554,219
silence1024‡	4	0.277	0.940	0.337	0.317	2,239,747,529	560,365,209
QEMind	5.33	0.260	0.953	0.288	0.274	2,244,034,844	560,982,532
NICT Kyoto	5.17	0.252	0.929	0.319	0.297	2,239,774,281	559,892,482
BASELINE	4.33	0.214	0.951	0.244	0.232	1,114,634,523	278,635,778

Table 48: Official results of the WMT21 Quality Estimation Task 3 for the **English-Japanese** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on William's test. Baseline systems are highlighted in grey. "Rank" indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters). ‡ indicates Codalab usernames of participants from whom we have not received further information.

Findings of the WMT 2021 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT

Jindřich Libovický and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
{libovicky,fraser}@cis.lmu.de

Abstract

We present the findings of the WMT2021 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT. Within the task, the community studied very low resource translation between German and Upper Sorbian, unsupervised translation between German and Lower Sorbian and low resource translation between Russian and Chuvash, all minority languages with active language communities working on preserving the languages, who are partners in the evaluation. Thanks to this, we were able to obtain most digital data available for these languages and offer them to the task participants. In total, six teams participated in the shared task. The paper discusses the background, presents the tasks and results, and discusses best practices for the future.

1 Introduction

For some languages, machine translation (MT) reached such a high quality that allows a discussion of whether and under what circumstance human parity might have been reached (Popel et al., 2020; Läubli et al., 2020). This is the case, however, for only a small minority of the world's language. For most of the 7k languages spoken in the world only very limited resources exist. The goal of the WMT Shared Task on Unsupervised and Very Low Resource MT is to promote research on methods for MT that alleviate such data sparsity in a real-world setup.

A task on unsupervised MT was already held at WMT in 2018 (Bojar et al., 2018) and 2019 (Barrault et al., 2019), where the lack of parallel data was simulated on high-resource language pairs: English–German in 2018 and German–Czech in 2019.

Starting from last year, we cooperate with local communities working on preserving their languages. In cooperation with the Sorbian Institute[1] and the Witaj Sprachzentrum[2], we offered a shared task in translation between German and Upper Sorbian in low-resource and unsupervised tracks (Fraser, 2020). For this year, we kept the low-resource track for Upper Sorbian and added unsupervised translation between German and Lower Sorbian. Upper and Lower Sorbian are minority languages spoken in the east part of Germany in the federal states of Saxony and Brandenburg. Having only 30k and 7k native speakers, processing of the languages is an inherently low-resource problem, without any chance that the size of available resources would ever get close to the size of resources available for languages with millions of speakers. On the other hand, being western Slavic languages, the Sorbian languages can take advantage of existing resources for Czech and Polish.

Additionally, in cooperation with the Chuvash Language Laboratory[3], we added another low-resource task, translation between Russian and Chuvash. Chuvash is a minority Turkic language spoken by approximately one million people in the Volga region in the southwest of Russia. There is a larger amount of training data available for Chuvash, but the language is rather isolated in the Turkic language family, so unlike Sorbian, it cannot benefit that much from the existence of closely related languages.

Five teams participated in the German-Upper Sorbian task, six teams in the German-Lower Sorbian task, and two teams in the Russian-Chuvash task.

2 Tasks and Evaluation

This year, there were three tasks for very low resource and unsupervised translation were:

[1] https://www.serbski-institut.de
[2] https://www.witaj-sprachzentrum.de/
[3] https://en.corpus.chv.su/content/about.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 726–732
November 10–11, 2021.

- Very Low Resource Supervised Machine Translation: *German ↔ Upper Sorbian.*
- Unsupervised Machine Translation: *German ↔ Lower Sorbian.*
- Low Resource Supervised Machine Translation: *Russian ↔ Chuvash*.

To make the submissions better comparable with each other, we only allowed using resources released for the task (see Section 3) and resources for related languages commonly used in other WMT tasks. The use of large models pre-trained on large datasets was not allowed. By this decision, we wanted to motivate the participants to find better use of limited language resources.

German↔Upper Sorbian. There is only a very limited amount of parallel data between Upper Sorbian and German. However, because Upper Sorbian is closely related to Czech and Polish, we encouraged the use of all German, Czech and Polish data released for WMT. Other parallel data released from the WMT News Task were also allowed, but the participants were recommended not to use them. Unlike last year, there was no unsupervised task for Upper Sorbian.

German↔Lower Sorbian. For this task, no parallel training data were available, as the only available Lower Sorbian data were monolingual. Lower Sorbian is closely related to other Western Slavic languages, so the same related language data as for the Upper Sorbian task was allowed.

Russian↔Chuvash. The Chuvash language is not that critically low-resource as the Sorbian languages, but it is still affected by being a minority language. The participants were provided with parallel and monolingual data that we released for the task. Additional data that might be used: Chuvash-Russian part of the JW300 corpus (Agić and Vulić, 2019). In addition, the participants were encouraged to use the Kazakh–Russian corpus and monolingual Kazakh data from WMT19 (Barrault et al., 2019) and monolingual Russian data made available for the WMT News tasks.

Evaluation. Following the recent literature on MT evaluation (Mathur et al., 2020; Marie et al., 2021; Kocmi et al., 2021), we evaluate the systems using multiple evaluation measures, both string-based and model-based, and perform statistical testing to decide the ranking of the systems. In particular, we use the BLEU Score (Papineni et al., 2002), chrF score (Popović, 2015) as implemented in SacreBLEU (Post, 2018).[5] Further, we evaluate the models using BERTScore (Zhang et al., 2020)[6] with XLM-RoBERTa Large (Conneau et al., 2020) as an underlying model for German and Russian

Dataset	# lines	# chars.
German↔Upper Sorbian		
WMT20 parallel data	60k	11M
Parallel data provided by the Witaj Sprachzentrum, collected for the development of its own tranlator SoTra[4].		
Additional parallel data	87k	17M
Additional parallel Witaj Sprachzentrum collected since the last year.		
Sorbian Institute mono	340k	39M
Upper Sorbian monolingual data provided by the Sorbian Institute. This contains a high quality corpus and some medium quality data which were mixed together.		
Witaj mono	222k	19M
Upper Sorbian monolingual data provided by the Witaj Sprachzentrum (high quality).		
Web monolingual	134k	12M
Upper Sorbian monolingual data scraped from the web by CIS, LMU. This should be used with caution, it is probably noisy, it might erroneously contain some data from related languages.		
German↔Lower Sorbian		
Sorbian Institute mono	145k	14M
The sentences come from the Lower Sorbian reference corpus and were provided by the Sorbian Institute.		
Russian↔Chuvash		
Parallel corpus	714k	181M
A parallel corpus being collected by the Chuvash Language Laboratory since 2016 with the goal of promoting automatic processing of Chuvash.		
Bilingual dictionary	74k	182k
Monolingual Chuvash	5.6M	749M
The dataset contains monolingual sentences from various publicly available sources including Wikipedia, web crawl and fiction.		

Table 1: Overview of the data made available for the shared task.

[4] https://sotra.app/

[5] BLEU score signature `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0` chrF score signature `nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.0.0`

[6] https://github.com/Tiiiger/bert_score

Team	Architecture	Pre-training	Pre-training data	German / Russian mono.	BT iter.	BT filtering	Data tricks	Segmentation	Ensembling	Tookit
German ↔ Upper Sorbian										
NoahNMT	Big	de-cs	15M	100M	5	None		BPE	Yes	Inhouse
NRC-CNRC	Base	de-cs	16.5M	5M	2	Moore and Lewis (2010)	Tagged BT, BPE Dropout, Lang. tags	BPE	Yes	Sockeye
IICT-Yverdon	Base	de-cs	3M	1M	1	Length		SP	Yes	OpenNMT
CFILT	Base	mono de, hsb	2×.7M	.7M	60	None	BPE Dropout	BPE	No	MASS
LMU Munich	Big	de-cs	25M	15M	4	Length	Tagged BT	BPE	Yes	Fairseq
German ↔ Lower Sorbian										
NRC-CNRC	Base	de-cs, de-hsb	16.5M	147k to 5.2M	2	Moore and Lewis (2010)	BPE Dropout	BPE	Yes	Sockeye
IICT-Yverdon	Base	de-hsb	150k	1M	1	Length		SP	Yes	OpenNMT
CFILT	Base	de-hsb	3×.7M	.7M	60	None	BPE Dropout	BPE	No	MASS
CL_RUG	XLM	de-cs, de-hsb	8.5M +.8M	10.6M	2	None		BPE	No	MASS
LMU Munich	Big	de-cs, de-hsb	45M	15M	8	Length		BPE	Yes	Fairseq
Russian ↔ Chuvash										
NoahNMT	Big	en-ru	17M	110M	3	None	Domain adap.	BPE	Yes	Inhouse
LMU Munich	Big	ru-kk	11M	18M	2	Length	Tagged BT	BPE	Yes	Fairseq

Table 2: Overview of the method used by the task participants. SP stands for SentencePiece, BT for backtranslation.

and mBERT (Devlin et al., 2019) for Chuvash. We conduct the significance test using bootstrap resampling (Koehn, 2004) at a significance level of 0.95.

The final ranking is determined by the number of points each system gets. The systems get one point for each system that is significantly worse in each of the metrics. This means that if a system is significantly better than 1 system in the BLEU score, 2 systems in the chrF score, and 3 systems in the BERTScore, it gets 6 points in total.

3 Data

Upper Sorbian. The data for this task was provided by the Sorbian Institute (monolingual data) and The Witaj Sprachzentrum (Witaj Language Center) (both parallel and monolingual data).

The development and test data for Upper Sorbian are the same as the last year. There was a different blind test set than the last year.

Lower Sorbian. As far as we know, there is no parallel data for Lower Sorbian except for the development and test data provided for this task.

Chuvash. The validation data are sampled from the training set. The development test data and blind test data were also sampled from the parallel corpus and manually filtered by a native speaker.

In addition to the described data, the use of other parallel and monolingual data available for WMT News Tasks was allowed (see Section 2).

4 Submitted systems

Six teams participated in the shared task, five teams in Upper Sorbian-German, slightly different five in Lower Sorbian-German, and two in the Russian-Chuvash direction. An overview of the systems is in Table 2, a brief description of the systems follows. For detailed information, we refer the reader to the respective system description papers.

NoahNMT (Zhang et al., 2021b). NoahNMT submitted their systems into the supervised tasks. The NoahNMT submission is a standard Transformer model equipped with our recent technique of dual transfer (Zhang et al., 2021a). Compared to other systems, these submissions used a significantly larger amount of monolingual data.

NRC-CNRC (Knowles and Larkin, 2021). The Upper Sorbian-German system is an ensemble of eight systems with 25k BPE vocabulary, incorporating transfer learning (from cs–de) with continued training, monolingual data filtering, back-translation (Sennrich et al., 2016), BPE-dropout (Provilkov et al., 2020), and multilingual models.

Upper Sorbian → German

Team	BLEU	chrF	BERTScore	Points
NRC-CNRC	67.3 3	83.6 3	.981 4	10
NoahNMT	67.7 3	83.4 3	.981 3	9
LMU	64.3 2	81.9 2	.979 2	6
IICT-Yverdon	61.4 0	80.2 0	.976 1	1
CFILT	60.1 0	79.2 0	.975 0	0

German → Upper Sorbian

Team	BLEU	chrF	BERTScore	Points
NRC-CNRC	66.3 3	83.7 3	—	6
NoahNMT	65.9 3	83.3 3	—	6
LMU	63.3 1	81.9 2	—	3
CFILT	60.2 0	79.6 0	—	0
IICT-Yverdon	61.6 0	80.6 0	—	0

Lower Sorbian → German

Team	BLEU	chrF	BERTScore	Points
NRC-CNRC	33.5 1	63.8 1	.953 2	4
CL_RUG	32.4 1	62.2 1	.953 2	4
LMU	33.3 1	62.0 1	.952 1	3
CFILT	5.9 0	31.6 0	.884 0	0

German → Lower Sorbian

Team	BLEU	chrF	BERTScore	Points
NRC-CNRC	29.9 3	59.9 3	—	6
LMU	27.5 3	57.9 3	—	6
CL_RUG	24.1 2	54.2 2	—	4
IICT-Yverdon	8.0 0	32.1 1	—	1
CFILT	6.4 0	29.0 0	—	0

Chuvash → Russian

Team	BLEU	chrF	BERTScore	Points
NoahNMT	23.4 0	47.6 0	.944 1	1
LMU	22.0 0	46.3 0	.942 0	0

Russian → Chuvash

Team	BLEU	chrF	BERTScore	Points
NoahNMT	22.1 0	51.3 0	.857 1	1
LMU	20.9 0	50.1 0	.856 0	0

Table 3: The main results of the task. Points awarded in the particular metrics are in gray.

In the opposite direction, the submission is an ensemble of 7 systems. The Lower Sorbian-German and German-Lower Sorbian systems are ensembles of 2 and 4 systems, respectively, with 20k BPE vocabulary, incorporating transfer learning from hsb–de and de-hsb systems along with iterative backtranslation.

IICT-Yverdon (Atrio et al., 2021). The system used the Transformer architecture with back-translation of large German corpora and parent-language initialization using Czech-German data. The final submission is an ensemble of different models with some changes in their training setups to maximize the diversity among the models.

CFILT. The submitted systems cover four language pairs: German↔Upper Sorbian German↔Lower Sorbian. For de↔hsb, the system pre-trained using the MASS objective (Song et al., 2019) and finetuned using iterative back-translation. Final finetuning is performed using the provided parallel data for translation objective. For de↔dsb, no parallel data is provided in the task. The final de↔hsb model is used for initialization of the de↔dsb model, which is further trained using iterative back-translation, using the same vocabulary as used in the de↔hsb model.

CL_RUG (Edman et al., 2021). CL_RUG's submission uses the MASS model, focusing pre-training on 2 languages at a time, from least to most related to Lower Sorbian. The largest improvement comes from a novel method for initializing the Lower Sorbian word embeddings from Upper Sorbian, using a bilingual dictionary created in an unsupervised fashion.

LMU Munich (Libovický and Fraser, 2021). The LMU submissions for all tasks are Transformer models first pre-trained on related languages and then finetuned on the low-resource languages. For the Sorbian languages, the systems are pre-trained on German–Czech translation. The system is fine-tuned using the authentic German–Upper Sorbian data, which is the starting point for four iterations of tagged back-translation. The unsupervised German–Lower Sorbian translation is trained by iterative backtranslation using the monolingual data only. The Upper Sorbian–German system is used to generate the first translation of Lower Sorbian. The Russian–Chuvash systems were pretrained on Russian–Kazakh translation and finetuned using the provided parallel data.

5 Results

The results are presented in Table 3. The most successful teams were NRC-CNRC, which was the best or on par with the best systems in all Sorbian

tasks, and NoahNMT which were the best in the Chuvash tasks, on par with the best systems in German-Upper Sorbian translation and the second in the Upper Sorbian-German direction.

In German-Upper Sorbian translation, the best two system, NRC-CNRC and NoahNMT reach very similar results although they use significantly different sizes of monolingual data for backtranslation. NRC-CNRC manage to compensate for the smaller data size by accumulating minor tricks including monolingual data selection (Moore and Lewis, 2010), tagged backtranslation (Caswell et al., 2019), BPE dropout (Provilkov et al., 2020), and language tags in multilingual training. LMU, which used data of a similar size to NRC-CNRC but did not use most of the further tricks, ranked below these two.

In Upper Sorbian-German translation, all teams used German-Czech parallel data for pre-training, except for CFILT who only used monolingual data for pre-training and scored 0 points in both directions.

In the unsupervised German-Lower Sorbian task, CL_RUG ranked on par with NRC-CNRC in translation into German (despite not using ensembling), but at third place in the opposite direction. This suggests that CL_RUG's innovative vocabulary transfer method works better on the encoder side than on the decoder side.

In the Russian-Chuvash translation, NoahNMT outperformed LMU Munich by using larger datasets and a more advanced transfer learning technique.

6 Conclusions

In WMT 2021 shard task on Unsupervised and Very Low Resource MT, we created realistic benchmarks for low-resource minority language which reflect the needs of the language communities trying to preserve their languages. In the task, we provided the participants with comprehensive resource for translation between German and Upper and Lower Sorbian and for translation between Russian and Chuvash. We hope that this will increase the interest of the community in these languages.

The six teams that participated in the task used state-of-the-art MT techniques to develop high quality systems. The main technical takeaway from the results are that pre-training on parallel data in related language is important and that carefully applying known tricks can to a large extent compensate for using smaller datasets.

Acknowledgments

This work was also supported by the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement #640550) and by the DFG (grant FR 2829/4-1). Thanks very much to coorganizers: Hauke Bartels – Sorbian Institute, Olaf Langner – Witaj Sprachzentrum, Marcin Szczepanski – Sorbian Institute, Alexander Antonov – Chuvash Language Laboratory. Many thanks also to Tom Kocmi and Christian Federmann from Microsoft for providing an instance of Ocelot for the system submissions.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Àlex R. Atrio, Gabriel Luthier, Axel Fahy, Giorgos Vernikos, Andrei Popescu-Belis, and Ljiljana Dolamic. 2021. The iict-yverdon system for the wmt 2021 unsupervised mt and very low resource supervised mt task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lukas Edman, Ahmet Üstün, Antonio Toral, and Gertjan van Noord. 2021. Unsupervised translation of german–lower sorbian: Exploring training and novel transfer methods on a low-resource language. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Alexander Fraser. 2020. Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Rebecca Knowles and Samuel Larkin. 2021. NRC–CNRC systems for upper sorbian-german and lower sorbian-german machine translation 2021. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *CoRR*, abs/2107.10821.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human-machine parity in language translation. *J. Artif. Intell. Res.*, 67:653–672.

Jindřich Libovický and Alexander Fraser. 2021. The lmu munich systems for the wmt21 unsupervised and very low-resource translation task. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. 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.

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.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. 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.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Meng Zhang, Liangyou Li, and Qun Liu. 2021a. Two parents, one child: Dual transfer for low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2726–2738, Online. Association for Computational Linguistics.

Meng Zhang, Minghao Wu, Pengfei Li, Liangyou Li, and Qun Liu. 2021b. Noahnmt at wmt 2021: Dual transfer for very low resource supervised machine translation. In *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain

Markus Freitag[(1)], **Ricardo Rei**[(2,3,4)], **Nitika Mathur**[(5,6)], **Chi-kiu Lo**[(7)], **Craig Stewart**[(2)], **George Foster**[(1)], **Alon Lavie**[(2)], **Ondřej Bojar**[(8)]
[(1)]Google Research [(2)]Unbabel [(3)]INESC-ID [(4)]Instituto Superior Técnico
[(5)]The University of Melbourne [(6)]Oracle Digital Assistant
[(7)]National Research Council Canada (NRC-CNRC) [(8)]Charles University

Abstract

This paper presents the results of the WMT21 Metrics Shared Task. Participants were asked to score the outputs of the translation systems competing in the WMT21 News Translation Task with automatic metrics on two different domains: news and TED talks. All metrics were evaluated on how well they correlate at the system- and segment-level with human ratings. Contrary to previous years' editions, this year we acquired our own human ratings based on expert-based human evaluation via Multidimensional Quality Metrics (MQM). This setup had several advantages: (i) expert-based evaluation has been shown to be more reliable, (ii) we were able to evaluate all metrics on two different domains using translations of the same MT systems, (iii) we added 5 additional translations coming from the same system during system development. In addition, we designed three challenge sets that evaluate the robustness of all automatic metrics. We present an extensive analysis on how well metrics perform on three language pairs: English→German, English→Russian and Chinese→English. We further show the impact of different reference translations on reference-based metrics and compare our expert-based MQM annotation with the DA scores acquired by WMT.

1 Introduction

The metrics shared task[1] has been a key component of WMT since 2008, serving as a way to validate the use of automatic MT evaluation metrics and driving the development of new metrics. We evaluate reference-based automatic metrics that score MT output by comparing the MT with a reference translation generated by human translators, who are instructed to translate "from scratch" without post-editing from MT. In addition, we also invited submissions of reference-free metrics (quality estimation metrics or QE metrics) that compare MT outputs directly with the source segments. All metrics are evaluated based on their agreement with human rating when scoring MT systems and human translations at the system or sentence level. This year, we implemented several changes to the methodology that was followed in previous years editions of the task:

- **Expert-based human evaluation** This year, we collected our own human ratings for selected language pairs (en→de, en→ru, zh→en) from professional translators via MQM (Lommel et al., 2014). As shown before (Freitag et al., 2021), this produces more reliable[2] scores when compared to the DA-based human ratings acquired by the WMT News-Translation task. This step was necessary as Freitag et al. (2021) suggested that some automatic metrics already outperform (taking MQM as the golden standard) the DA-based human ratings that were usually used in the past for the metrics task and thus the DA-based ground-truth may be of lower quality than some of our submissions.

- **Additional Training Data** We encouraged the participants to further fine tune or test their metrics on the already existing MQM annotations for newstest2020 (Freitag et al., 2021)[3].

- **Additional domain** Since we collected our own human ratings, we were also able to expand the domain of the test sets beyond news and evaluate the performance of the metrics on translations of the same MT systems on TED talks, in order to test the generalization power of metrics.

[1] http://www.statmt.org/wmt21/metrics-task.html

[2] DA is unreliable for high-quality MT output; ranks human translations lower than MT; correlates poorly with metrics. Expert-based MQM ranks human translations higher than MT and correlates generally much better with automatic metrics.

[3] https://github.com/google/wmt-mqm-human-evaluation

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 733–774
November 10–11, 2021.

- **Additional MT systems** One use case for automatic metrics is choosing the better among different model versions of the same MT system during system development. To address this scenario, in addition to the WMT submissions and online systems, we added extra development systems to the set of MT systems on which we evaluated the metrics.

- **Additional challenge sets** We generated three challenge sets containing specific translation errors that are believed to be challenging for automatic MT evaluation metrics to identify. These challenge sets test metrics robustness on several different phenomena such as sentiment polarity, antonym replacement, named entities, among others.

- **Designated primary metrics** Participants had to designate a single metric as their primary submission for each track (reference-free and unconstrained). Other submissions were permitted, but only the primary metric is included in the official main results.

- **Accuracy for ranking system pairs** We calculate a joint score across all language pairs and adopt the pairwise accuracy score for ranking system pairs to generate the final metric ranking (Kocmi et al., 2021).

Our main findings are:

- WMT direct assessment (DA) scores generally correlate poorly with MQM scores, and exhibit weaker preference for human translations compared to machine output. In particular for English→German and Chinese→English, the two human evaluations methodologies produce very different rankings (Tables 16 and 18). In both language pairs, DA ranks the human translations below many MT systems, demonstrating again that expert-based evaluation is needed to generate a reliable ground truth for metric development for high quality language pairs.

- The majority of automatic metrics correlate better with MQM than the DA scores from WMT. This confirms the findings of Freitag et al. (2021) that automatic metrics are already more reliable than non-expert human evaluations. A metrics task with ground truth ratings acquired by non-experts would consequently not be very helpful.

- The performance of many metrics largely varies depending on the underlying domain (being either news or TED talks), resulting in distinct clusters of winning metrics for these two domains. All metrics of the winning cluster on the news domain show lower correlation with human ratings when switching to the TED talks domain (Table 8). Lower ranked metrics are more robust and can sometimes even improve the correlation to humans on the TED domain.

- Trainable embedding-based metrics are typically better at rating and correctly ranking (with respect to MQM golden truth) human-generated translations. (Table 8).

- Reference-free metrics, in particular COMET-QE and OpenKiwi perform very well when human translations are included in the setup. Nevertheless, once we focus on MT output only, reference-free metrics perform worse compared to reference-based metrics (Table 8).

- Reference-based metrics performance is significantly worse when reference translations contain major errors (Table 13).

- Some metrics are more robust than others when presented with alternate reference translations (Table 14). It is unclear so far what characterizes a good reference translation in addition to the clear requirement of fidelity of the translation to the source.

- When counting top performances across different language pairs, granularities, and test conditions (Table 12), three embedding-based metrics—C-SPECPN, BLEURT-20[4], and COMET-MQM_2021— emerge as distinctly better than the others, especially at the segment level and when rating human translations. Reference-free metrics are also relatively good at rating human translations, but under-perform at segment-level. Metric performance is distributed more evenly on

[4]BLEURT-20 denotes the new retrained version of BLEURT which is different from last years BLEURT submission (Sellam et al., 2020)

system-level tasks, especially when the test set is out-of-domain.

- Most metrics struggle to accurately penalize translations with errors in reversing negation or sentiment polarity (Table 9).
- Of the 14 linguistically motivated categories represented in the challenge sets, high-performing metrics have lower correlations for *Subordination* and *Named Entities and Terminology* (Tables 10 and 11).
- MQM annotations on TED data, both between annotation setups (Google and Unbabel) and between annotators themselves, show relatively low levels of agreement. However, we note that many of the system rankings remain relatively consistent; critically we note that the human reference comes out on top in both setups and that resulting metrics ranking is not significantly affected. This indicates that whilst MQM is an attractive framework for evaluation, the annotation task itself is still subject to human disagreement, especially on challenging content. TED talks in particular are highly specialized and ambiguous, presenting a unique challenge for annotators and evaluation.

2 Data

Similar to the previous years' editions, the source, reference texts, and MT system outputs for the metrics task are mainly derived from the WMT21 News Translation Task. This year, we expand the domain and evaluate the same MT systems on an additional out-of-domain data – TED talks, for our three primary language pairs: English→German, English→Russian and Chinese→English. In addition to the MT system outputs from the WMT evaluation campaign, we added translations from five additional MT systems that represent different versions of the same system during system development.

2.1 WMT Test Sets

The Newstest2021 set contains between 1000 and 2000 segments for each translation direction. All test sets are from the news domain. The reference translations provided for Newstest2021 were created in the same translation direction as the MT systems. We have two reference translations for English→Russian and Chinese→English and four reference translations for English→German. For more details regarding the news test sets, we refer the reader to the WMT21 news translation task findings paper.

2.2 TED Talks Test Suite

A long standing question about automated MT evaluation metrics has been whether metrics *generalize and perform well across domains*. In the past, metrics were mostly tested on news translation evaluation. The WMT2016 metrics shared task (Bojar et al., 2016) experimented on the IT and medical domains but the number of MT systems involved were small (2-10 in each translation direction). Thus, there was insufficient statistical evidence collected for a detailed analysis on how well metrics perform in different domains.

In an attempt to conduct a detailed analysis on the robustness of metrics when evaluating translations in a domain other than news, we generated and provided an additional test suite for translation by the MT systems participating in the news translation task, consisting of transcriptions of TED talks. The TED domain is quite different from the news domain, particularly in its more informal and disfluent language style, yet it covers a wide variety of topics and vocabularies.

The TED talk transcripts translation test set was extracted from OPUS[5] based on the corpus released by Reimers and Gurevych (2020). The English TED talk transcripts were translated by volunteers into multiple languages. To minimize the problem of translationese as the source for the Chinese→English part of the test suite, we had a first-language Chinese speaker select talks with Chinese translations that were judged to be natural-sounding in Chinese. (Unfortunately, there are still some problems in the translation quality for the Chinese→English part of the test suite which we will further discuss in Section 8.1.1.) Then, the same talks were extracted from the corpus to create the English→German and English→Russian parts of the test suite, where the translation was already available in the corpus and the quality of the translation was approved by professional translators. Table 1 shows the basic statistics of the TED talks test suite.

[5] https://opus.nlpl.eu/TED2020.php

language	#talks	#source sent.
en→de	6	606
en→ru	8	684
zh→en	9	843

Table 1: Statistics of the TED talks test suite.

2.3 Additional MT Output

One major use case for automatic metrics is choosing among different versions of the same system during system development. We translated all test sets for all language pairs with five different versions of the same system which we call *metricsystem{1,..,5}*. The underlying NMT models are trained on unconstrained training data and the model variations include baseline models, fine-tuned models and models considering document context. As we will see, the quality performance of these systems and their relative rankings can be quite different depending on the language pair, as these were not trained to yield the highest performance on the news or TED domain.

3 MQM Human Evaluation

Automatic metrics are usually evaluated by measuring correlations with human ratings. The quality of the underlying human ratings is critical and recent findings (Freitag et al., 2021) have shown that crowd-sourced human ratings are not reliable for high quality MT output. Furthermore, an evaluation schema based on MQM (Lommel et al., 2014) which requires explicit error annotation is preferable to an evaluation schema that only asks raters for a single scalar value per translation. Contrarily to the previous versions of the WMT metrics task, for our primary evaluation this year, we decided not to use the crowd-sourced DA human ratings from the WMT News Translation task, and conducted our own MQM-based human evaluation on a subset of submissions and a subset of language pairs that are most interesting for evaluating current metrics. This not only had the advantage of more reliable ratings for a subset of language pairs, but also gave us the opportunity to run the same human evaluation on a different domain (TED talks) on output generated by the same MT systems, in order to test the generalization capabilities of the metrics.

MQM is a general framework that provides a hierarchy of translation errors which can be tailored to specific applications. Google and Unbabel sponsored the human evaluation for this year's metrics task for a subset of language pairs using either professional translators (English→German, Chinese→English) or trusted and trained raters (English→Russian). The error annotation typology and guidelines used by Google's and Unbabel's annotators differs slightly and is described in the following two sections.

3.1 English→German and Chinese→English

Annotations for English→German and Chinese→English were sponsored and executed by Google, using 23 professional translators (14 for English→German, 9 for Chinese→English) with access to the full document context. Instead of assign a scalar value to each translation, annotators were instructed to "just" label error spans within each segment in a document, paying particular attention to document context. Each error was highlighted in the text, and labeled with an error category and a severity. To temper the effect of long segments, we imposed a maximum of five errors per segment, instructing raters to choose the five most severe errors for segments containing more errors. Segments that are too badly garbled to permit reliable identification of individual errors are assigned a special *Non-translation* error. Error severities are assigned independent of category, and consist of *Major*, *Minor*, and *Neutral* levels, corresponding respectively to actual translation or grammatical errors, smaller imperfections, and purely subjective opinions about the translation. Since we are ultimately interested in scoring segments, we adopt the weighting scheme shown in Table 2, in which segment-level scores can range from 0 (perfect) to 25 (worst). The final segment-level score is an average over scores from all annotators. For more details, exact annotator instructions and a list of error categories, we refer the reader to Freitag et al. (2021) as the exact same setup was used for the WMT21 metrics task.

Severity	Category	Weight
Major	Non-translation	25
	all others	5
Minor	Fluency/Punctuation	0.1
	all others	1
Neutral	all	0

Table 2: Google's MQM error weighting.

3.2 English→Russian

Annotation for English→Russian was performed by Unbabel who used a single professional native language annotator with several years of translation error experience based on variations of the MQM framework (Lommel et al., 2014). For this task, Unbabel provided a proprietary variant of MQM, specifically tailored for Russian language annotation. In a manner similar to the Google annotation, the annotator was given full document context and instructed to highlight spans of errors according to the categories specified in the typology. As with the Google annotation, the annotator was also instructed to indicate error severity. The Unbabel severity options differ slightly from that of Google in that we also specify a 'critical' error severity and do not specify a 'neutral' category. Additionally, in the Unbabel typology, all error categories are weighted equally within each severity level.

MQM scores at a segment level are calculated by summing the number of errors in the segment in each severity and applying a severity weight as described in Table 3. In contrast to the Google scheme, Unbabel does not impose a limit on the number of errors in a segment. We do, however, apply a normalization of the score by segment length. The full score calculation is shown in Equation 1 below:

$$\text{MQM} = 100 \cdot (1 - \frac{10 \cdot \#critical + 5 \cdot \#major + \#minor}{\#tokens}) \quad (1)$$

The same type of MQM annotations were previously used in the WMT QE shared tasks for the document-level subtask (Fonseca et al., 2019; Specia et al., 2020a) also sponsored by Unbabel.

Severity	Category	Weight
Critical	all	10
Major	all	5
Minor	all	1

Table 3: Unbabel's MQM error weighting.

3.3 Human Evaluation Results

As discussed in Section 1, we decided to run our own human evaluation in order to generate our golden-truth ratings and come to stronger conclusions about the quality of each automatic metric across two domains. Unfortunately, this also meant that we were only able to evaluate a subset of documents of newstest2021 and TED talks. In Table 4, you can see the number of segments for each language pair and test set that we used for human evaluation.

language	news	TED
en→de	527/1002	529/606
en→ru	527/1003	512/684
zh→en	650/1948	529/843

Table 4: Numbers of MQM-annotated segments for each test set.

The results of the MQM human evaluation can be seen in Table 5. Most of the reference translations are ranked first, surpassing all MT systems, except for *ref-B* for zh→en TED talks and *ref-A* for en→de newstest2021. This confirms the findings in Freitag et al. (2021) that when human evaluation is conducted by professional translators and MQM, high-quality human translations typically still outperform MT. We will discuss the impact of the identified low-quality reference translations in Section 8.1.1 in more detail. We wish to highlight one more important observation: the ranking of the MT systems is sharply different when switching from the commonly used Newstest2021 test sets to TED talks. This is particularly interesting for the metrics task, as metrics need to assess MT quality purely on the basis of the translations themselves and cannot rely on features that are specific to any particular MT system. We will analyse the differences between Google's and Unbabel's MQM approach in Section 8.2 and compare our MQM human evaluation with the DA assessment from WMT in more detail in Section 8.3.

4 Metric Submissions and Baselines

4.1 Baselines

SacreBLEU baselines We use the following metrics from the SacreBLEU *v1.5.0* (Post, 2018) as baselines, with the default parameters:

- BLEU (Papineni et al., 2002) is the precision of n-grams of the MT output compared to the reference, weighted by a brevity penalty to Using SacreBLEU we obtained sentence-BLEU values using the `sentence_bleu` python function and for corpus-level BLEU we used `corpus_bleu`. Both functions were used with the default arguments.[6]

[6]BLEU+case.mixed+lang.LANGPAIR-+numrefs.1 +smooth.exp+tok.13a-+version.1.5.0

English→German ↓

System	news	TED
ref.C	0.48 (1)	n/a
ref.D	0.52 (2)	n/a
ref.B	0.80 (3)	n/a
VolcTrans-GLAT	1.04 (4)	1.49 (6)
Facebook-AI	1.05 (5)	1.06 (2)
ref.A	1.22 (6)	0.91 (1)
Nemo	1.34 (7)	2.14 (14)
HuaweiTSC	1.38 (8)	1.50 (7)
Online-W	1.46 (9)	1.12 (3)
UEdin	1.51 (10)	1.77 (11)
eTranslation	1.69 (11)	1.96 (13)
VolcTrans-AT	1.74 (12)	1.24 (4)
metricsystem4	2.05 (13)	1.78 (12)
metricsystem1	2.07 (14)	1.63 (8)
metricsystem3	2.27 (15)	1.44 (5)
metricsystem2	2.58 (16)	1.69 (9)
metricsystem5	2.61 (17)	1.72 (10)

Chinese→English ↓

System	news	TED
ref.B	4.271 (1)	0.42 (1)
ref.A	4.35 (2)	5.52 (15)
metricsystem1	4.42 (3)	1.90 (4)
metricsystem4	4.62 (4)	2.05 (7)
NiuTrans	4.63 (5)	2.49 (11)
SMU	4.84 (6)	2.202 (9)
MiSS	4.93 (7)	1.97 (5)
Borderline	4.94 (8)	2.40 (10)
metricsystem2	5.04 (9)	1.76 (3)
DIDI-NLP	5.09 (10)	1.65 (2)
IIE-MT	5.14 (11)	1.98 (6)
Facebook-AI	5.21 (12)	2.64 (12)
metricsystem3	5.39 (13)	2.99 (14)
Online-W	5.57 (14)	2.93 (13)
metricsystem5	6.39 (15)	2.15 (8)

English→Russian ↑

System	news	TED
ref-A	99.65 (1)	97.51 (1)
ref-B	98.40 (2)	n/a
Facebook-AI	92.75 (3)	87.40 (3)
Online-W	91.80 (4)	90.84 (2)
metricsystem4	91.25 (5)	70.63 (11)
metricsystem5	90.88 (6)	74.15 (7)
metricsystem1	90.79 (7)	72.08 (9)
metricsystem2	89.86 (8)	75.19 (6)
Online-A	87.87 (9)	71.93 (10)
Nemo	87.50 (10)	73.77 (8)
Online-G	87.23 (11)	77.62 (5)
Manifold	86.86 (12)	68.27 (13)
Online-B	85.66 (13)	78.05 (4)
metricsystem3	85.65 (14)	60.17 (15)
NiuTrans	83.47 (15)	69.94 (12)
Online-Y	79.27 (16)	61.91 (14)

Table 5: MQM human evaluations for Newstest2021 and TED. Lower average error counts represent higher MT quality for En→De and Zh→En (using Google's formulation of MQM), while higher scores represent higher quality for En→Ru (using Unbabel's MQM definition).

- TER (Snover et al., 2006) measures the number of edits (insertions, deletions, shifts and substitutions) required to transform the MT output to the reference. As in BLEU, for TER we used SacreBLEU `sentence_ter` and `corpus_ter` functions (with default arguments[7]) to obtain segment-level and system-level scores.

- CHRF (Popović, 2015) uses character n-grams instead of word n-grams to compare the MT output with the reference. For CHRF we used the SacreBLEU `sentence_chrf` function (with default arguments[8]) for segment-level scores and we average those scores to obtain a corpus-level score.

BERTscore BERTSCORE (Zhang et al., 2020) leverages contextual embeddings from pre-trained transformers to create soft-alignments between words in candidate and reference sentences using a cosine similarity. Based on the alignment matrix, BERTSCORE returns a precision, recall and F1 score. We used F1 without TF-IDF weighting.

Prism PRISM (Thompson and Post, 2020) is an automatic MT metric which uses a sequence-to-sequence paraphraser to score MT system outputs conditioned on their respective human references. We used the default parameters with version 0.1 and model *m39v1*.

4.2 Submissions

The rest of this section summarizes participating metrics.

COMET All COMET* metrics (Rei et al., 2021) were built using the Estimator architecture presented in Rei et al. (2020a,b). The difference between all the submitted metrics stem from: the data used for training, the size of the encoder model and whether or not they take advantage of the reference translation.

- COMET-DA_2020 is the same model submitted for last year's shared task (Rei et al., 2020b; Mathur et al., 2020b) while COMET-DA_2021 is a retrained version of the previous model that includes the DA udgements collected in 2020.

- COMET-MQM_2021 is an MQM adaptation of the COMET-DA_2021 model that further trains for 1 additional epoch on MQM z-scores extracted from the MQM ratings for newstest2020 provided for the task this year.

- COMETINHO-MQM and COMETINHO-DA are lightweight versions of COMET-MQM_2021 and COMET-DA_2021 respectively, which use a distilled version of XLM-RoBERTa as the encoder.

[7] TER+lang.LANGPAIR+tok.tercom-nonorm-punct noasian-uncased+version.1.5.0

[8] chrF2+lang.LANGPAIR- +numchars.6+space.false- +version.1.5.0.

	Metrics	broad category	Citation	Availability
Baselines	SENTBLEU	lexical overlap	Papineni et al. (2002)	https://github.com/mjpost/sacrebleu
	BLEU	lexical overlap	Papineni et al. (2002)	https://github.com/mjpost/sacrebleu
	TER	lexical overlap	Snover et al. (2006)	https://github.com/mjpost/sacrebleu
	CHRF	lexical overlap	Popović (2015)	https://github.com/mjpost/sacrebleu
	BERTSCORE	embedding similarity	Zhang et al. (2020)	https://github.com/Tiiiger/bert_score
	PRISM	MT-model-based	Thompson and Post (2020)	https://github.com/thompsonb/prism
Participants	COMET-*	neural finetuned metrics	Rei et al. (2021)	https://github.com/Unbabel/COMET
	OPENKIWI-MQM	neural finetuned metrics	Kepler et al. (2019)	https://github.com/Unbabel/OpenKiwi
	YISI-*	embedding similarity	Lo (2019)	https://github.com/nrc-cnrc/yisi
	MTEQA	question-answer	Krubiński et al. (2021a)	https://github.com/ufal/MTEQA
	REGEMT-*	Ensemble	Stefanik et al. (2021)	https://github.com/MIR-MU/regemt
	ROBLEURT	neural finetuned metrics	Wan et al. (2021)	Not a public metric
	BLEURT-*	neural finetuned metrics	Sellam et al. (2020)	https://github.com/google-research/bleurt
	CUSHLEPOR-*	lexical overlap	Han et al. (2021)	https://github.com/poethan/cushLEPOR
	C-SPEC-*	neural finetuned metrics	Takahashi et al. (2021)	Not a public metric
	MEE-*	lexical and embedding similarity	Mukherjee et al. (2020)	https://github.com/AnanyaCoder/MEE_WMT2021

Table 6: Baseline metrics and participants of WMT21 Metrics Shared Task.

- Finally, COMET-QE-MQM_2021 and COMET-QE-DA_2021 are the reference-free versions of COMET-MQM_2021 and COMET-DA_2021 respectively.

From all the submitted models, the authors identified COMET-MQM_2021 and COMET-QE-MQM_2021 as their primary submissions to this years shared task edition.

OPENKIWI-MQM OPENKIWI-MQM (Kepler et al., 2019; Rei et al., 2021) is a multitask model that estimates a sentence-level MQM score along with word-level OK/BAD tags. The model is trained on top of XLM-RoBERTa using proprietary MQM data from several customer support domains. While word-level QE typically tags each word with an OK/BAD tag depending on post-edition information (Specia et al., 2020a), the OK/BAD tags used in OPENKIWI-MQM are derived directly from MQM annotation spans ignoring error types and/or severities.

YISI YISI (Lo, 2019) is a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources.

- YISI-1 is a reference-based MT evaluation metric. It measures the semantic similarity between a machine translation and human references by aggregating the idf-weighted lexical semantic similarities based on the contextual embeddings extracted from pretrained language models (e.g. BERT, CamemBERT, RoBERTa, XLM, XLM-RoBERTa, etc.).

- YISI-2 is the bilingual, reference-less version for MT quality estimation. It uses bilingual mappings of the contextual embeddings extracted from multilingual pretrained language models (e.g. XLM-RoBERTa) to evaluate the crosslingual lexical semantic similarity between the input and MT output.

YISI is an untrained metric and the submissions this year are the same as those in WMT20. The metric settings are described in Lo (2020) and Lo and Larkin (2020).

MTEQA MTEQA (Krubiński et al., 2021a,b) is an MT evaluation metric that leverages automatically generated questions and answers to assess the quality of MT systems. It builds upon the assumption that a good translation should preserve all of the key information that one can extract from the reference. Based on syntactic structure and NER system, they extract potential answers from the reference, and for each of them generate a human readable question. They then use a question-answering system to provide a new (test) answer given the question and the MT output as the context. The test answer is then compared to the reference answer to obtain the numerical score.

REGEMT REGEMT (Stefanik et al., 2021) is a family of ensemble metrics trained on MQM labels.

- {SRC, TGT}-REGEMT: This ensemble combine selected metrics of surface-, syntactic- and semantic-level similarity as input features to a regression model that estimates a quality assessment. Some of these features are newly introduced and some are based on related work. The reference-free ensemble uses as input features: Source length, Target length, Contextual SCM, Contextual WMD, BERTScore, Prism and Compositionality the reference-base ensemble uses: COMET, BLEURT, BLEU, METEOR, Non-contextual SCM and WMD.

- REGEMT-BASELINE: This ensemble uses only Source length and Target length of the given texts, in characters

The authors identified {SRC, TGT}-REGEMT as their primary submissions.

ROBLEURT ROBLEURT (Wan et al., 2021), short for Robustly Optimizing the training of BLEURT, is a model-based metric based on powerful language model XLM-RoBERTa. The ROBLEURT metric is constructed by the following steps: 1) jointly leveraging the advantages of source-included and reference-only metric models, 2) continuously pre-training the model with massive synthetic data produced by the real-world machine translation engines, and 3) fine-tuning the model with a data denoising strategy.

BLEURT BLEURT-20 and BLEURT-21-BETA are obtained by fine-tuning Rebalanced mBERT (Chung et al., 2021) (a multilingual variant of BERT) on a combination of two datasets: previous ratings from the WMT shared, task and generated data. The generated data consists of "perfect" sentence pairs, obtained by copying the reference into the hypothesis, as

well as “catastrophic" sentence pairs, obtained by randomly sampling tokens for each language pair. The fine-tuning methodology is similar to (Sellam et al., 2020). BLEURT-20 was trained on human ratings from WMT metrics 2015 to 2019 (z-scores) using WMT20 for test, and BLEURT-21-BETA was trained on WMT 2015 to 2020. The suffixes “-20” and “-21” denote the year of the WMT Metrics ratings that were used to build the test sets. The authors identified BLEURT-20 as their primary submission.

hLEPOR and cushLEPOR

- HLEPOR (Han et al., 2013) is an augmented metric with factors including enhanced sentence length penalty, precision, recall, and positional difference penalty which captures word order.

- CUSHLEPOR(LM) (Han et al., 2021) is a customized hLEPOR metric that uses LABSE pre-trained language model to automatically optimise hLEPOR parameters towards better correlation to human judgement and lower cost.

- CUSHLEPOR(PSQM) (Han et al., 2021) is trained and validated on the MQM and pSQM annotations from human professionals (Freitag et al., 2021). The tuned cushLEPOR achieves very high agreement to LABSE pre-trained language model in performance but uses much less computational cost as a distilled model.

The authors identified CUSHLEPOR(LM) as their primary submission.

C-SPEC C-SPEC (Takahashi et al., 2021) is designed for both segment-level and system-level translation evaluation. The authors' objective was to design a better metric by detecting significant translation errors that would not be ignored in real instances of human evaluation. Thus, pseudo-negative examples are generated in which selected words in the translation are replaced with alternatives based on a Word Attribute Transfer, and a metric model is built to handle such serious translation errors (denoted as C-SPECPN). A multilingual large pretrained model is fine-tuned on the provided corpus of past years' metrics task and fine-tuned again further on the synthetic negative samples that is derived from the same fine-tuned corpus. The authors identified C-SPECPN as their primary submission.

MEE

- MEE (Mukherjee et al., 2020) is an automatic evaluation metric that leverages the similarity between embeddings of words in candidate and reference sentences to assess translation quality focusing mainly on adequacy. Unigrams are matched based on their surface forms, root forms and meanings which aids to capture lexical, morphological and semantic equivalence. Semantic evaluation is achieved by using pretrained fasttext embeddings provided by Facebook to calculate the word similarity score between the candidate and the reference words. MEE computes evaluation scores using three modules namely exact match, root match and synonym match. In each module, fmean-score is calculated using harmonic mean of precision and recall by assigning more weight to recall. A final translation score is obtained by taking the average of fmean-scores from the individual modules.

- MEE2 is an improved version of MEE, focusing on computing contextual and syntactic equivalences along with lexical, morphological and semantic similarity. The intent of MEE2 is to capture fluency and context of the MT outputs along with their adequacy. Fluency is captured using syntactic similarity and context is captured using sentence similarity leveraging sentence embeddings. The final sentence translation score is the weighted combination of three similarity scores: a) Syntactic Similarity achieved by modified BLEU score; b) Lexical, Morphological and Semantic Similarity: measured by explicit unigram matching similar to MEE score; c) Contextual Similarity: Sentence similarity scores are calculated by leveraging sentence embeddings of *Language-Agnostic BERT* models.

The authors identified MEE2 as their primary submission.

5 Main Results

Currently, the main use case of automatic metrics is to rank systems either during system development or by comparing your own output with the one

from other research institutes or competitors. Consequently, we present system-level correlations as our main metric in this year's WMT21 metrics task. To be in line with the main use case, we present *pairwise accuracy* numbers for each metric that calculate the accuracy scores on binary comparison of system outputs for each language pair. We refer the reader to Section 7 for language pair specific results on both the segment and system level with more traditional correlation metrics.

5.1 System-Level

The system-level metric scores submitted by the participants pertained to the complete WMT test set, but we collected human MQM scores for only a subset of documents, as shown in Table 4. To correct for this discrepancy, we re-computed system-level scores as averages over the segments for which MQM scores were available, after first verifying with all participants that their system-level scores were computed in the same fashion.

To generate a single score combining the data from all 3 language pairs, we calculate *pairwise accuracy* (Kocmi et al., 2021) as our primary scoring metric. Pairwise accuracy is defined as follows: For each language pair and system pair, we calculate the difference of the metric scores (metricΔ) and the difference in average human judgements (humanΔ) for each system pair. We calculate accuracy for a given metric as the number of rank agreements between the metric and human deltas, divided by the total number of comparisons:

$$\text{Pairwise accuracy} = \frac{|\text{sign}(\text{metric}\Delta) = \text{sign}(\text{human}\Delta)|}{|\text{all system pairs}|} \quad (2)$$

We present results for three different settings: Looking at the news domain with and without human translations (HT) as additional systems: (a) *Newstest2021 w/o HT*, (b) *Newstest2021 w/ HT*, and (c) looking at *TED talks w/o HT*. In this section, we consider only the primary submissions of each metric team and the baseline metrics. We have multiple reference translations for some settings. Instead of reporting results with respect to all reference translations, we use here for reference-based metrics only the single reference that was judged best by the MQM raters for each language pair. The remaining reference translations are used in the role of participating MT systems in the "w/ HT" evaluations. Table 7 summarizes the use of reference translations for different language pairs and domains. We will analyse the impact of using different reference translations in Section 8.1 in more detail.

language	news		TED
	best ref	scored refs	best ref
en→de	C	A, D	A
en→ru	A	B	A
zh→en	B	A	B

Table 7: Use of reference translations.

Metric rankings based on pairwise accuracy can be found in Table 8. The top significance cluster (bolded in the table) consists of primary or baseline metrics that are not significantly outperformed by any other primary or baseline metrics nor outperformed by a primary or baseline metric not in the top cluster.[9]

Metric	newstest21 w/o HT	newstest21 w/ HT	TED w/o HT
tgt-regEMT	**0.773** (1)	0.694 (5)	0.636 (15)
Prism	**0.769** (2)	0.641 (7)	0.733 (5)
cushLEPOR(LM)	**0.763** (3)	0.622 (9)	0.647 (14)
C-SPECpn	**0.757** (4)	**0.784** (1)	0.704 (10)
bleurt-20	**0.753** (5)	0.718 (3)	0.749 (3)
MEE2	**0.753** (6)	0.628 (8)	0.713 (7)
BERTScore	**0.745** (7)	0.621 (10)	0.721 (6)
chrF	**0.745** (8)	0.621 (11)	0.713 (8)
BLEU	0.741 (9)	0.618 (12)	0.741 (4)
YiSi-1	0.737 (10)	0.615 (13)	**0.757** (2)
COMET-QE-MQM_21	0.733 (11)	**0.774** (2)	0.652 (13)
COMET-MQM_21	0.713 (12)	0.688 (6)	**0.773** (1)
MTEQA*	0.705 (13)	0.577 (15)	0.705 (9)
TER	0.696 (14)	0.585 (14)	0.636 (16)
OpenKiwi-MQM	0.692 (15)	0.698 (4)	0.680 (12)
RoBLEURT*	0.641 (16)	0.549 (16)	0.692 (11)
YiSi-2	0.510 (17)	0.429 (17)	0.494 (17)
src-regEMT	0.494 (18)	0.415 (18)	0.405 (18)

Table 8: Pairwise accuracy for Chinese→English, English→German, and English→Russian using the MQM annotations. Correlations for metrics in the top significance cluster are bolded. All submissions labelled with * participated only in 1 or 2 language pairs and are not considered during significance testing. Metrics not using reference translation (QE-metrics) are indicated by italics.

- **Newstest2021 w/o HT** This setting is most similar to previous years' settings. Metrics are required to score all MT outputs without considering human translations (HT). This setup investigates how metrics evaluate current SOTA MT

[9]Note that this definition is different from the metric clustering used in previous metrics tasks, in which every metric in a cluster must be significantly better than all metrics in lower clusters.

models. Looking at the ranking in Table 8, we can see that in total 8 metrics fall into the first significance cluster. The cluster includes a variety of embedding-based metrics and surface metrics. None of the QE metrics (i.e. reference-less metrics) are part of the first cluster.

- **Newstest2021 w/ HT** When considering the additional reference translations as system outputs (ref-A for zh→en, ref-B for en→ru, ref-A and ref-D for en→de), the ranking of the metrics is sharply revised. The QE metric *COMET-QE-MQM_2021* and the reference-based metric *C-SPECpn* are the winners in this setup. Overall, the embedding-based metrics that also rely on fine-tuning are much better in rating human translation higher than MT output and thus dominate this setting.

- **TED talks w/o HT** This year, we also measured the domain robustness of each metric on the TED talks domain. In Table 8, we can see that *COMET-MQM_2021* and *YiSi-1* show the highest correlation with human ratings on the TED domain. Interestingly, both metrics did not fall into the first significance cluster in the previous two settings of the news domain, leading to very different conclusion about the quality of metrics.

5.2 Significance Testing

We run PERM-BOTH hypothesis test (Deutsch et al., 2021) on the pairwise system-level accuracy of Table 8 to measure significance between metrics' performance.[10] Results can be seen in Figure 1. By looking at the heat map of Newstest2021 without human translations (*Newstest2021 w/o HT*), we observe that the top performing metrics are not significantly different. This observation changes when we add human translations to the setup (*Newstest2021 w/ HT*). The top 2 performing metrics, although different ones, are significantly better than all other metrics. This setup gives us the clearest result of all our 3 different setups and highlights that embedding-based metrics that are fine-tuned on previous years' human ratings rate human translations much better than all the other metrics and are good at distinguishing human-produced text. Another different situation can be seen when looking at the TED talk setting (*TED talks w/o HT*). Even though we see more significant differences compared to *Newstest2021 w/o HT*, most pairs of metrics are not significantly different.

[10]Previous editions of the metrics task used the Williams test (Williams, 1959), but we adopted PERM-BOTH because it is more general, and because Deutsch et al (2021) demonstrate that it has higher power.

6 Challenge Sets

While the correlation analysis is testing the evaluation metrics on their ability to rank MT systems according to translation quality, we are also interested in understanding metrics' performance on identifying certain types of translation errors. We created three challenge sets containing translation errors that are believed to be challenging for automatic MT evaluation metrics to identify. A good metric should not only rank candidate translations by their quality but also be sufficiently sensitive to these types of errors.

Each challenge set consists of two MT outputs (and the corresponding source and reference) where one of them contains the type of translation error of interest and the other does not. Metrics are expected to give a lower score to the MT output containing the error.

We use Kendall's tau-like correlation, typically used for DARR (Bojar et al., 2017; Ma et al., 2018, 2019; Mathur et al., 2020b), for evaluating the primary submissions on the challenge sets. Kendall's tau-like correlation is defined as follows:

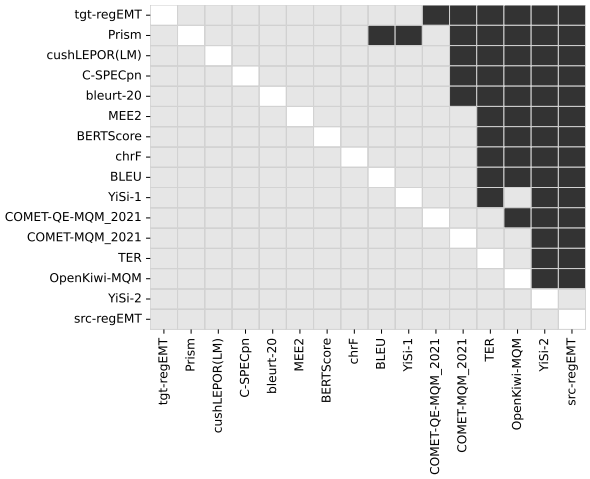
$$\tau = \frac{Concordant - Discordant}{Concordant + Discordant} \quad (3)$$

where *Concordant* is the number of times a metric assigns a higher score to the MT output without the error and *Discordant* is the number of times a metric assigns a higher score to the MT output containing the error of interest.

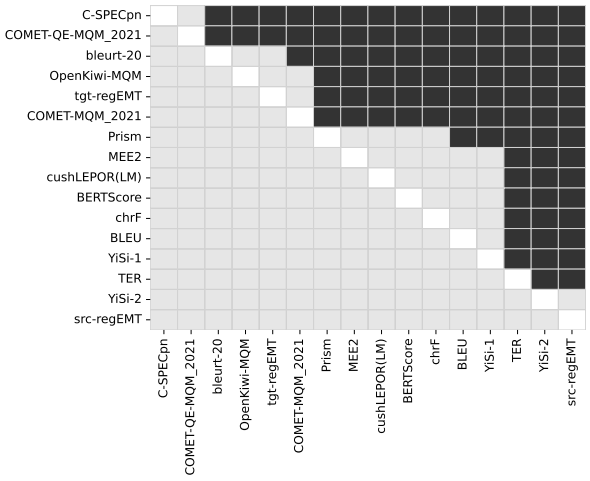
6.1 Negation and Sentiment Polarity Challenge Set

The goal of this challenge set is to test metrics' ability to penalize translations when there is a catastrophic error in reversing of a negation or of sentiment polarity. It is a common phenomenon that MT systems may either introduce or remove a negation (with or without an explicit negation word), or may reverse the sentiment polarity of the sentence (e.g. a negative sentence becomes positive or vice-versa). These types of errors could result in serious consequences of misleading users of MT.

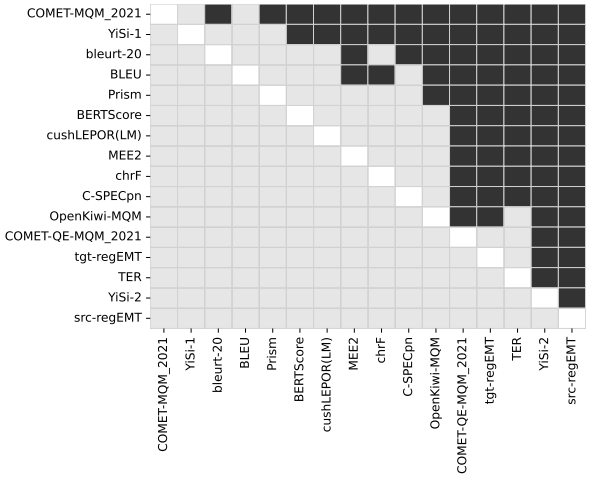
The WMT2020 MT Robustness shared task (Specia et al., 2020b) collected Wikipedia Edit comments with toxic content that could lead to possible



(a) newstest2021 w/o HT



(b) newstest2021 w/ HT



(c) TED talks w/o HT

Figure 1: The results of running PERM-BOTH hypothesis test to find a significant difference between metrics' pairwise system-level accuracy. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

catastrophic errors in the MT output. After selecting segments of interest they created reference translations for the entire test set using professional translators. Finally, they collected annotations of catastrophic errors on the translations performed by participating systems[11].

To test metrics on sentiment polarity we looked for source sentences from the English→German data portion where we can find an MT output annotated with a sentiment polarity error and another MT output without the polarity error. The resulting challenge set contains 177 source sentences (not necessarily distinct), each equipped with two MT outputs, one with a catastrophic error and one without it. We note that most of the sentences in this challenge set contain toxic language.

Table 9 shows the results for this challenge set. We also show the actual number of concordant pairs here because this challenge set is rather small. Despite the high severity of the translation error in reversing the sentiment polarity or negation, we see that both the baselines and the submissions struggle to accurately discriminate between translations with and without such errors. TER and BERTSCORE are the only two metrics that are able to achieve a medium correlation (i.e. greater than 0.4) with human annotators on ranking the translation with the catastrophic error as lower in translation quality. Perhaps more importantly, embedding-based and semantic-oriented metrics, such as BERTSCORE, YISI-1, etc., do not significantly outperform surface-form matching metrics, such as TER, CHRF and SENT-BLEU. This may indicate that the pretrained language models used by the embedding-based metrics are weak at learning language representations that explicitly reflect differences in negation and sentiment polarity.

6.2 Corrupted Reference Challenge Set

The goal of this challenge set is to sanity check the behaviour of the submitted metrics and possibly identify some weaknesses in detecting specific anomalies in a corrupted reference translation. In order to do this we used this years' Chinese→English Newstest corpus, which contains two human systems (referenceA and referenceB) and we perturb one of these human systems while using the other as reference. Given that, our final corpus is composed of $14,080$ tuples with

[11] Professional translators with access to the original source sentence, the reference and the system output were used during this evaluation.

Metric	*Concordant*	τ
TER	132	0.492
BERTSCORE	124	0.401
CHRF	123	0.390
YISI-1	122	0.379
MEE2	120	0.356
BLEURT-20	119	0.345
SENT-BLEU	118	0.333
C-SPECPN	118	0.333
COMET-MQM_2021	117	0.322
TGT-REGEMT	115	0.299
SRC-REGEMT	112	0.266
CUSHLEPOR(LM)	108	0.220
OPENKIWI-MQM	108	0.220
PRISM	107	0.209
MTEQA	106	0.198
COMET-QE-MQM_2021	106	0.198
YISI-2	104	0.175

Table 9: Results for the Negation and Sentiment Polarity Challenge Set. (Out of 177 hypothesis pairs)

(source, referenceBpert, referenceB, referenceA) where referenceBpert denotes the perturbed reference.

The perturbations used are: *antonym replacement*, *word omission*, *tokenization*, *sentence omission*,[12] *punctuation removal*, *number swapping*, *lowercasing*, *word addition* and addition of *spelling errors*. Table 22 in Appendix A shows examples for each perturbation.

From Table 10 we can observe that for most embedding based metrics (YISI, BERTSCORE, BLEURT-21-BETA, ROBLEURT, PRISM) correlations are close to 1.0 for all perturbation types. The only exceptions are COMET-MQM_2021 and C-SPECPN that seem to struggle with *sentence omission* and *punctuation removal*. This behaviour is even more unexpected if we take into consideration that they seem to be sensitive to *word omission*. Regarding punctuation removal, since both metrics are fine-tuned on Google MQM annotations (see Section 3.1) we hypothesize that they learn to be less sensitive to punctuation errors. Regarding the lexical metrics, we can observe that SENT-BLEU, CHRF and CUSHLEPOR(LM) are not sensitive to tokenized text. This is an expected behaviour for lexical metrics since they typically ignore whitespaces. Also, CUSHLEPOR(LM) scores -1.0 in lowercased text. This seems to indicate that this metric does not encode casing information.

6.3 German→English Challenge Set

The challenge set is based on the test suite by Macketanz et al. (2018a). It is a test suite for German-English that consists of around 5,500 German test sentences covering 107 linguistically motivated phenomena (listed in Avramidis et al. (2020)), organized in 14 categories. These phenomena do not follow a linguistic theory but rather cover various grammatical aspects which are relevant for MT. Each phenomenon is represented by at least 20 test sentences to guarantee a balanced test set. The test suite is used to evaluate MT systems with regard to their performance on the test sentences. The evaluation operates semi-automatically and is based on a set of handwritten rules which contain regular expressions and fixed strings.

The test suite has been used to evaluate the outputs of 40 German-English systems submitted at the translation task of the Conference of Machine Translation (WMT) for three consecutive years (Macketanz et al., 2018b; Avramidis et al., 2019, 2020) and also this year (Macketanz et al., 2021). Across the past three years, this amounts to 40 system outputs. We use these outputs to construct the challenge items for the metrics task, since the test suite contains only source sentences and handwritten rules for the outputs but no reference translations. For every source sentence of the test suite we separate MT outputs into "correct" and "incorrect" ones using the handwritten rules of the test suite and create a tuple including; (1) a set of "correct" MT outputs, to be given to the metrics as supposedly correct reference translations and, (2) a pair of one "incorrect" and one "correct" translation randomly sampled from the respective set. Note that the "correct" candidate does appear among the references (1). The goal of the metric is to score the "incorrect" translation worse than the "correct" one.

The same source sentence may be appear more than once, if there is more than one WMT translation marked as wrong by the rules for this item. The above process resulted in a metrics challenge set with 1,819 items with source, wrong hypothesis, correct hypothesis, and a pseudo-reference (another MT that was deemed correct for that phenomenon).

[12]Note that after experimenting with paragraph-level translation in WMT20, WMT21 moved back to segments again corresponding to individual sentences. In Chinese→English corpus, paragraph boundaries are not apparent (all documents consist of one paragraph). For the purposes of this experiment, we used *nltk.sentence_tokenizer* and looked for all the references B with more than 1 sentence and randomly delete 1 of those sentences to create referenceBpert. Note that since we do not have entire paragraph, the size of this challenge is 88 samples only.

Metric	Antonym	W. Omission	Tokenized	Sent. Omission	Punct.	Numbers	Lower.	W. Add.	Spell.
SENT-BLEU	0.792	0.787	-0.617	0.409	0.640	0.715	0.633	0.986	0.954
TER	0.994	0.597	0.966	0.568	0.739	0.996	1.000	1.000	0.997
CHRF	0.887	0.983	-0.516	0.523	0.761	0.899	0.708	0.903	0.981
BERTSCORE	0.986	0.984	0.994	0.909	0.950	0.993	0.799	0.996	0.998
PRISM	0.998	0.995	0.972	0.864	0.990	1.000	0.969	1.000	0.999
MTEQA	0.329	0.721	-0.522	0.273	-0.340	0.712	-0.415	0.649	0.624
YISI-1	0.991	0.996	0.993	0.977	0.951	0.996	0.960	0.999	1.000
BLEURT-20	0.992	0.989	0.983	0.909	0.931	0.993	0.976	0.998	0.997
COMET-MQM_2021	0.996	0.994	0.994	-0.068	0.235	0.993	0.965	0.993	1.000
C-SPECPN	0.991	0.988	0.576	0.409	0.622	0.876	0.922	0.991	0.996
CUSHLEPOR(LM)	0.826	0.779	-0.431	0.500	0.877	0.730	-1.000	0.982	0.957
MEE2	0.975	0.968	0.681	0.955	0.853	0.981	0.855	0.987	0.989
ROBLEURT	0.998	0.991	0.995	0.818	0.919	1.000	0.986	0.996	1.000
TGT-REGEMT	0.930	0.772	0.675	0.364	0.599	0.798	0.510	0.923	0.978
YISI-2	0.979	0.953	0.542	0.977	0.947	0.835	0.806	0.991	0.990
COMET-QE-MQM_2021	0.991	0.983	0.983	-0.318	-0.199	0.989	0.931	0.982	0.998
OPENKIWI-MQM	0.962	0.952	0.070	0.091	0.797	0.243	0.719	0.979	0.991
SRC-REGEMT	0.637	0.512	0.357	0.341	0.209	0.333	0.365	0.342	0.300

Table 10: Kendall's tau-like correlation results for the Corrupted References Challenge set (Section 6.2). The horizontal lines delimit baseline metrics (top), participating reference-based metrics (middle) and participating reference-free metrics (bottom).

The covered phenomena are: *Function Words (FW), Non-verbal Agreement (NVA), Verb Tense/Aspect/Mood (VT), Composition (Comp.), Multi-Word Expressions Negation (MWE N.), Punctuation (Punct.), Verb Valency (VV), Subordination (Sub.), Coordination and Ellipsis (CE), Named Entities and Terminology and Long Distance Dependencies and Interrogative (LDD).*

Overall, from Table 11 we observe that embedding-based metrics such as BLEURT-20 and COMET-MQM_2021 seem to be less sensitive to *Subordination*, *Named Entities and Terminology*, and to *Punctuation*. We can also observe a clear performance difference between reference-free and reference-based metrics. Nonetheless most metrics have positive correlations in all covered phenomena. Note that this corpus is composed of "pseudo-references" which can have a negative impact on metrics' performance (see Section 8.1).

7 Results per Language Pair

We computed individual correlation results for each focus language pair (English→German, English→Russian, Chinese→English) at both the system and segment level. The system-level metric scores submitted by the participants pertained to the complete WMT test set, but we collected human MQM scores for only a subset of documents, as shown in Table 4. To correct for this discrepancy, we re-computed system-level scores as averages over the segments for which MQM scores were available, after first verifying with all participants that their system-level scores were computed in the same fashion.[13] Exceptions to this pattern are the baseline metrics BLEU and TER: the system-level versions of these metrics are not averages over segment-level scores, and we computed them only over the MQM segments.

Since we have multiple reference translations for the focus language pairs, we required participants to submit versions of their (reference-based) metrics for each reference. We used only the scores corresponding to the reference that was judged best by the MQM raters for each language pair. For the news domain, we evaluated metric performance both when using only MT outputs and using MT outputs augmented by human references, adding all remaining references in the latter condition except for English→German, where we excluded reference B since it was very similar to the best reference C. Table 7 summarizes the use of reference translations for different language pairs and domains.

We measure correlation using the Pearson-r statistic at the system level and the Kendall-tau statistic at the segment level. Pearson correlation is complementary to the pairwise accuracy used for our global results as discussed in Section 5: it tests linear fit with MQM scores, a stringent but

[13] In contrast to the standard practice with WMT DA scores, where scored segments for each system are sampled independently, the segments for which we have MQM scores comprise a fixed set, independent of the MT system being scored.

Metric	FW	NVA	FF	VT	Comp.	Amb.	MWE N.	Neg.	Punct.	VV	Sub.	CE	NE & Term.	LDD
SENT-BLEU	0.50	0.66	0.32	0.37	0.09	0.42	0.38	0.77	0.64	0.42	0.48	0.43	0.30	0.52
TER	0.64	0.80	0.74	0.67	0.43	0.60	0.58	0.76	0.71	0.59	0.59	0.6	0.53	0.65
CHRF	0.42	0.56	0.63	0.57	0.37	0.70	0.46	0.71	0.43	0.51	0.38	0.45	0.44	0.54
BERTSCORE	0.61	0.59	0.89	0.76	0.76	0.74	0.60	0.71	0.68	0.77	0.47	0.63	0.48	0.68
PRISM	0.72	0.56	0.74	0.82	0.74	0.82	0.70	0.65	0.58	0.80	0.45	0.71	0.53	0.71
MTEQA	-0.64	-0.38	0.26	-0.77	0.03	0.54	-0.05	-0.59	-0.87	-0.57	-0.58	-0.30	0.34	-0.34
YISI-1	0.63	0.58	0.95	0.80	0.76	0.77	0.64	0.76	0.62	0.82	0.40	0.61	0.60	0.68
BLEURT-20	0.70	0.58	0.79	0.72	0.68	0.83	0.65	0.65	0.30	0.72	0.49	0.68	0.38	0.73
COMET-MQM	0.58	0.55	0.89	0.76	0.52	0.74	0.66	0.71	0.33	0.74	0.41	0.67	0.44	0.67
C-SPECPN	0.45	0.45	0.47	0.56	0.35	0.83	0.54	0.41	0.24	0.57	0.33	0.65	0.38	0.58
ROBLEURT	0.60	0.65	0.68	0.77	0.64	0.77	0.68	0.71	0.30	0.81	0.39	0.58	0.38	0.70
TGT-REGEMT	0.54	0.53	0.47	0.38	0.07	0.36	0.21	0.29	0.31	0.32	0.18	0.23	0.38	0.35
YISI-2	0.10	-0.03	0.11	0.36	0.37	0.29	0.03	0.18	0.45	0.35	0.11	0.25	0.13	0.42
COMET-QE-MQM	0.47	0.41	0.63	0.27	0.33	0.52	0.37	0.53	0.09	0.53	0.31	0.49	0.17	0.61
OPENKIWI-MQM	0.42	0.25	0.37	0.45	0.23	0.47	0.44	0.53	0.41	0.56	0.27	0.62	0.27	0.47
SRC-REGEMT	0.45	0.08	-0.05	0.29	0.41	0.26	0.00	-0.06	0.37	0.44	0.05	0.05	0.12	0.18
Average	0.45	0.43	0.56	0.49	0.42	0.60	0.43	0.48	0.35	0.52	0.30	0.46	0.37	0.51

Table 11: Kendall's tau-like correlation results for the German→English challenge set based on (Macketanz et al., 2018a) test suite. Note that not all metrics submitted to this challenge set hence some metrics are missing. The horizontal lines delimit baseline metrics (top), participating reference-based metrics (middle) and participating reference-free metrics (bottom).

Metric	Total "wins"	Language Pair en→de	en→ru	zh→en	Granularity sys	seg	Data condition news w/o HT	news w/ HT	TED
C-SPECpn	11	4	3	4	6	5	3	5	3
bleurt-20	10	4	5	1	4	6	4	3	3
COMET-MQM_2021	10	3	3	4	3	7	3	2	5
tgt-regEMT	4	1	1	2	3	1	2	1	1
COMET-QE-MQM_2021	3	1	1	1	3			3	
OpenKiwi-MQM	3	2		1	3		1	2	
RoBLEURT*	3			3	1	2	1		2
cushLEPOR(LM)	2	1		1	2		1		1
BERTScore	2	1	1		2		1		1
Prism	2		2		2		1		1
YiSi-1	2		2		2		1		1
MEE2	2	2			2		1		1
BLEU	1	1			1		1		
hLEPOR	1		1		1				1
MTEQA*	1			1	1				1
TER	1			1	1				1
chrF	1			1	1				1

Table 12: Summary of language-specific results. Numbers give the count of times each primary metric occurred in the top cluster for the specified condition. Metrics not being among the winners in any competition are not listed. Reference-free metrics are indicated by italics. All submissions labelled with * participated only in 1 or 2 language pairs.

reasonable criterion since we expect these scores to conform to a linear scale (for example, a translation with two minor errors is twice as bad as one with only a single error). Pearson has well-known drawbacks (Mathur et al., 2020a), notably sensitivity to outliers, which we avoided by choosing only relatively high-performing systems. In preliminary tests, Pearson also yielded a larger number of pairwise significant differences among metrics than Kendall, an important property since our fairly small number of systems makes it difficult to reliably distinguish metrics at the system level.

Segment-level scores—metric or human—are naturally arranged as a system × segment matrix (rows × columns). There are several ways to extract vectors for input to correlation statistics. Comparing metric and human row vectors corresponds to a use case of judging the relative quality of different segments output by a given MT system ("where is my system making mistakes on this test set?"); comparing column vectors corresponds to judging the relative quality of outputs for a given source

segment across different MT systems ("which systems performed better or worse than mine on this segment?"). To avoid emphasizing either of these scenarios at the expense of the other, we flattened the metric and human score matrices into single vectors (row1, row2, ...) before comparing them. This measures the metrics' ability to assign independent scores to MT segments, abstracting away from system or source segment, and provides a large number of comparisons to boost statistical significance. We used Kendall rather than Pearson correlation for robustness to segment-level noise.[14]

The results for each language pair and granularity are shown in Tables 23 to 28, with corresponding pairwise significance plots derived using the PERM-BOTH test in Figures 2 to 7. The tables contain results for all metrics; the significance plots include only primary and baseline metrics. In the tables, primary submissions are in bold, baseline metrics are underlined, and metrics that used only the source have "-src" appended to their name. For each condition (news without human translations, news with human translations, or TED), the *scores* of primary and baseline metrics in the top cluster are in bold. The top cluster consists of primary or baseline metrics that are not significantly outperformed by other primary or baseline metrics nor outperformed by a primary or baseline metric not in the top cluster.[15] In the significance plots, this corresponds to the leftmost block of columns containing no dark squares.

Table 12 summarizes all results in this section by counting the number of times each metric occurs in the top cluster (it got a "win"), summed across different ways of partitioning the results. This synthesis is fairly crude, since it treats all conditions as equally important. Also, membership in the top cluster is likely to be subject to high statistical variance, and metrics that fall outside this cluster are not accounted for; in particular, those that sometimes perform very poorly are not penalized. Nevertheless, the counts permit some general observations.

In terms of total "wins", three metrics stand out clearly: C-SPECPN, BLEURT-20, and COMET-MQM_2021. These have fairly evenly-distributed performance across languages, granularities, and data conditions, with the exception of BLEURT-20, which does relatively poorly on Chinese→English. Their advantage over other metrics is most pronounced at the segment level and when human translations are included among the systems to be judged (*w/ HT*)—both of which are more challenging tasks. In contrast, the distribution of metrics that achieve top-level performance is much broader for system-level granularity, the out-of-domain TED setting, and to a lesser extent the news *w/o HT* setting. Two metrics that do not use a reference translation—COMET-QE-MQM_2021-src and OpenKiwi-MQM-src—do surprisingly well overall, particularly in the *w/ HT* condition, but perform poorly at the segment level. This could be explained by these metrics benefiting from their ability to distinguish human vs. machine produced text. Finally, the surface-level baselines—BLEU, TER, and chrF—join the winners exclusively at the system level and almost exclusively in the out-of-domain TED condition.

8 Additional Results

8.1 Impact of Reference Translation

The quality of the reference translation can have a higher impact on the correlation to human ratings than the actual choice of metric (Freitag et al., 2020). For all our different test sets and language pairs, we consequently included all reference translations in our human evaluation to (a) assure that we have reference translation with high quality and (b) to choose the best reference translation for our main results. In this section, we present two interesting observations by looking into the Chinese→English TED talks and the English→German news setups.

8.1.1 zh→en TED

We started by having only one reference translation for all TED talks. Unfortunately, the MQM evaluation revealed that the reference translation *ref-A* for Chinese→English was ranked last – lower than all the MT systems – and that it contained on average more than one major error (= 5 MQM points) per segment. We spot checked the errors and agreed that the reference translation indeed contained many errors. We then decided to acquire

[14] Our use of Kendall differs in two major aspects from the "Kendall-like" statistic used for segment-level correlations in previous editions of the WMT metrics task: we do not threshold MQM score differences, as we consider them to be more reliable than DA scores; and we compare all pairs of scores over complete flattened matrices rather than comparing pairs of scores in each column, and micro-averaging results across columns.

[15] Note that this definition is different from the metric clustering used in previous metrics tasks, in which every metric in a cluster must be significantly better than all metrics in lower clusters.

	ref-A	ref-B
MQM	5.52	0.42
MTEQA	0.47 (3)	0.74 (1)
TER	0.40 (9)	0.71 (2)
BERTScore	0.42 (6)	0.69 (3)
bleurt-20	0.45 (5)	0.68 (4)
cushLEPOR (LM)	0.39 (11)	0.68 (5)
Prism	0.46 (4)	0.68 (6)
COMET-MQM_2021	0.40 (8)	0.67 (7)
BLEU	0.30 (13)	0.65 (8)
YiSi-1	0.42 (7)	0.65 (9)
chrF	0.40 (10)	0.62 (10)
MEE2	0.36 (12)	0.60 (11)
C-SPECpn	0.49 (2)	0.54 (12)
tgt-regEMT	0.5 (1)	0.37 (13)
average	0.42	0.64

Table 13: Pairwise accuracy for ranking system pairs for TED Chinese→English using either ref-A (original ref) or ref-B (extra generated ref).

a new reference translation (*ref-B*) which turned out to be better than all MT systems after running a human evaluation. The impact of using an excellent versus an inaccurate human reference translation can be seen in Table 13. All metrics achieve an accuracy score lower than 0.5 when using ref-A to calculate their scores. This means that the metrics would perform worse than by chance. By switching to ref-B, all but one metric (*tgt-regEMT*) greatly improve their correlation score. This demonstrates once again that metrics become unreliable when they are provided with inaccurate reference translations.

8.1.2 en→de Newstest2021

For English→German Newstest2021, we started with two reference translations (*ref-A* and *ref-B*). Both reference translations had issues: ref-A was ranked lower than two MT systems (see Table 5) and we agreed with that assessment after spot checking the errors. ref-B had high-levels of overlap with the Online-W MT system and is most likely a post-edited translation of Online-W. Therefore, Microsoft and Google sponsored two new reference translations (*ref-C* and *ref-D*) which turned out to be the best translations based on MQM. In Table 14, you can see the pairwise accuracy scores from all reference-based primary and baseline metrics. Despite the good (low) MQM scores of both ref-C and ref-D, the ranking of the metrics when using these two references is quite different. Some metrics are more robust when switching the reference translation (e.g. *Prism*, *YiSi-1*, or *C-SPECpn*, but others yield very different correlation scores (e.g. *BERTScore*, *tgt-refEMT*, or *BLEU*). Something else in addition to quality makes *ref-C* more appealing for metrics than *ref-D*. We do not have an explanation why the quality of some metrics is so different when switching the reference translation and leave this as an open challenge for the community to better understand why this is happening.

	ref-A	ref-C	ref-D
MQM	1.22	0.48	0.52
BERTScore	0.91 (1)	0.94 (1)	0.77 (10)
cushLEPOR (LM)	0.81 (10)	0.92 (2)	0.81 (6)
BLEU	0.87 (5)	0.90 (3)	0.69 (12)
MEE2	0.87 (4)	0.90 (4)	0.80 (8)
TER	0.89 (2)	0.90 (5)	0.80 (7)
chrF	0.82 (8)	0.87 (6)	0.77 (11)
bleurt-20	0.86 (6)	0.85 (7)	0.81 (4)
Prism	0.83 (7)	0.83 (8)	0.81 (5)
YiSi-1	0.87 (3)	0.82 (9)	0.82 (2)
C-SPECpn	0.80 (11)	0.82 (10)	0.82 (3)
COMET-MQM_2021	0.81 (9)	0.80 (11)	0.77 (9)
MTEQA	0.78 (13)	0.80 (12)	0.67 (13)
tgt-regEMT	0.78 (12)	0.80 (13)	0.82 (1)
average	0.84	0.86	0.78

Table 14: Pairwise accuracy for ranking system pairs for newstest2021 w/o HT English→German using either ref-C (main ref) or ref-A/ref-D (alternative refs), where ref-A is of substantially lower quality. ref-B was excluded because it is likely a post-edit of one of the participating systems.

8.2 Google vs. Unbabel MQM

Given that annotations were undertaken for English→Russian using a different setup and MQM scheme than those for English→German and English→Chinese we sought to provide some insight into the compatibility of the two schemes by repeating the annotation for English→German using Unbabel's scheme and annotator pool: For a subset of 5056 segments of the TED talk data for English→German from 10 MT systems, Unbabel had another expert annotator trained on MQM provide annotations using their proprietary typology. MQM was calculated for each set of annotations (using their respective scoring) and the latter were then converted to a sequence of OK/BAD tags as a means of evaluating the level of agreement between the two annotations at a token level.

The Pearson's r correlation score between the two sets of MQM annotations was found to be 0.212, significant to p<0.05. Given the levels of correlation of metrics with Google's MQM scores on the full set of English→German, this is surprisingly low. Similarly, Cohen's Kappa on the

annotated tags was found to be 0.165. Not only do scores correlate poorly but agreement at the tag level is also fairly weak. Equally, Cohen's Kappa on the subset of annotations on which both sets of annotators found some error was found to be improved but still low (0.2). This indicates that even when limited to erroneous sentences, the annotators struggled to agree on where the errors were.

We note that the Google annotators left 59.5% of the sample untouched (i.e. error free), whereas the Unbabel annotator left only 46.9% untouched. It appears that the Unbabel annotator was on average more aggressive in their annotation which might partially explain low levels of agreement.

A number of the MT systems often produced the same translation of the same source text. With this in mind, and given that Google used a pool of annotators, we were able to also compare annotations within the Google set. For every source/target pair with more than two annotations we calculated and averaged the pairwise Cohen's Kappa. The mean Kappa across all of these segments was 0.21, which suggests equally low levels of agreement between Google annotators.

Despite low segment level agreement we note that the ranking of systems remains fairly consistent between annotation schemes with a few outlying exceptions. Table 15 details the rankings for our sample across annotation schemes. In particular it is encouraging to note that the human reference (albeit one of the worse ones, see Section 8.1.2) is ranked first in both cases; at a high level both schemes are making meaningful quality judgements. For the sake of completeness, we similarly examined the rankings of metrics at segment level (measuring Pearson's r correlation score and ranking the result) against both sets of MQM scores for our sample. Rankings in both cases were found to be sufficiently similar to official results reported in this paper and no metric moved more than three positions.

To rationalize these low segment-level agreement numbers, we asked an independent native language German speaker to look at a subset of annotations where we noticed the worst levels of segment-level agreement. The independent rater provided some rudimentary annotation of the most obvious errors and some qualitative analysis of the segments themselves. From this independent analysis, we were able to conclude at a high-level that the nature of TED talk text broken into segments is highly complex, context dependent and ambiguous even in the original language which resulted in equally ambiguous translation errors. This serves as a harsh reminder of the complexity of the annotation task and that inevitably even human annotation using highly granular schemes like MQM is only as reliable as the simplicity of the underlying text. The same reminder extends to human-generated references where highly specialized content will inevitably require specialized translators to ensure the most accurate translation.

System	Unbabel MQM	Google MQM
metricsystem1	88.71 (4)	-1.61 (6)
metricsystem2	87.71 (10)	-1.68 (7)
metricsystem3	86.88 (11)	-1.41 (4)
metricsystem4	87.88 (7)	-1.77 (9)
metricsystem5	87.85 (9)	-1.74 (8)
ref-A	95.49 (1)	-0.89 (1)
Facebook-AI	91.54 (3)	-1.05 (2)
Online-W	93.27 (2)	-1.12 (3)
Nemo	88.21 (6)	-2.15 (11)
VolcTrans-GLAT	88.27 (5)	-1.49 (5)
eTranslation	87.87 (8)	-1.96 (10)

Table 15: System-level MQM scores for Unbabel and Google annotation schemes

We note that whilst we do not have human direct assessment (DA) scores on TED data in order to provide a direct comparison of the two annotation schemes in this setting, we observe in the following section that MQM appears to provide a more stable basis for evaluation in general.

8.3 Comparison to WMT Scoring

The WMT evaluation campaign (Akhbardeh et al., 2021) ran a human direct assessment (DA) evaluation for the primary submissions in the news domain for all language pairs. Segment-level ratings with document context (SR+DC) on a 0-100 scale were collected either using source-based evaluation with a mix of researchers/translators (for translations out of English) or reference-based evaluation with crowd-workers (for translations into English). In general, for each MT system, only a subset of documents receive ratings, with the rated subset differing across systems. System-level DA scores are averages over the available segment-level scores. Both raw scores and per-rater z-normalized versions of the scores are provided.

Appendix C contains correlations to WMT Newstest DA scores for all metrics, at both segment and system level, for all 16 language pairs. There is significant variation in metric performance and

ranking across languages, although a general pattern is that correlations are substantially higher for out-of-English pairs than into-English. Although the WMT correlations are not strictly comparable to the MQM results in previous sections, MQM scores tend to correlate somewhat better with metric scores for two of our three focus languages (English→German and Chinese→English), and somewhat worse for English→Russian.

System	MQM	WMT-raw	WMT-z
ref-C	-0.511 (1)	85.964 (5)	0.320 (3)
VolcTrans-GLAT	-1.039 (2)	87.265 (2)	0.301 (6)
Facebook-AI	-1.052 (3)	87.887 (1)	0.378 (2)
ref-A	-1.221 (4)	84.939 (9)	0.280 (8)
Nemo	-1.340 (5)	86.090 (4)	0.250 (10)
HuaweiTSC	-1.381 (6)	85.787 (6)	0.312 (4)
Online-W	-1.460 (7)	86.262 (3)	0.391 (1)
UEdin	-1.507 (8)	85.573 (8)	0.305 (5)
eTranslation	-1.695 (9)	85.706 (7)	0.281 (7)
VolcTrans-AT	-1.743 (10)	83.402 (10)	0.280 (9)

Table 16: MQM versus DA for English→German.

System	MQM	WMT-raw	WMT-z
ref-A	99.652 (1)	84.428 (1)	0.409 (1)
ref-B	98.397 (2)	83.492 (2)	0.386 (3)
Facebook-AI	92.749 (3)	81.541 (4)	0.338 (4)
Online-W	91.797 (4)	82.286 (3)	0.395 (2)
Online-A	87.866 (5)	76.177 (9)	0.227 (7)
Nemo	87.496 (6)	78.012 (7)	0.214 (8)
Online-G	87.225 (7)	78.466 (6)	0.242 (6)
Manifold	86.858 (8)	75.572 (10)	0.197 (9)
Online-B	85.663 (9)	79.962 (5)	0.294 (5)
NiuTrans	83.474 (10)	76.436 (8)	0.148 (10)
Online-Y	79.274 (11)	71.989 (11)	-0.015 (11)

Table 17: MQM versus DA for English→Russian.

System	MQM	WMT-raw	WMT-z
ref-A	-4.350 (1)	74.107 (3)	0.019 (3)
NiuTrans	-4.633 (2)	74.969 (2)	0.042 (1)
SMU	-4.844 (3)	70.559 (6)	-0.034 (7)
MiSS	-4.932 (4)	70.095 (9)	-0.029 (5)
Borderline	-4.945 (5)	70.486 (7)	-0.023 (4)
DIDI-NLP	-5.095 (6)	75.641 (1)	0.031 (2)
IIE-MT	-5.145 (7)	73.077 (4)	-0.031 (6)
Facebook-AI	-5.215 (8)	70.125 (8)	-0.037 (8)
Online-W	-5.567 (9)	72.851 (5)	-0.087 (9)

Table 18: MQM versus DA for Chinese→English.

Tables 16 to 18 compare MQM and DA scores for our focus language pairs, on all systems where both sets of scores were available. Notably, MQM scores rank human translations at or near the top more consistently than do DA scores. The only reference ranked worse than MT by MQM is ref-A for English→German, which as discussed above is a low-quality translation. In contrast, DA z-normalized scores rank all references below at least one MT system except for ref-A in English→Russian, which is ranked first, in agreement with MQM. For English→German and English→Russian, MQM correlates better with raw DA scores than with z-normalized scores; Pearson correlations are 0.508 versus 0.243 for the former and 0.911 versus 0.898 for the latter. For Chinese→English the pattern reverses, with correlations of 0.216 versus 0.729.

8.4 WMT DA as a Metric

Metric	news w/o HT	news w/ HT
BERTScore	0.902	0.097 (11)
cushLEPOR(LM)	0.898	0.023 (15)
TER	0.851	0.065 (14)
BLEU	0.850	0.090 (12)
MEE2	0.836	0.107 (9)
COMET-QE-MQM_2021-src	0.831	0.807 (1)
sentBLEU	0.824	0.114 (8)
bleurt-20	0.801	0.718 (3)
COMET-MQM_2021	0.790	0.697 (4)
Prism	0.778	-0.008 (17)
C-SPECpn	0.773	0.788 (2)
chrF	0.758	0.086 (13)
YiSi-1	0.735	0.102 (10)
regEMT	0.700	0.301 (6)
OpenKiwi-MQM-src	0.656	0.468 (5)
MTEQA	0.496	0.015 (16)
wmt-z	0.357	0.282 (7)
regEMT-src	-0.415	-0.311 (18)
YiSi-2-src	-0.609	-0.316 (19)

Table 19: System-level Pearson correlations, including WMT DA z-normalized scores as a metric, for English→German.

The correlations between MQM and WMT DA scores in the previous section motivated us to investigate how DA scores would fare in comparison to automatic metric scores when using MQM as gold scores. We computed system-level Pearson correlations using z-normalized DA scores for MT outputs only and MT outputs augmented with human references for which DA, MQM, and metric scores were all available.[16] Tables 19 to 21 compare these to the performance of primary and baseline metrics using the references from Table 7.[17] The performance of DA varies across languages: for English→German and English→Russian it ranks roughly among the

[16] ref-A for en→de, ref-B for en→ru, and ref-A for zh→en.

[17] These numbers do not match others in the paper due to the use of a reduced set of MT systems, and, for the *w/ HT* condition, a reduced set of human outputs.

Metric	news w/o HT	news w/ HT
OpenKiwi-MQM-src	0.973	0.815 (5)
C-SPECpn	0.967	0.934 (2)
Prism	0.966	-0.220 (14)
COMET-MQM_2021	0.964	0.866 (4)
BLEU	0.957	-0.025 (11)
COMET-QE-MQM_2021-src	0.953	0.946 (1)
sentBLEU	0.950	-0.011 (10)
bleurt-20	0.948	0.722 (6)
MEE2	0.937	-0.151 (12)
chrF	0.934	0.034 (9)
YiSi-1	0.932	0.079 (8)
BERTScore	0.926	-0.177 (13)
wmt-z	0.918	0.891 (3)
TER	0.841	-0.254 (15)
regEMT	0.803	0.370 (7)
regEMT-src	0.314	-0.612 (16)
YiSi-2-src	0.257	-0.652 (17)

Table 20: System-level Pearson correlations, including WMT DA z-normalized scores as a metric, for English→Russian.

Metric	news w/o HT	news w/ HT
C-SPECpn	0.797	0.882 (1)
regEMT	0.764	0.477 (5)
wmt-z	0.724	0.729 (3)
RoBLEURT	0.722	-0.237 (9)
COMET-MQM_2021	0.683	-0.034 (6)
BERTScore	0.673	-0.224 (8)
bleurt-20	0.656	-0.090 (7)
YiSi-1	0.649	-0.244 (10)
OpenKiwi-MQM-src	0.623	0.604 (4)
Prism	0.596	-0.371 (11)
COMET-QE-MQM_2021-src	0.586	0.748 (2)
chrF	0.573	-0.438 (14)
YiSi-2-src	0.519	-0.431 (13)
MEE2	0.515	-0.438 (15)
BLEU	0.507	-0.472 (16)
MTEQA	0.469	-0.424 (12)
cushLEPOR(LM)	0.460	-0.490 (18)
sentBLEU	0.441	-0.477 (17)
TER	0.316	-0.495 (19)
regEMT-src	0.004	-0.607 (20)

Table 21: System-level Pearson correlations, including WMT DA z-normalized scores as a metric, for Chinese→English.

bottom half of the automatic metrics; while for Chinese→English it ranks third. DA scores tend to perform better when judging human output, ranking 7th, 3rd, and 3rd for English→German, English→Russian, and Chinese→English, respectively.

9 Conclusion

This paper summarized the results of the WMT21 shared task on automated machine translation evaluation, the Metrics Shared Task. This year, we collected our own human ratings for selected language pairs (En→De, En→Ru, and Zh→En) from professional translators via MQM to generate a reliable ground truth across two domains. WMT direct assessment (DA) scores generally correlate poorly with MQM scores, and exhibit weaker preference for human translations compared to machine output. For En→De and Zh→En, DA ranks the human translations below many MT systems, demonstrating that expert-based evaluation is needed to generate a reliable ground truth for the Metrics Shared Task. The majority of metrics correlate better with MQM than with WMT DA, confirming previous findings that the best automatic metrics are already more reliable than crowd worker human evaluations. The performance of each metric varies depending on the underlying domain (being either TED talks or news) demonstrating that most metrics are not domain robust. Further, the challenge sets revealed that most metrics struggle to penalize translations with errors in reversing negation or sentiment polarity, and show lower correlations for Subordination, Named Entities and Terminology.

Overall, metrics perform very differently based on domain, language pair or setting (with or without human translations among candidate systems) making it hard to declare a clear winner. When counting top performances across all test conditions, three embedding-based metrics—C-SPECPN, BLEURT-20, and COMET-MQM_2021— emerge as distinctly better than the others, especially at the segment level and when rating human translations. Nevertheless, it is unclear which test scenario and correlation metric is best to yield reliable results. We would encourage the community to investigate different ways of how to evaluate automatic metrics. We are very open to apply new suggestions in the next round of the Metrics Shared Task.

Another challenge is to define the overall ground truth (i.e. the human ratings). Even though, we are convinced that expert-based ratings via MQM are more reliable, we also found that the two MQM methodologies of Unbabel and Google disagree for some systems. We would encourage the community to further work on establishing a reliable human evaluation setup. The field would benefit from a reliable human evaluation standard that could be used by everyone.

10 Ethical considerations

MQM annotations and additional reference translations in this paper are done by professional translators. They are all paid at professional rates.

Our TED talks test suite is created based on TED transcripts and translations under CC BY–NC–ND 4.0 International. Additional translations done for this shared task follow the TED Talks Usage Policy.

Organizers from the National Research Council Canada and Unbabel have submitted to this task the frozen stable versions of their metrics (YiSi and COMET) dated before they joined the organizing committee. Newer versions of COMET were developed without using any of the test set, test suite or challenge sets.

11 Acknowledgments

Results for this shared task would not be possible without tight collaboration with the organizers of the WMT News Translation Task. We are grateful to Google and Unbabel for sponsoring and overseeing the human evaluation. We would also like to thank the organizers of the WMT20 Robustness Task, especially Lucia Specia, for providing the pre-release version of the MT critical error annotations. We are grateful to Roman Grundkiewicz for his help in deciphering raw scores exported from Appraise and André Martins, Thibault Sellam, Qijun Tan and Macduff Hughes for their valuable review. Finally, we would like to thank Eleftherios Avramidis and Vivien Macketanz for providing us the German→English Challenge set (Section 6.3).

Nitika Mathur is supported by the Australian Research Council. Ricardo Rei is supported by the P2020 programs MAIA (contract 045909) and Unbabel4EU (contract 042671). Ondřej Bojar would like to acknowledge the support from the Czech Science Foundation (grant n. 19-26934X, NEUREM3).

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, and Marta R. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, Aljoscha Burchardt, and Sebastian Möller. 2020. Fine-grained linguistic evaluation for state-of-the-art machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 346–356, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic evaluation of German-English machine translation using a test suite. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 445–454, Florence, Italy. Association for Computational Linguistics.

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, Copenhagen, Denmark. 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*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *arXiv preprint arXiv:2104.00054*.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *CoRR*, abs/2104.14478.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Lifeng Han, Irina Sorokina, Gleb Erofeev, and Serge Gladkoff. 2021. cushlepor: Customised hlepor metric using labse distilled knowledge model to improve agreement with human judgements. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Lifeng Han, Derek F. Wong, Lidia S. Chao, Liangye He, Yi Lu, Junwen Xing, and Xiaodong Zeng. 2013. Language-independent model for machine translation evaluation with reinforced factors. In *Machine Translation Summit XIV*, pages 215–222. International Association for Machine Translation.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens, and Pavel Pecina. 2021a. Just ask! evaluating machine translation by asking and answering questions. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens, and Pavel Pecina. 2021b. Mteqa at wmt21 metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo. 2020. Extended study on using pretrained language models and YiSi-1 for machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 895–902, Online. Association for Computational Linguistics.

Chi-kiu Lo and Samuel Larkin. 2020. Machine translation reference-less evaluation using YiSi-2 with bilingual mappings of massive multilingual language model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 903–910, Online. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional Quality Metrics (MQM) : A Framework for Declaring and Describing Translation Quality Metrics. *Tradumàtica*, pages 0455–463.

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: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Vivien Macketanz, Renlong Ai, Aljoscha Burchardt, and Hans Uszkoreit. 2018a. TQ-AutoTest – an automated test suite for (machine) translation quality. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018b. Fine-grained evaluation of German-English machine translation based on a test suite. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 578–587, Belgium, Brussels. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova, and Sebastian Möller. 2021. Linguistic evaluation for the 2021 state-of-the-art machine translation systems for german to english and english to german. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Ananya Mukherjee, Hema Ala, Manish Shrivastava, and Dipti Misra Sharma. 2020. Mee : An automatic metric for evaluation using embeddings for machine translation. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 292–299.

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.

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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro G Ramos, Taisiya Glushkova, André Martins, and Alon Lavie. 2021. Are References Really Needed?Unbabel-IST 2021 Submission for the Metrics Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich 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: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020a. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Zhenhao Li, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel, and Xian Li. 2020b. Findings of the WMT 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation*, pages 76–91, Online. Association for Computational Linguistics.

Michal Stefanik, Vít Novotný, and Petr Sojka. 2021. Regressive ensemble for machine translation quality evaluation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Kosuke Takahashi, Yoichi Ishibashi, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Multilingual machine translation evaluation metrics fine-tuned on pseudo-negative examples for wmt 2021 metrics task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Tianchi Bi, Haibo Zhang, Boxing Chen, Weihua Luo, Derek F. Wong, and Lidia S. Chao. 2021. Robleurt submission for wmt2021 metrics task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Evan James Williams. 1959. *Regression Analysis*, volume 14. wiley.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Challenge Set Perturbation Examples

Perturbation	Description	Example
Antonym	Randomly replaces as word with it's antonym.	Orig.: *Fire in French chemical plant extinguished* New: *Fire in French chemical plant ignite*
Word omission	Randomly drops a word from a sentence	Orig.: *Fire in French chemical plant extinguished* New: *Fire in French␣ plant extinguished*
Tokenized	Applies tokenization to the sentence.	Orig.: *Spain: It is safe here.* New: *Spain␣: It is safe here␣.*
Sentence Omission	Removes a sentence from a paragraph.	Orig.: *The No.3 flood of the Yangtze River in 2020 was formed. The Ministry of Water Resources has refined and implemented countermeasures - www.chinanews.com* New: *The No.3 flood of the Yangtze River in 2020 was formed. ␣*
Punctuation	Removes punctuation from the input.	Orig.: *Spain: It is safe here.* New: *Spain␣ It is safe here␣.*
Numbers	Replaces a number by another randomly generated number.	Orig.: *Around 65 people work at the plant.* New: *Around 400 people work at the plant.*
Lowercasing	Applies lowercasing to the entire input.	Orig.: *Fire in French chemical plant extinguished* New: *fire in french chemical plant extinguished*
Word Addition	Adds a word in the middle of a sentence using `distilbert-base-uncased`. This perturbation is applied on top of lowercase perturbation.	Orig.: *fire in french chemical plant extinguished* New: *fire in french underground chemical plant extinguished*
Spelling	Adds spelling errors to the input.	Orig.: *Fire in French chemical plant extinguished* New: *Fire in French chemical pants extinguished*

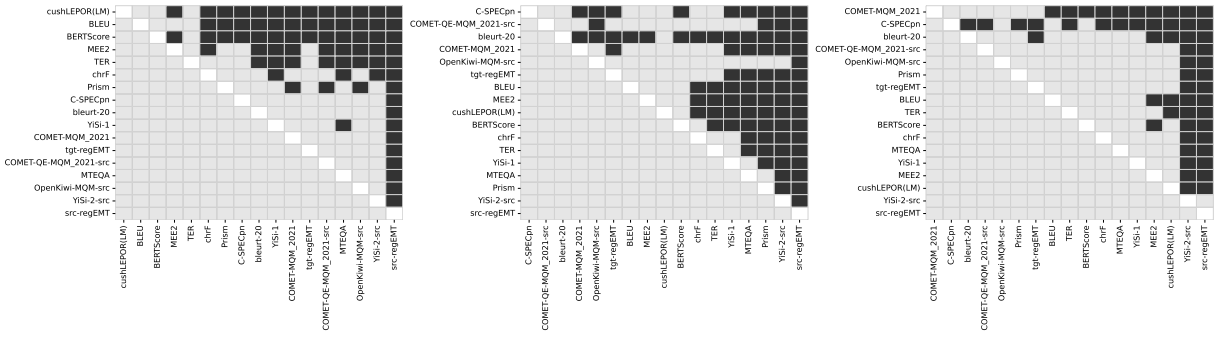
Table 22: List of all perturbations used to construct the Challenge Set described in Section 6.2. The right column provides for each perturbation an example with the original sentence and the corresponding new corrupted sentence.

B Language-Specific Results Tables

Language-specific results are given on the following pages. Each page contains results for a single language pair and granularity (system or segment). Correlation results in tables are followed by pairwise significance plots for each condition (news without human outputs, news with human outputs, TED talks) considering only primary and baseline metrics.

Metric	news w/o HT	news w/ HT	TED
cushLEPOR(LM)	**0.938** (1)	0.085 (17)	0.239 (23)
BLEU	**0.937** (2)	0.132 (13)	0.620 (13)
BERTScore	**0.930** (3)	0.074 (19)	0.506 (17)
cushLEPOR(pSQM)	0.921 (4)	0.085 (18)	0.067 (25)
MEE	0.916 (5)	0.109 (14)	0.449 (19)
MEE2	0.900 (6)	0.098 (15)	0.392 (22)
TER	0.898 (7)	0.003 (22)	0.609 (14)
hLEPOR	0.896 (8)	0.094 (16)	0.127 (24)
COMET-QE-DA_2021-src	0.847 (9)	0.807 (3)	0.527 (16)
chrF	0.846 (10)	0.017 (21)	0.471 (18)
Prism	0.841 (11)	-0.123 (26)	0.659 (11)
COMET-DA_2020	0.814 (12)	0.658 (8)	0.788 (4)
COMET-DA_2021	0.812 (13)	0.607 (9)	0.780 (5)
C-SPECpn	0.804 (14)	**0.823** (1)	**0.802** (2)
bleurt-20	0.802 (15)	**0.774** (5)	0.739 (6)
YiSi-1	0.789 (16)	-0.009 (23)	0.414 (21)
C-SPEC	0.777 (17)	0.822 (2)	0.788 (3)
COMET-MQM_2021	0.771 (18)	0.720 (7)	**0.818** (1)
bleurt-21-beta	0.771 (19)	0.758 (6)	0.695 (7)
COMETinho-DA	0.768 (20)	0.054 (20)	0.548 (15)
tgt-regEMT	0.742 (21)	0.411 (11)	0.641 (12)
COMET-QE-MQM_2021-src	0.711 (22)	**0.792** (4)	0.694 (8)
MTEQA	0.658 (23)	-0.116 (25)	0.418 (20)
tgt-regEMT-baseline	0.653 (24)	0.148 (12)	-0.078 (26)
COMETinho-MQM	0.557 (25)	-0.034 (24)	0.663 (10)
OpenKiwi-MQM-src	0.494 (26)	0.439 (10)	0.669 (9)
YiSi-2-src	0.283 (27)	-0.416 (28)	-0.419 (28)
src-regEMT-baseline	-0.173 (28)	-0.224 (27)	-0.133 (27)
src-regEMT	-0.606 (29)	-0.558 (29)	-0.699 (29)

Table 23: System-level Pearson correlations for English→German. Primary submissions are bolded, and baselines are underlined. Correlations for metrics in the top cluster (considering only primary and baseline metrics) are bolded.

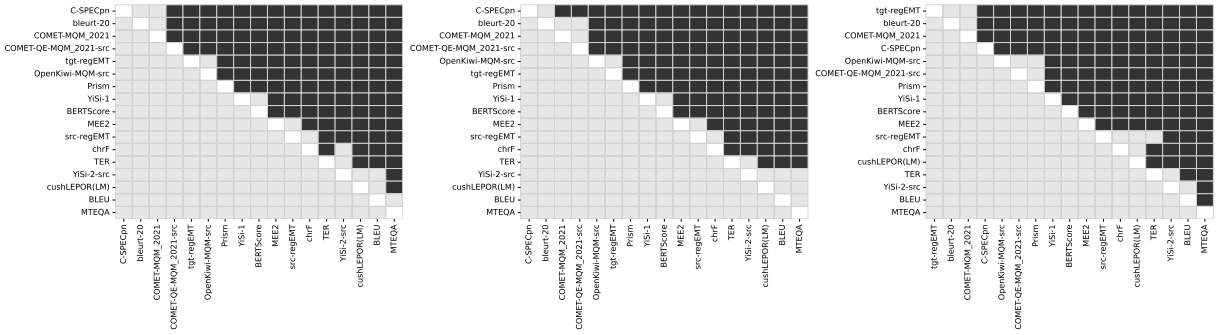


(a) news w/o HT (b) news w/ HT (c) TED w/o HT

Figure 2: System-level Pearson pairwise significance for English→German primary submissions and baselines. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

Metric	news w/o HT	news w/ HT	TED
C-SPECpn	**0.267** (1)	**0.254** (2)	0.270 (5)
C-SPEC	0.266 (2)	0.256 (1)	0.285 (2)
bleurt-20	**0.264** (3)	**0.247** (3)	**0.283** (3)
COMET-MQM_2021	**0.263** (4)	0.241 (4)	**0.282** (4)
COMET-DA_2021	0.253 (5)	0.226 (8)	0.267 (6)
bleurt-21-beta	0.252 (6)	0.238 (5)	0.252 (10)
COMET-QE-MQM_2021-src	0.248 (7)	0.235 (6)	0.253 (9)
COMET-QE-DA_2021-src	0.244 (8)	0.227 (7)	0.221 (14)
COMET-DA_2020	0.239 (9)	0.212 (11)	0.259 (7)
tgt-regEMT	0.234 (10)	0.214 (10)	**0.290** (1)
OpenKiwi-MQM-src	0.232 (11)	0.219 (9)	0.255 (8)
COMETinho-MQM	0.202 (12)	0.186 (12)	0.245 (11)
COMETinho-DA	0.198 (13)	0.172 (13)	0.236 (13)
Prism	0.192 (14)	0.164 (14)	0.238 (12)
YiSi-1	0.172 (15)	0.145 (15)	0.212 (15)
BERTScore	0.169 (16)	0.143 (16)	0.189 (16)
MEE2	0.142 (17)	0.117 (17)	0.173 (17)
src-regEMT	0.128 (18)	0.106 (18)	0.149 (19)
MEE	0.126 (19)	0.105 (19)	0.142 (22)
chrF	0.114 (20)	0.090 (20)	0.146 (20)
TER	0.098 (21)	0.078 (22)	0.131 (23)
YiSi-2-src	0.098 (22)	0.071 (23)	0.119 (25)
cushLEPOR(LM)	0.090 (23)	0.068 (24)	0.144 (21)
tgt-regEMT-baseline	0.084 (24)	0.080 (21)	0.161 (18)
sentBLEU	0.083 (25)	0.064 (26)	0.113 (27)
cushLEPOR(pSQM)	0.078 (26)	0.057 (28)	0.127 (24)
MTEQA	0.071 (27)	0.060 (27)	0.082 (29)
hLEPOR	0.071 (28)	0.050 (29)	0.117 (26)
src-regEMT-baseline	0.067 (29)	0.067 (25)	0.112 (28)

Table 24: Segment-level Kendall correlations for English→German. Primary submissions are bolded, and baselines are underlined. Correlations for metrics in the top cluster (considering only primary and baseline metrics) are bolded.

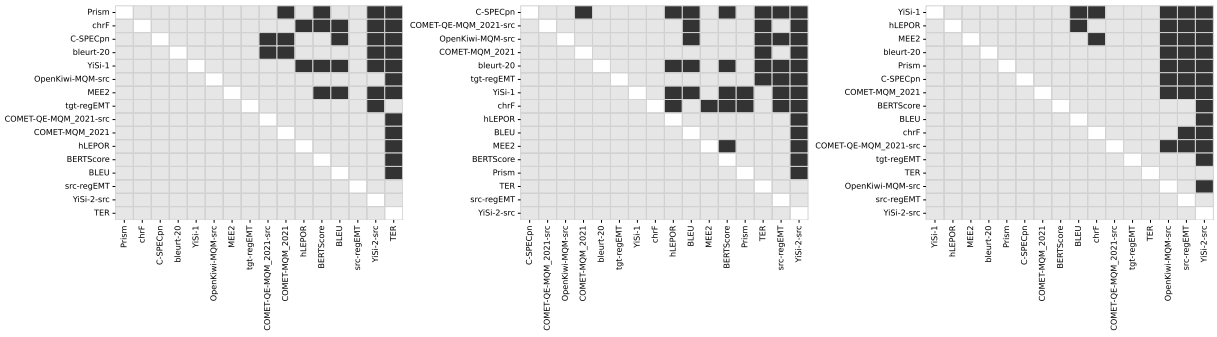


(a) news w/o HT (b) news w/ HT (c) TED w/o HT

Figure 3: Segment-level Kendall significance for English→German primary submissions and baselines. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

Metric	news w/o HT	news w/ HT	TED
Prism	**0.799** (1)	-0.136 (21)	**0.867** (8)
chrF	**0.783** (2)	0.123 (14)	0.825 (16)
C-SPECpn	**0.782** (3)	**0.824** (1)	**0.855** (12)
bleurt-20	**0.768** (4)	0.653 (8)	**0.868** (7)
C-SPEC	0.763 (5)	0.817 (2)	0.858 (10)
YiSi-1	**0.761** (6)	0.138 (13)	**0.905** (1)
MEE	0.759 (7)	0.051 (15)	0.881 (5)
OpenKiwi-MQM-src	**0.755** (8)	**0.729** (4)	0.691 (21)
MEE2	**0.750** (9)	-0.069 (18)	**0.882** (4)
bleurt-21-beta	0.743 (10)	0.692 (5)	0.856 (11)
tgt-regEMT	**0.740** (11)	0.390 (11)	0.758 (19)
COMET-QE-MQM_2021-src	0.688 (12)	**0.784** (3)	0.817 (17)
COMET-DA_2020	0.676 (13)	0.556 (10)	0.859 (9)
COMET-MQM_2021	0.659 (14)	0.685 (6)	**0.841** (13)
COMET-DA_2021	0.655 (15)	0.645 (9)	0.871 (6)
hLEPOR	0.648 (16)	-0.038 (16)	**0.894** (2)
COMET-QE-DA_2021-src	0.632 (17)	0.681 (7)	0.884 (3)
BERTScore	0.629 (18)	-0.123 (20)	**0.831** (14)
COMETinho-DA	0.578 (19)	0.239 (12)	0.758 (18)
BLEU	0.507 (20)	-0.043 (17)	0.828 (15)
src-regEMT	0.301 (21)	-0.436 (24)	0.115 (24)
tgt-regEMT-baseline	0.186 (22)	-0.413 (23)	0.121 (23)
COMETinho-MQM	0.089 (23)	-0.083 (19)	0.432 (22)
YiSi-2-src	0.046 (24)	-0.585 (26)	0.085 (25)
TER	-0.041 (25)	-0.289 (22)	0.697 (20)
src-regEMT-baseline	-0.585 (26)	-0.583 (25)	-0.228 (26)

Table 25: System-level Pearson correlations for English→Russian. Primary submissions are bolded, and baselines are underlined. Correlations for metrics in the top cluster are bolded.

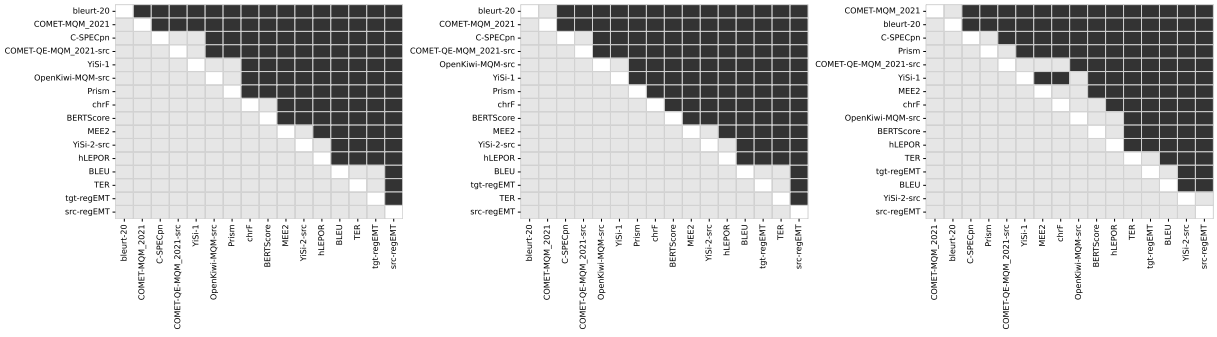


(a) news w/o HT (b) news w/ HT (c) TED w/o HT

Figure 4: System-level Pearson pairwise significance for English→Russian primary submissions and baselines. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

Metric	news w/o HT	news w/ HT	TED
COMET-DA_2021	0.307 (1)	0.296 (1)	0.274 (1)
bleurt-20	**0.286** (2)	**0.276** (3)	**0.255** (5)
COMET-QE-DA_2021-src	0.284 (3)	0.278 (2)	0.245 (6)
bleurt-21-beta	0.278 (4)	0.271 (4)	0.269 (2)
COMET-DA_2020	0.278 (5)	0.265 (6)	0.242 (7)
COMET-MQM_2021	0.276 (6)	**0.268** (5)	**0.258** (4)
C-SPEC	0.259 (7)	0.259 (7)	0.263 (3)
C-SPECpn	0.248 (8)	0.248 (8)	0.233 (8)
COMETinho-DA	0.248 (9)	0.233 (10)	0.218 (10)
COMET-QE-MQM_2021-src	0.242 (10)	0.239 (9)	0.204 (12)
YiSi-1	0.233 (11)	0.216 (12)	0.204 (11)
OpenKiwi-MQM-src	0.225 (12)	0.222 (11)	0.187 (15)
Prism	0.224 (13)	0.205 (13)	0.219 (9)
COMETinho-MQM	0.197 (14)	0.188 (14)	0.182 (17)
chrF	0.193 (15)	0.178 (15)	0.189 (14)
BERTScore	0.185 (16)	0.168 (16)	0.185 (16)
MEE2	0.169 (17)	0.153 (17)	0.193 (13)
YiSi-2-src	0.163 (18)	0.140 (18)	0.084 (23)
MEE	0.150 (19)	0.135 (20)	0.176 (19)
hLEPOR	0.150 (20)	0.135 (19)	0.178 (18)
sentBLEU	0.120 (21)	0.106 (21)	0.112 (22)
TER	0.117 (22)	0.104 (23)	0.142 (20)
tgt-regEMT	0.110 (23)	0.105 (22)	0.129 (21)
src-regEMT	0.085 (24)	0.070 (24)	0.070 (24)
tgt-regEMT-baseline	0.053 (25)	0.050 (25)	0.053 (25)
src-regEMT-baseline	-0.045 (26)	-0.043 (26)	0.018 (26)

Table 26: Segment-level Kendall correlations for English→Russian. Primary submissions are bolded, and baselines are underlined. Correlations for metrics in the top cluster (considering only primary and baseline metrics) are bolded.

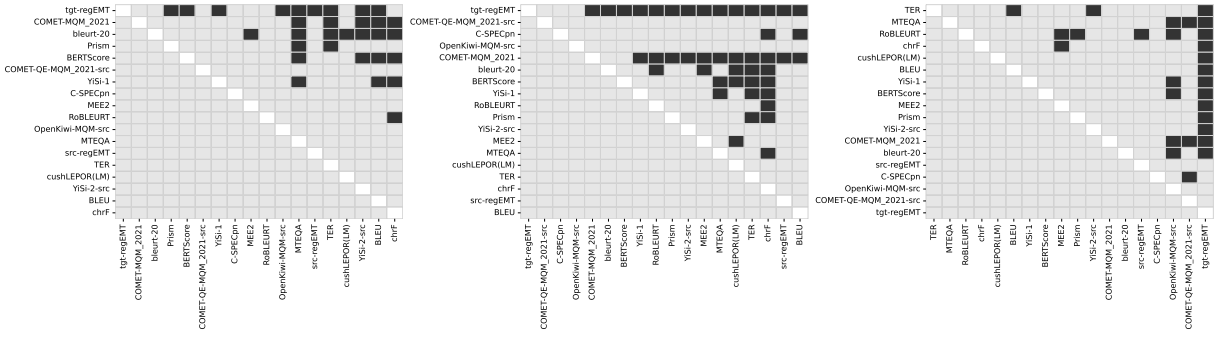


(a) news w/o HT (b) news w/ HT (c) TED w/o HT

Figure 5: Segment-level Kendall significance for English→Russian primary submissions and baselines. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

Metric	news w/o HT	news w/ HT	TED
tgt-regEMT	**0.834** (1)	**0.727** (1)	-0.404 (30)
COMET-MQM_2021	**0.628** (2)	0.336 (7)	0.266 (18)
bleurt-20	**0.563** (3)	0.294 (9)	0.239 (20)
Prism	0.558 (4)	0.031 (17)	0.272 (15)
BERTScore	0.542 (5)	0.095 (14)	0.306 (12)
bleurt-21-beta	0.537 (6)	0.265 (10)	0.235 (21)
COMETinho-MQM	0.530 (7)	0.129 (13)	0.395 (5)
COMET-QE-MQM_2021-src	0.529 (8)	**0.619** (2)	-0.209 (29)
C-SPEC	0.526 (9)	0.619 (3)	-0.064 (26)
COMET-DA_2021	0.516 (10)	0.186 (12)	0.306 (11)
YiSi-1	0.515 (11)	0.077 (15)	0.310 (10)
COMET-DA_2020	0.511 (12)	0.221 (11)	0.251 (19)
hLEPOR	0.498 (13)	-0.061 (24)	0.372 (6)
C-SPECpn	0.492 (14)	**0.594** (4)	-0.053 (25)
MEE2	0.453 (15)	-0.011 (19)	0.289 (14)
COMET-QE-DA_2021-src	0.453 (16)	0.535 (5)	0.057 (24)
RoBLEURT	0.451 (17)	0.065 (16)	**0.400** (3)
OpenKiwi-MQM-src	0.445 (18)	**0.489** (6)	-0.077 (27)
MTEQA	0.423 (19)	-0.050 (21)	**0.403** (2)
src-regEMT	0.419 (20)	-0.149 (29)	0.077 (23)
TER	0.416 (21)	-0.085 (26)	**0.421** (1)
cushLEPOR(LM)	0.412 (22)	-0.052 (22)	**0.327** (8)
YiSi-2-src	0.411 (23)	0.013 (18)	0.270 (16)
COMETinho-DA	0.340 (24)	-0.019 (20)	0.397 (4)
MEE	0.324 (25)	-0.125 (27)	0.301 (13)
src-regEMT-baseline	0.310 (26)	0.300 (8)	-0.105 (28)
BLEU	0.310 (27)	-0.152 (30)	0.324 (9)
chrF	0.302 (28)	-0.143 (28)	**0.363** (7)
cushLEPOR(pSQM)	0.237 (29)	-0.058 (23)	0.267 (17)
tgt-regEMT-baseline	0.089 (30)	-0.075 (25)	0.201 (22)

Table 27: System-level Pearson correlations for Chinese→English. Primary submissions are bolded, and baselines are underlined. Correlations for metrics in the top cluster are bolded.

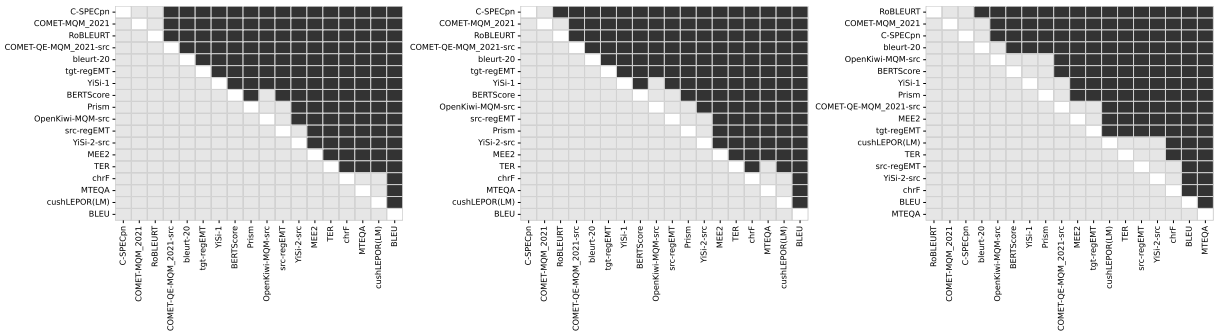


(a) news w/o HT (b) news w/ HT (c) TED w/o HT

Figure 6: System-level Pearson pairwise significance for Chinese→English primary submissions and baselines. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

Metric	news w/o HT	news w/ HT	TED
C-SPECpn	**0.402** (1)	**0.390** (1)	**0.233** (7)
C-SPEC	0.401 (2)	0.388 (2)	0.241 (2)
COMET-MQM_2021	**0.395** (3)	**0.384** (3)	**0.233** (5)
RoBLEURT	**0.394** (4)	0.380 (4)	**0.238** (4)
COMET-DA_2021	0.371 (5)	0.357 (7)	0.219 (11)
COMETinho-MQM	0.370 (6)	0.362 (5)	0.239 (3)
COMET-QE-MQM_2021-src	0.367 (7)	0.358 (6)	0.178 (17)
COMET-DA_2020	0.360 (8)	0.347 (8)	0.220 (10)
bleurt-21-beta	0.357 (9)	0.344 (9)	0.233 (6)
bleurt-20	0.354 (10)	0.341 (10)	0.224 (9)
COMETinho-DA	0.339 (11)	0.327 (11)	0.199 (14)
tgt-regEMT	0.328 (12)	0.318 (12)	0.173 (18)
COMET-QE-DA_2021-src	0.305 (13)	0.294 (13)	0.122 (28)
YiSi-1	0.302 (14)	0.289 (14)	0.195 (15)
BERTScore	0.296 (15)	0.281 (15)	0.199 (13)
Prism	0.285 (16)	0.270 (19)	0.194 (16)
OpenKiwi-MQM-src	0.283 (17)	0.277 (16)	0.213 (12)
src-regEMT	0.280 (18)	0.274 (17)	0.135 (23)
tgt-regEMT-baseline	0.278 (19)	0.272 (18)	0.248 (1)
YiSi-2-src	0.270 (20)	0.263 (20)	0.125 (26)
src-regEMT-baseline	0.255 (21)	0.251 (21)	0.231 (8)
MEE2	0.247 (22)	0.233 (22)	0.173 (19)
TER	0.210 (23)	0.198 (23)	0.136 (22)
hLEPOR	0.205 (24)	0.193 (24)	0.129 (25)
chrF	0.201 (25)	0.188 (25)	0.124 (27)
MEE	0.196 (26)	0.186 (27)	0.131 (24)
MTEQA	0.194 (27)	0.187 (26)	0.028 (30)
cushLEPOR(LM)	0.193 (28)	0.182 (28)	0.138 (21)
sentBLEU	0.176 (29)	0.165 (29)	0.092 (29)
cushLEPOR(pSQM)	0.167 (30)	0.158 (30)	0.143 (20)

Table 28: Segment-level Kendall correlations for Chinese→English. Primary submissions are bolded, and baselines are underlined. Correlations for metrics in the top cluster are bolded.



(a) news w/o HT (b) news w/ HT (c) TED w/o HT

Figure 7: Segment-level Kendall significance for Chinese→English primary submissions and baselines. Dark squares mean the row metric correlates significantly better than the column metric at $\alpha = 0.05$.

C WMT Direct Assessment Results

Correlations with WMT Direct Assessment scores for the news and FLORES test sets are given in the following tables, with results for news to-English language pairs followed by to-non-English pairs, followed by FLORES. Since most language pairs contained only a single reference, we used reference A for all pairs, and report results only for scoring MT output (omitting additional scored references for language pairs where these were available). System-level correlations use Pearson over z-normalized rater scores. Segment-level correlations use the traditional Kendall-like formula over raw rater scores, discarding segment pairs whose scores differ by less than 25.[18]

metric	correlation
regEMT-src	0.778
COMETinho-MQM	0.652
Prism	0.651
RoBLEURT	0.648
OpenKiwi-MQM-src	0.641
COMET-MQM_2021	0.638
COMET-DA_2020	0.632
BERTScore	0.629
bleurt-21-beta	0.628
COMET-DA_2021	0.626
COMET-QE-MQM_2021-src	0.625
C-SPEC	0.623
bleurt-20	0.620
regEMT	0.609
YiSi-1	0.607
COMET-QE-DA_2021-src	0.606
C-SPECpn	0.590
COMETinho-DA	0.588
MTEQA	0.586
chrF	0.562
sentBLEU	0.550
TER	0.509
hLEPOR	0.496
YiSi-2-src	0.248
regEMT-baseline	-0.195
regEMT-baseline-src	-0.335

metric	correlation
RoBLEURT	0.044
COMET-MQM_2021	0.037
COMETinho-MQM	0.034
COMET-QE-MQM_2021-src	0.033
COMET-DA_2021	0.032
COMET-DA_2020	0.032
regEMT	0.027
COMET-QE-DA_2021-src	0.026
OpenKiwi-MQM-src	0.018
YiSi-2-src	0.017
COMETinho-DA	0.015
C-SPECpn	0.008
regEMT-src	0.003
Prism	-0.002
C-SPEC	-0.012
bleurt-20	-0.017
BERTScore	-0.019
bleurt-21-beta	-0.026
YiSi-1	-0.039
chrF	-0.053
sentBLEU	-0.088
hLEPOR	-0.098
regEMT-baseline	-0.118
regEMT-baseline-src	-0.135
TER	-0.226
MTEQA	-0.237

Table 29: Correlations for Czech→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

[18] Note that we used average sentence-level BLEU rather than corpus BLEU for system-level results, in contrast to our main results.

metric	correlation
regEMT-baseline	0.520
hLEPOR	0.493
YiSi-1	0.395
MTEQA	0.394
regEMT	0.362
COMET-DA_2020	0.361
chrF	0.357
YiSi-2-src	0.354
COMET-DA_2021	0.354
RoBLEURT	0.353
Prism	0.349
COMET-MQM_2021	0.346
bleurt-20	0.340
BERTScore	0.336
COMETinho-DA	0.333
bleurt-21-beta	0.325
COMET-QE-DA_2021-src	0.320
COMET-QE-MQM_2021-src	0.293
sentBLEU	0.231
OpenKiwi-MQM-src	0.215
COMETinho-MQM	0.163
C-SPECpn	0.122
C-SPEC	0.090
TER	0.070
regEMT-src	0.064
regEMT-baseline-src	-0.499

metric	correlation
RoBLEURT	0.011
COMET-QE-MQM_2021-src	0.004
COMETinho-DA	0.001
COMET-DA_2020	-0.002
YiSi-2-src	-0.003
COMET-QE-DA_2021-src	-0.003
COMET-MQM_2021	-0.003
COMETinho-MQM	-0.005
COMET-DA_2021	-0.006
OpenKiwi-MQM-src	-0.020
regEMT	-0.025
regEMT-src	-0.034
Prism	-0.037
C-SPECpn	-0.091
C-SPEC	-0.093
BERTScore	-0.098
bleurt-20	-0.146
YiSi-1	-0.151
bleurt-21-beta	-0.153
chrF	-0.162
hLEPOR	-0.209
sentBLEU	-0.215
regEMT-baseline	-0.231
regEMT-baseline-src	-0.234
TER	-0.340
MTEQA	-0.413

Table 30: Correlations for German→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-20	0.955
COMET-DA_2020	0.949
Prism	0.948
bleurt-21-beta	0.947
BERTScore	0.947
YiSi-1	0.944
RoBLEURT	0.944
regEMT	0.940
COMET-DA_2021	0.939
sentBLEU	0.936
chrF	0.924
COMETinho-DA	0.923
MTEQA	0.909
COMET-MQM_2021	0.902
COMET-QE-DA_2021-src	0.898
COMETinho-MQM	0.880
TER	0.823
C-SPEC	0.810
OpenKiwi-MQM-src	0.806
YiSi-2-src	0.795
COMET-QE-MQM_2021-src	0.782
C-SPECpn	0.720
regEMT-baseline	0.525
regEMT-src	0.363
regEMT-baseline-src	0.014

metric	correlation
COMET-MQM_2021	0.076
RoBLEURT	0.075
COMET-DA_2021	0.072
C-SPEC	0.070
Prism	0.070
C-SPECpn	0.066
COMET-QE-DA_2021-src	0.064
COMET-DA_2020	0.062
BERTScore	0.062
COMETinho-DA	0.056
OpenKiwi-MQM-src	0.051
YiSi-1	0.049
COMET-QE-MQM_2021-src	0.047
bleurt-20	0.046
YiSi-2-src	0.046
regEMT	0.043
bleurt-21-beta	0.039
COMETinho-MQM	0.036
chrF	0.021
regEMT-src	0.009
sentBLEU	-0.010
regEMT-baseline	-0.067
regEMT-baseline-src	-0.067
MTEQA	-0.067
TER	-0.125

Table 31: Correlations for Hausa→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
RoBLEURT	0.891
bleurt-21-beta	0.889
bleurt-20	0.888
COMET-QE-MQM_2021-src	0.887
OpenKiwi-MQM-src	0.879
COMET-MQM_2021	0.872
TER	0.869
YiSi-1	0.868
BERTScore	0.867
COMET-DA_2021	0.866
sentBLEU	0.858
COMET-QE-DA_2021-src	0.857
regEMT	0.856
chrF	0.854
COMET-DA_2020	0.849
Prism	0.846
MTEQA	0.831
COMETinho-DA	0.818
COMETinho-MQM	0.800
regEMT-src	0.665
regEMT-baseline-src	0.632
YiSi-2-src	0.628
C-SPECpn	0.622
regEMT-baseline	0.445
C-SPEC	-0.104

metric	correlation
COMET-MQM_2021	0.069
Prism	0.063
RoBLEURT	0.063
COMET-DA_2021	0.061
COMET-QE-MQM_2021-src	0.061
COMET-DA_2020	0.057
C-SPEC	0.057
COMETinho-DA	0.055
COMET-QE-DA_2021-src	0.051
COMETinho-MQM	0.048
regEMT	0.041
C-SPECpn	0.041
BERTScore	0.038
YiSi-2-src	0.035
OpenKiwi-MQM-src	0.031
bleurt-20	0.030
bleurt-21-beta	0.028
regEMT-src	0.027
YiSi-1	0.023
chrF	0.018
sentBLEU	-0.018
regEMT-baseline	-0.063
regEMT-baseline-src	-0.083
TER	-0.126
MTEQA	-0.157

Table 32: Correlations for Icelandic→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
COMET-DA_2020	0.846
COMETinho-DA	0.839
COMET-DA_2021	0.832
chrF	0.831
Prism	0.827
COMETinho-MQM	0.824
YiSi-1	0.821
RoBLEURT	0.820
BERTScore	0.819
bleurt-20	0.806
bleurt-21-beta	0.803
sentBLEU	0.787
MTEQA	0.784
COMET-MQM_2021	0.766
COMET-QE-DA_2021-src	0.759
regEMT	0.739
regEMT-baseline	0.716
YiSi-2-src	0.696
TER	0.693
OpenKiwi-MQM-src	0.584
COMET-QE-MQM_2021-src	0.567
C-SPEC	0.365
regEMT-src	0.071
C-SPECpn	-0.074
regEMT-baseline-src	-0.710

metric	correlation
RoBLEURT	0.045
Prism	0.035
COMET-DA_2020	0.033
COMET-MQM_2021	0.032
COMET-DA_2021	0.031
C-SPEC	0.030
BERTScore	0.028
COMETinho-DA	0.025
COMET-QE-DA_2021-src	0.025
C-SPECpn	0.024
YiSi-1	0.022
OpenKiwi-MQM-src	0.021
COMET-QE-MQM_2021-src	0.012
regEMT	0.009
YiSi-2-src	0.009
bleurt-20	0.007
chrF	0.005
COMETinho-MQM	0.002
bleurt-21-beta	0.002
regEMT-src	-0.004
sentBLEU	-0.023
regEMT-baseline	-0.054
regEMT-baseline-src	-0.070
MTEQA	-0.082
TER	-0.129

Table 33: Correlations for Japanese→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
COMET-QE-DA_2021-src	0.764
OpenKiwi-MQM-src	0.759
regEMT	0.748
COMET-QE-MQM_2021-src	0.742
bleurt-21-beta	0.732
COMET-MQM_2021	0.728
bleurt-20	0.728
COMET-DA_2020	0.726
COMET-DA_2021	0.711
MTEQA	0.705
RoBLEURT	0.687
regEMT-baseline	0.676
BERTScore	0.668
COMETinho-MQM	0.658
Prism	0.657
YiSi-1	0.652
COMETinho-DA	0.627
chrF	0.593
hLEPOR	0.527
sentBLEU	0.512
TER	0.481
C-SPEC	0.456
C-SPECpn	0.394
YiSi-2-src	0.335
regEMT-src	0.092
regEMT-baseline-src	-0.535

metric	correlation
OpenKiwi-MQM-src	0.024
COMET-QE-MQM_2021-src	0.018
regEMT	0.017
COMET-QE-DA_2021-src	0.007
COMET-MQM_2021	0.005
COMET-DA_2021	-0.006
RoBLEURT	-0.006
C-SPECpn	-0.017
YiSi-2-src	-0.017
regEMT-src	-0.017
COMETinho-DA	-0.021
COMET-DA_2020	-0.022
COMETinho-MQM	-0.023
C-SPEC	-0.029
Prism	-0.033
BERTScore	-0.081
bleurt-20	-0.105
bleurt-21-beta	-0.109
chrF	-0.126
YiSi-1	-0.127
regEMT-baseline-src	-0.139
sentBLEU	-0.144
hLEPOR	-0.144
regEMT-baseline	-0.167
TER	-0.263
MTEQA	-0.314

Table 34: Correlations for Russian→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
COMET-MQM_2021	0.762
COMET-DA_2021	0.756
bleurt-21-beta	0.754
bleurt-20	0.749
COMET-DA_2020	0.748
COMET-QE-DA_2021-src	0.746
YiSi-1	0.735
COMET-QE-MQM_2021-src	0.731
BERTScore	0.727
MEE	0.726
Prism	0.726
chrF	0.723
RoBLEURT	0.720
regEMT	0.712
sentBLEU	0.709
MEE2	0.707
OpenKiwi-MQM-src	0.706
regEMT-baseline	0.699
COMETinho-DA	0.693
cushLEPOR(pSQM)	0.679
cushLEPOR(LM)	0.678
MTEQA	0.661
BLEU	0.653
COMETinho-MQM	0.568
hLEPOR	0.550
YiSi-2-src	0.542
TER	0.527
regEMT-src	0.378
C-SPEC	0.218
C-SPECpn	0.214
regEMT-baseline-src	-0.669

metric	correlation
OpenKiwi-MQM-src	0.021
COMET-MQM_2021	0.020
COMET-QE-MQM_2021-src	0.020
RoBLEURT	0.019
COMET-DA_2021	0.018
COMETinho-DA	0.018
COMET-QE-DA_2021-src	0.017
COMET-DA_2020	0.016
YiSi-2-src	0.008
COMETinho-MQM	0.008
Prism	0.007
regEMT-src	-0.005
C-SPECpn	-0.005
regEMT	-0.006
C-SPEC	-0.007
BERTScore	-0.013
bleurt-21-beta	-0.022
YiSi-1	-0.026
bleurt-20	-0.028
MEE2	-0.035
chrF	-0.035
hLEPOR	-0.050
cushLEPOR(LM)	-0.050
cushLEPOR(pSQM)	-0.056
sentBLEU	-0.057
MEE	-0.063
regEMT-baseline	-0.089
regEMT-baseline-src	-0.090
MTEQA	-0.118
TER	-0.165

Table 35: Correlations for Chinese→English: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
YiSi-1	0.781
COMET-DA_2021	0.774
COMET-DA_2020	0.743
bleurt-20	0.734
bleurt-21-beta	0.721
COMET-MQM_2021	0.693
COMET-QE-DA_2021-src	0.658
COMETinho-DA	0.620
Prism	0.584
regEMT	0.534
regEMT-baseline	0.526
COMET-QE-MQM_2021-src	0.516
OpenKiwi-MQM-src	0.460
YiSi-2-src	0.414
chrF	0.413
BERTScore	0.378
TER	0.363
sentBLEU	0.363
C-SPECpn	0.329
C-SPEC	0.320
COMETinho-MQM	0.298
regEMT-src	0.101
regEMT-baseline-src	-0.433

metric	correlation
COMET-MQM_2021	0.223
COMET-DA_2021	0.220
Prism	0.208
COMET-QE-DA_2021-src	0.203
bleurt-20	0.202
COMET-DA_2020	0.202
bleurt-21-beta	0.193
C-SPEC	0.189
YiSi-1	0.173
COMET-QE-MQM_2021-src	0.161
COMETinho-DA	0.156
C-SPECpn	0.143
OpenKiwi-MQM-src	0.123
COMETinho-MQM	0.118
BERTScore	0.116
chrF	0.110
regEMT	0.104
YiSi-2-src	0.104
sentBLEU	0.055
regEMT-src	0.041
regEMT-baseline	0.031
TER	-0.063
regEMT-baseline-src	-0.201

Table 36: Correlations for German→French: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-21-beta	0.991
bleurt-20	0.989
YiSi-1	0.985
COMET-MQM_2021	0.979
COMET-DA_2021	0.979
sentBLEU	0.976
chrF	0.975
COMET-QE-MQM_2021-src	0.974
Prism	0.971
COMET-DA_2020	0.971
regEMT	0.969
TER	0.967
COMET-QE-DA_2021-src	0.966
BERTScore	0.965
hLEPOR	0.956
COMETinho-DA	0.941
MTEQA	0.905
COMETinho-MQM	0.895
OpenKiwi-MQM-src	0.873
regEMT-baseline	0.866
regEMT-src	0.595
C-SPEC	0.123
C-SPECpn	0.072
YiSi-2-src	-0.007
regEMT-baseline-src	-0.920

metric	correlation
COMET-DA_2021	0.774
bleurt-20	0.764
COMET-MQM_2021	0.757
C-SPEC	0.753
bleurt-21-beta	0.752
COMET-DA_2020	0.737
COMET-QE-DA_2021-src	0.724
C-SPECpn	0.723
COMET-QE-MQM_2021-src	0.714
Prism	0.712
YiSi-1	0.686
OpenKiwi-MQM-src	0.652
regEMT	0.641
COMETinho-DA	0.573
BERTScore	0.571
chrF	0.531
COMETinho-MQM	0.492
hLEPOR	0.441
sentBLEU	0.383
YiSi-2-src	0.240
MTEQA	0.212
TER	0.208
regEMT-src	0.160
regEMT-baseline	0.126
regEMT-baseline-src	-0.349

Table 37: Correlations for English→Czech: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-20	0.885
bleurt-21-beta	0.882
COMET-MQM_2021	0.880
COMET-QE-MQM_2021-src	0.877
COMET-DA_2021	0.875
OpenKiwi-MQM-src	0.870
COMET-QE-DA_2021-src	0.866
COMET-DA_2020	0.864
regEMT	0.855
COMETinho-DA	0.854
Prism	0.853
YiSi-1	0.847
COMETinho-MQM	0.835
chrF	0.820
MTEQA	0.797
TER	0.794
BERTScore	0.794
MEE2	0.793
sentBLEU	0.769
MEE	0.761
BLEU	0.738
cushLEPOR(pSQM)	0.699
cushLEPOR(LM)	0.691
hLEPOR	0.671
regEMT-src	0.481
C-SPECpn	0.408
C-SPEC	0.258
YiSi-2-src	0.025
regEMT-baseline	-0.272
regEMT-baseline-src	-0.727

metric	correlation
COMET-DA_2021	0.255
COMET-DA_2020	0.255
COMET-MQM_2021	0.247
COMET-QE-DA_2021-src	0.237
COMET-QE-MQM_2021-src	0.230
regEMT	0.220
Prism	0.208
OpenKiwi-MQM-src	0.205
bleurt-21-beta	0.202
C-SPEC	0.200
bleurt-20	0.200
C-SPECpn	0.199
YiSi-1	0.162
COMETinho-DA	0.162
COMETinho-MQM	0.157
BERTScore	0.146
MEE2	0.102
chrF	0.098
YiSi-2-src	0.075
regEMT-src	0.067
cushLEPOR(LM)	0.033
hLEPOR	0.026
cushLEPOR(pSQM)	0.025
MEE	0.019
sentBLEU	0.014
MTEQA	-0.122
TER	-0.123
regEMT-baseline	-0.136
regEMT-baseline-src	-0.180

Table 38: Correlations for English→German: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-20	0.915
bleurt-21-beta	0.907
regEMT	0.901
YiSi-1	0.892
COMET-DA_2020	0.871
COMET-DA_2021	0.863
BERTScore	0.838
COMET-MQM_2021	0.811
OpenKiwi-MQM-src	0.791
sentBLEU	0.789
COMET-QE-DA_2021-src	0.786
chrF	0.768
COMET-QE-MQM_2021-src	0.746
COMETinho-DA	0.708
COMETinho-MQM	0.463
regEMT-baseline	0.376
YiSi-2-src	0.362
TER	0.288
C-SPEC	0.174
C-SPECpn	0.077
regEMT-src	-0.266
regEMT-baseline-src	-0.357

metric	correlation
COMET-DA_2021	0.237
COMET-DA_2020	0.234
COMET-MQM_2021	0.214
C-SPEC	0.210
COMET-QE-DA_2021-src	0.198
bleurt-20	0.186
C-SPECpn	0.186
chrF	0.186
bleurt-21-beta	0.183
YiSi-1	0.180
COMET-QE-MQM_2021-src	0.176
BERTScore	0.167
OpenKiwi-MQM-src	0.157
COMETinho-DA	0.131
regEMT	0.130
sentBLEU	0.124
YiSi-2-src	0.102
COMETinho-MQM	0.088
regEMT-baseline	0.049
regEMT-src	0.016
TER	-0.025
regEMT-baseline-src	-0.112

Table 39: Correlations for English→Hausa: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
regEMT	0.989
bleurt-20	0.975
sentBLEU	0.962
bleurt-21-beta	0.962
chrF	0.961
COMET-MQM_2021	0.960
COMET-DA_2021	0.959
COMET-QE-MQM_2021-src	0.959
YiSi-1	0.957
COMET-DA_2020	0.952
BERTScore	0.950
COMET-QE-DA_2021-src	0.945
COMETinho-DA	0.931
TER	0.928
COMETinho-MQM	0.908
OpenKiwi-MQM-src	0.873
C-SPECpn	0.750
C-SPEC	0.736
regEMT-baseline	0.478
YiSi-2-src	0.348
regEMT-src	0.125
regEMT-baseline-src	-0.922

metric	correlation
COMET-MQM_2021	0.489
COMET-DA_2021	0.487
COMET-DA_2020	0.474
C-SPEC	0.472
bleurt-20	0.469
C-SPECpn	0.460
COMET-QE-DA_2021-src	0.454
COMET-QE-MQM_2021-src	0.453
bleurt-21-beta	0.444
YiSi-1	0.410
OpenKiwi-MQM-src	0.404
COMETinho-DA	0.384
chrF	0.373
BERTScore	0.355
COMETinho-MQM	0.330
regEMT	0.312
sentBLEU	0.279
TER	0.121
YiSi-2-src	0.105
regEMT-src	0.012
regEMT-baseline	0.002
regEMT-baseline-src	-0.199

Table 40: Correlations for English→Icelandic: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-21-beta	0.991
COMET-DA_2020	0.988
bleurt-20	0.985
COMET-DA_2021	0.984
COMET-QE-DA_2021-src	0.978
YiSi-1	0.974
COMETinho-DA	0.972
regEMT	0.972
Prism	0.971
COMET-MQM_2021	0.970
chrF	0.966
COMETinho-MQM	0.955
COMET-QE-MQM_2021-src	0.947
BERTScore	0.939
OpenKiwi-MQM-src	0.929
C-SPECpn	0.678
regEMT-src	0.502
YiSi-2-src	0.470
regEMT-baseline	0.423
C-SPEC	0.325
TER	-0.025
regEMT-baseline-src	-0.216
sentBLEU	-0.629

metric	correlation
COMET-DA_2021	0.531
COMET-DA_2020	0.519
COMET-MQM_2021	0.490
C-SPEC	0.484
COMET-QE-DA_2021-src	0.484
bleurt-21-beta	0.483
bleurt-20	0.483
COMETinho-DA	0.457
C-SPECpn	0.454
Prism	0.440
YiSi-1	0.425
BERTScore	0.417
COMET-QE-MQM_2021-src	0.379
chrF	0.371
regEMT	0.369
COMETinho-MQM	0.348
OpenKiwi-MQM-src	0.333
YiSi-2-src	0.229
regEMT-baseline	0.079
regEMT-src	0.065
regEMT-baseline-src	-0.161
TER	-0.791
sentBLEU	-0.881

Table 41: Correlations for English→Japanese: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-21-beta	0.978
COMET-DA_2020	0.973
COMET-DA_2021	0.973
bleurt-20	0.972
COMET-MQM_2021	0.972
COMET-QE-DA_2021-src	0.970
COMET-QE-MQM_2021-src	0.969
sentBLEU	0.967
BERTScore	0.964
hLEPOR	0.959
MEE	0.956
MEE2	0.951
YiSi-1	0.948
OpenKiwi-MQM-src	0.948
chrF	0.946
BLEU	0.946
COMETinho-DA	0.944
Prism	0.924
TER	0.903
COMETinho-MQM	0.835
regEMT	0.810
regEMT-baseline	0.377
YiSi-2-src	0.029
regEMT-src	0.005
C-SPECpn	-0.160
regEMT-baseline-src	-0.410
C-SPEC	-0.417

metric	correlation
COMET-DA_2021	0.401
COMET-MQM_2021	0.397
COMET-DA_2020	0.368
COMET-QE-DA_2021-src	0.365
C-SPEC	0.360
C-SPECpn	0.348
bleurt-21-beta	0.340
Prism	0.330
COMET-QE-MQM_2021-src	0.326
bleurt-20	0.323
YiSi-1	0.294
BERTScore	0.255
COMETinho-DA	0.246
OpenKiwi-MQM-src	0.234
MEE2	0.233
chrF	0.201
COMETinho-MQM	0.167
MEE	0.161
hLEPOR	0.157
regEMT	0.122
sentBLEU	0.105
YiSi-2-src	0.051
regEMT-src	0.024
regEMT-baseline	-0.002
TER	-0.078
regEMT-baseline-src	-0.183

Table 42: Correlations for English→Russian: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
COMET-DA_2021	0.946
COMET-DA_2020	0.939
COMET-QE-DA_2021-src	0.927
bleurt-21-beta	0.910
COMET-MQM_2021	0.903
COMETinho-DA	0.900
OpenKiwi-MQM-src	0.892
YiSi-1	0.888
BERTScore	0.851
regEMT	0.823
COMET-QE-MQM_2021-src	0.815
bleurt-20	0.813
Prism	0.750
chrF	0.570
COMETinho-MQM	0.467
YiSi-2-src	0.313
regEMT-src	0.302
C-SPECpn	0.286
C-SPEC	0.235
TER	0.169
regEMT-baseline	0.014
regEMT-baseline-src	-0.039
sentBLEU	-0.156

metric	correlation
COMET-DA_2021	0.270
COMET-QE-DA_2021-src	0.261
COMET-DA_2020	0.247
COMET-MQM_2021	0.246
bleurt-20	0.240
bleurt-21-beta	0.239
C-SPECpn	0.224
C-SPEC	0.224
COMET-QE-MQM_2021-src	0.216
Prism	0.207
COMETinho-DA	0.202
YiSi-1	0.192
BERTScore	0.189
OpenKiwi-MQM-src	0.180
COMETinho-MQM	0.121
regEMT	0.119
YiSi-2-src	0.095
chrF	0.092
regEMT-src	0.016
regEMT-baseline	-0.047
regEMT-baseline-src	-0.187
TER	-0.701
sentBLEU	-0.715

Table 43: Correlations for English→Chinese: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-21-beta	0.789
COMET-DA_2020	0.770
bleurt-20	0.768
COMET-MQM_2021	0.763
COMET-DA_2021	0.759
Prism	0.729
COMET-QE-MQM_2021-src	0.712
YiSi-1	0.709
COMET-QE-DA_2021-src	0.708
regEMT	0.702
COMETinho-DA	0.695
BERTScore	0.674
sentBLEU	0.660
chrF	0.647
COMETinho-MQM	0.640
OpenKiwi-MQM-src	0.626
TER	0.615
MTEQA	0.609
C-SPEC	0.191
regEMT-baseline	0.081
regEMT-src	-0.002
regEMT-baseline-src	-0.082
C-SPECpn	-0.267
YiSi-2-src	-0.290

metric	correlation
COMET-DA_2021	0.108
COMET-DA_2020	0.101
regEMT	0.097
COMET-QE-DA_2021-src	0.091
COMET-MQM_2021	0.090
Prism	0.090
bleurt-21-beta	0.081
C-SPEC	0.079
bleurt-20	0.079
YiSi-2-src	0.072
C-SPECpn	0.069
BERTScore	0.068
COMETinho-DA	0.068
OpenKiwi-MQM-src	0.056
chrF	0.054
YiSi-1	0.053
COMET-QE-MQM_2021-src	0.052
COMETinho-MQM	0.052
regEMT-src	0.039
sentBLEU	0.005
regEMT-baseline	-0.081
MTEQA	-0.089
TER	-0.093
regEMT-baseline-src	-0.109

Table 44: Correlations for French→German: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
regEMT	0.964
bleurt-21-beta	0.964
COMET-QE-MQM_2021-src	0.963
bleurt-20	0.963
COMET-QE-DA_2021-src	0.957
COMET-DA_2020	0.955
OpenKiwi-MQM-src	0.953
COMET-MQM_2021	0.949
YiSi-1	0.948
COMET-DA_2021	0.946
chrF	0.941
BERTScore	0.935
COMETinho-DA	0.928
COMETinho-MQM	0.923
TER	0.912
sentBLEU	0.901
regEMT-baseline	0.889
C-SPEC	0.743
YiSi-2-src	0.668
C-SPECpn	0.503
regEMT-baseline-src	0.033
regEMT-src	-0.245

metric	correlation
bleurt-20	0.179
bleurt-21-beta	0.170
C-SPEC	0.157
COMET-DA_2020	0.156
COMET-MQM_2021	0.153
C-SPECpn	0.150
COMET-QE-DA_2021-src	0.146
COMET-DA_2021	0.146
OpenKiwi-MQM-src	0.137
YiSi-1	0.134
COMETinho-DA	0.125
regEMT	0.111
YiSi-2-src	0.110
COMET-QE-MQM_2021-src	0.109
COMETinho-MQM	0.101
BERTScore	0.093
chrF	0.071
sentBLEU	0.070
regEMT-src	-0.027
TER	-0.030
regEMT-baseline	-0.040
regEMT-baseline-src	-0.054

Table 45: Correlations for FLORES Bengali→Hindi: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
Prism	0.990
regEMT	0.987
COMET-QE-DA_2021-src	0.987
COMET-QE-MQM_2021-src	0.986
OpenKiwi-MQM-src	0.982
bleurt-21-beta	0.975
COMETinho-MQM	0.975
COMET-DA_2020	0.974
bleurt-20	0.973
COMET-MQM_2021	0.970
COMET-DA_2021	0.965
COMETinho-DA	0.964
YiSi-1	0.947
BERTScore	0.918
YiSi-2-src	0.898
chrF	0.872
TER	0.871
regEMT-baseline	0.856
sentBLEU	0.784
C-SPECpn	-0.116
C-SPEC	-0.539
regEMT-baseline-src	-0.886
regEMT-src	-0.955

metric	correlation
Prism	0.566
COMET-QE-MQM_2021-src	0.524
COMET-QE-DA_2021-src	0.524
COMET-DA_2020	0.518
COMET-MQM_2021	0.517
COMET-DA_2021	0.510
bleurt-20	0.499
bleurt-21-beta	0.488
C-SPECpn	0.477
YiSi-2-src	0.468
COMETinho-MQM	0.462
COMETinho-DA	0.453
YiSi-1	0.442
C-SPEC	0.418
OpenKiwi-MQM-src	0.412
BERTScore	0.366
chrF	0.327
sentBLEU	0.246
regEMT	0.205
TER	0.108
regEMT-baseline	0.050
regEMT-src	-0.067
regEMT-baseline-src	-0.188

Table 46: Correlations for FLORES Hindi→Bengali: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
COMET-DA_2021	0.999
bleurt-21-beta	0.998
YiSi-1	0.998
chrF	0.998
COMET-MQM_2021	0.997
bleurt-20	0.997
COMET-QE-MQM_2021-src	0.997
COMETinho-DA	0.997
BERTScore	0.995
COMET-DA_2020	0.993
regEMT	0.990
TER	0.981
sentBLEU	0.979
C-SPECpn	0.974
COMETinho-MQM	0.971
OpenKiwi-MQM-src	0.952
C-SPEC	0.942
COMET-QE-DA_2021-src	0.936
regEMT-baseline	0.781
regEMT-src	0.536
YiSi-2-src	0.381
regEMT-baseline-src	0.363

metric	correlation
C-SPEC	0.368
bleurt-20	0.363
bleurt-21-beta	0.359
C-SPECpn	0.340
chrF	0.301
COMET-DA_2021	0.297
COMET-MQM_2021	0.293
YiSi-1	0.293
OpenKiwi-MQM-src	0.286
COMET-QE-DA_2021-src	0.285
COMET-DA_2020	0.281
COMET-QE-MQM_2021-src	0.276
BERTScore	0.270
COMETinho-MQM	0.219
COMETinho-DA	0.209
sentBLEU	0.188
YiSi-2-src	0.153
regEMT-src	0.150
regEMT	0.126
TER	0.074
regEMT-baseline	-0.014
regEMT-baseline-src	-0.053

Table 47: Correlations for FLORES Xhosa→Zulu: system-level Pearson (left panel) , segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

metric	correlation
bleurt-21-beta	1.000
chrF	0.999
YiSi-1	0.998
bleurt-20	0.998
BERTScore	0.997
COMET-MQM_2021	0.997
COMETinho-DA	0.996
COMET-DA_2021	0.996
COMETinho-MQM	0.991
COMET-QE-MQM_2021-src	0.990
COMET-DA_2020	0.990
regEMT	0.983
TER	0.978
OpenKiwi-MQM-src	0.973
COMET-QE-DA_2021-src	0.953
sentBLEU	0.903
YiSi-2-src	0.758
C-SPECpn	0.713
regEMT-baseline	0.681
C-SPEC	0.604
regEMT-baseline-src	0.432
regEMT-src	-0.044

metric	correlation
COMET-DA_2021	0.571
bleurt-20	0.564
bleurt-21-beta	0.559
C-SPEC	0.552
C-SPECpn	0.552
COMET-MQM_2021	0.550
COMET-DA_2020	0.545
YiSi-1	0.544
COMET-QE-MQM_2021-src	0.538
chrF	0.530
COMET-QE-DA_2021-src	0.530
OpenKiwi-MQM-src	0.523
BERTScore	0.491
YiSi-2-src	0.472
COMETinho-DA	0.436
COMETinho-MQM	0.423
sentBLEU	0.381
TER	0.296
regEMT	0.202
regEMT-baseline	0.022
regEMT-src	-0.010
regEMT-baseline-src	-0.037

Table 48: Correlations for FLORES Zulu→Xhosa: system-level Pearson (left panel), segment-level Kendall-Like (right panel). Primary submissions are bolded, and baselines are underlined.

Efficient Machine Translation with Model Pruning and Quantization

Maximiliana Behnke[†] **Nikolay Bogoychev**[†] **Alham Fikri Aji**[†] **Kenneth Heafield**[†]
Graeme Nail[†] **Qianqian Zhu**[†] **Svetlana Tchistiakova**[†] **Jelmer van der Linde**[†]
Pinzhen Chen[†] **Sidharth Kashyap**[‡] **Roman Grundkiewicz**[†§]
[†]University of Edinburgh [‡]Intel Corporation [§]Microsoft
{maximiliana.behnke, n.bogoych, a.fikri, kenneth.heafield, graeme.nail, qianqian.zhu,
stchisti, jelmer.vanderlinde, pinzhen.chen, rgrundki}@ed.ac.uk,
sidharth.n.kashyap@intel.com

Abstract

We participated in all tracks of the WMT 2021 efficient machine translation task: single-core CPU, multi-core CPU, and GPU hardware with throughput and latency conditions. Our submissions combine several efficiency strategies: knowledge distillation, a simpler simple recurrent unit (SSRU) decoder with one or two layers, lexical shortlists, smaller numerical formats, and pruning. For the CPU track, we used quantized 8-bit models. For the GPU track, we experimented with FP16 and 8-bit integers in tensorcores. Some of our submissions optimize for size via 4-bit log quantization and omitting a lexical shortlist. We have extended pruning to more parts of the network, emphasizing component- and block-level pruning that actually improves speed unlike coefficient-wise pruning.

1 Introduction

This paper describes the University of Edinburgh's submission to Sixth Conference on Machine Translation (WMT2021) Efficiency Task[1], which measures performance on latency and throughput on both CPU and GPU, in addition to translation quality. Our submission focused on the trade-off between these metrics and quality.

Our submission builds upon the work of last year's submission (Bogoychev et al., 2020). We trained our models in a teacher-student setting (Kim and Rush, 2016), using Edinburgh's En-De system submitted to the WMT2021 news translation task as the teacher model. For the students, we used a Simpler Simple Recurrent Unit (SSRU) (Kim et al., 2019) decoder, used a target vocabulary shortlist, and experimented with pruning the student models by removing component- and block-level parameters to improve speed. We further experimented with quantizing into smaller numerical formats, including fixed point 8-bit quantization on the CPU, and both 8-bit and log based 4-bit quantization on the GPU, as well as post-quantization fine-tuning of 4-bit quantized models.

For running our experiments, we improved upon the Marian (Junczys-Dowmunt et al., 2018) machine translation framework by incorporating speed ups for 8-bit matrix multiplication operations, optimizations for pruning neural network parameters on Intel CPUs, and exploring tensorcores on the GPU.

1.1 Efficiency Shared Task

The WMT21 efficiency shared task consists of two sub-tasks: throughput and latency. Systems should translate English to German under the constrained conditions of the WMT21 news task. For each task, systems are provided 1 million lines of raw English input with at most 150 space-separated words. The throughput task receives this input directly. The latency task, introduced this year, is fed input one sentence at a time, waiting for the translation output before providing the next sentence.

Throughput is measured on multi-core CPU or GPU system, and latency is measured on single-core CPU or GPU systems. The CPU-based evaluations use an Intel Ice Lake system via Oracle Cloud BM.Optimized3.36, while the GPU-based use a single A100 via Oracle Cloud BM.GPU4.8.

Entries to both tasks are measured on quality, approximated via BLEU score (Papineni et al., 2002), speed, model size, Docker image size, and memory consumption. We did not optimise specifically for the latency task beyond configuring the relevant batch sizes to one. We used Ubuntu 20.04 based images for our systems, with standard Ubuntu for CPU-only systems and NVIDIA's Ubuntu-based CUDA-11.4 docker for GPU-capable systems. Docker images were created using multi-stage builds, with model disk size reduced by compression with xzip.

[1] http://statmt.org/wmt21/efficiency-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 775–780
November 10–11, 2021.

2 Training teacher models

We used Edinburgh's En↔De systems submitted to the WMT 2021 news translation task as teacher models (Chen et al., 2021). We trained transformer-big models (Vaswani et al., 2017), using a shared 32K SentencePiece (Kudo and Richardson, 2018) vocabulary, built in three stages: corpus filtering, back-translation and fine-tuning. The models achieved 29.90 and 51.78 BLEU on En→De and De→En WMT 2021 test respectively (scored by the task organizers, with multiple references).

We used sequence-level knowledge distillation (Kim and Rush, 2016) to synthesize forward, backward, and backward-forward translations using the teachers. We filtered the synthesized parallel data using handcrafted rules[2], followed by removing bottom 5% according to cross-entropy per word on the generated side using KenLM (Heafield et al., 2013).

3 Knowledge distillation

We ran experiments using different combinations of teacher-synthesized corpora. One variant included all of the synthesized data: parallel, monolingual backward and forward as well as backward-forward (Aji and Heafield, 2020b). Another variant excludes only the fully-synthetic monolingual backward-forward data, while the final variant used parallel data only. All student models were trained using a validation set consisting of the subset of sentences in the English-German WMT test sets from 2015–2019 that were originally in English. Training concluded after reaching 20 consecutive validations without an improvement in BLEU score. The student models used the same shared vocabulary as the teacher ensemble. During decoding, we used a lexical shortlist (Schwenk et al., 2007; Le et al., 2012; Devlin et al., 2014) of the top 50 most probable alignments, combined through a union with the top 50 most frequent vocabulary items. Other than this, we used the default training hyperparameters from Marian for the transformer-base model.

Each of the student models used transformer encoders (Vaswani et al., 2017) and RNN-based decoders with Simpler Simple Recurrent Unit (SSRU) (Kim et al., 2019). Several different architectures were explored; these differ in the number of encoder and decoder blocks as well as in the sizes of the embedding and FFN layers. Further to this, some of our transformer architectures use a modified attention matrix of shape $(d_{\text{emb}}, n_{\text{head}} \times d_{\text{head}})$ rather than the typical $(d_{\text{emb}}, d_{\text{emb}})$. In all cases we use 8 transformer heads per layer, and set $d_{\text{head}} = 32$ across all modified attention models.

The student architectures are summarized in Table 1. A baseline comparison of student models trained on all synthesized data can be seen in Table 1.

3.1 Pruning

Attention is a crucial part of the transformer architecture, but it is also computationally expensive. Research has shown that many heads can be pruned after training; with further work suggesting that pruning during training can be less damaging to quality. Feedforward layers are also expensive and could be reduced.

Among many experiments, we applied group lasso regularisation to sparsify and prune 12–1.tiny and 12–1.micro architectures. We follow the directions set by Behnke and Heafield (2021). We tried two pruning settings: *rowcol-lasso* and *head-lasso*. Both prune feedforward and attention layers in the encoder. *rowcol-lasso* regularised individual connections (rows and columns) and removed an entire attention head if at least half of its connections are dead. *head-lasso* applied lasso to a whole head submatrix. Due to the scale of the task, we had no opportunity to grid-search for the best pruning hyperparameters, thus the experiments are as close to 'out-of-the-box' usage as they can be. We control pruning with $\lambda = 0.5$ for both methods. The models were pretrained for 50k updates and regularised for 150k, after which the models were sliced and trained until convergence. The results are presented in Tab. 2.

head-lasso left attention layers almost completely unpruned, focusing on removing connections from feedforward layers instead. *rowcol-lasso* was much more aggressive in both layers at the cost of quality. Behnke and Heafield (2021) have shown that group lasso pruning results in a better quality model than training the same exact architecture from scratch. To further optimise the models, they were quantised to work within 8bit representation. However, we observe that the smaller a model is, the larger the quality drop after its quantisation. Additional finetuning allows us to recover at least partially from the quantisation

[2] https://github.com/browsermt/students/tree/master/train-student/clean

Model	Depth Enc	Dec	Dimensions Emb.	FFN	Att.	Heads	Params.	Size	BLEU WMT20	WMT21	COMET WMT20	WMT21	Speed (s)
teacher x 3	6	6/6/8	1024	4096	1024	16	619.0M	1.59GB	38.3	28.8	56.8	50.8	-
12-1.large	12	1	1024	3072	256	8	130.5M	498MB	37.6	28.7	54.0	47.7	92.2
12-1.base	12	1	512	2048	256	8	51.1M	195MB	36.7	28.2	50.7	44.1	38.9
12-1.tiny	12	1	256	1536	256	8	22.0M	85MB	36.1	27.6	48.2	41.9	19.2
12-1.micro	12	1	256	1024	256	8	18.6M	72MB	35.4	27.6	46.2	40.2	17.1
8-4.tied.tiny	8	4	256	1536	256	8	17.8M	69MB	35.7	27.8	50.3	43.9	30.4
6-2.tied.tiny	6	2	256	1536	256	8	15.7M	61MB	34.9	27.4	47.4	42.1	18.6
6-2.base	6	2	512	2048	512	8	42.7M	163MB	37.7	28.7	54.3	48.5	56.2
6-2.tiny	6	2	256	1536	256	8	16.9M	65MB	35.8	27.4	50.2	44.5	19.2

Table 1: Architectures for the different student models. The number of encoder/decoder layers are reported with the size of the embedding, attention and FFN layers, the total number of parameters, the model size on disk, quality in both BLEU and COMET as well as speed on WMT21 testset. The first and second groups use a modified attention matrix shape, with second group consisting of tied models. The third group uses the typical shape attention matrices.

damage. Evaluating on the latest testset WMT21, our pruned models are 1.2–1.7× faster at the cost of 0.6–1.3 BLEU. With quantisation, those models are 1.9–2.7× faster losing 0.9–1.7 BLEU in comparison to the unpruned and unquantised baselines.

3.2 Fixed Point 8-bit Quantization

Quantizing fp32 models into 8-bit integers is a known strategy to reduce decoding time, specifically on CPU, with a minimal impact on quality (Kim et al., 2019; Bhandare et al., 2019; Rodriguez et al., 2018). This year's submission closely follows the quantization scheme of last year's work (Bogoychev et al., 2020).

Quantization entails computing a scaling factor to collapse the range of values to $[-127, 127]$. For parameters, this scaling factor is computed offline using the maximum absolute value but activation tensors change at runtime. This year, we changed from computing a dynamic scaling factor on the fly for activations to computing a static scaling factor offline. We decoded the WMT16-20 datasets and recorded the scaling factor $\alpha(A_i) = 127/max(|A_i|)$ for each instance A_i of an activation tensor A. Then, for production, we fixed the scaling factor for activation tensor A to the mean scaling factor plus 1.1 standard deviation: $\alpha(A) = \mu(\{\alpha(A_i)\}) + 1.1 * \sigma(\{\alpha(A_i)\})$. These scaling factors were baked into the model file so that statistics were not computed at runtime.

Quantization does not extend to the attention layer, which is still computed in fp32. The reason being is that in the attention layer, both the A and B matrices of the GEMM operation would need to be quantized at runtime, which makes the quantization too expensive. We note that we only perform the GEMM operations in 8-bit integers.

3.3 Log 4-bit Quantization

We further quantize the models with log based 4-bit quantization (Aji and Heafield, 2020a). In this case, model weights are represented in a 16 unique quantization centers in a form of $S * 2^k$. S is a scaling factor that is optimized to minimize the MSE of the quantized weight to the actual weight. Following Aji and Heafield (2020a), we only perform 4-bit quantization on non-bias layers.

Unfortunately, the hardware used is not designed to perform native 4-bit operations. Therefore, our 4-bit quantization experiment is used solely for model compression purposes, in which we can reduce the model size to be 8x smaller. To perform inference, we de-quantize the 4-bit model back to fp32 representation, therefore does not achieve any speed up over the vanilla fp32 models.

3.4 Quantization fine-tuning

Quantizing models degrades the quality, especially on smaller architectures. Therefore, after applying quantization, we fine-tune the model under the quantized weight. We find that lowering the learning rate to 0.0001 yields better model quality. Moreover, for 4-bit models, we also find that doubling the warm-up duration helps.

Our 8-bit quantization models mainly aim for speed improvement. Therefore, we apply 8-bit quantization to pruned models to further boost the speed. As shown in Table 2, 8-bit inference achieves significant speedup. However, fine-tuning is necessary to restore the quality degradation.

	BLEU		COMET		Sparsity		
	WMT20	WMT21	WMT20	WMT21	Att.	FFN	Speed (s)
12-1.tiny	36.1	27.6	48.2	41.9	0%	0%	19.2
+ head-lasso pruning	34.7	27.0	42.9	38.8	3%	75%	14.5
+ 8bit quantisation	33.9	26.2	38.8	33.6	3%	75%	9.3
+ finetuning	34.1	26.7	39.8	33.0	3%	75%	9.3
+ rowcol-lasso pruning	33.8	26.3	39.3	34.2	68%	73%	11.6
+ 8bit quantisation	32.9	25.6	33.7	28.7	68%	73%	6.9
+ finetuning	32.9	26.0	35.7	31.3	68%	73%	7.1
12-1.micro	35.4	27.6	46.2	40.2	0%	0%	17.1
+ head-lasso pruning	34.6	26.7	43.0	35.4	3%	72%	14.1
+ 8bit quantisation	33.4	26.0	36.7	31.2	3%	72%	9.2
+ finetuning	33.7	26.5	38.3	33.3	3%	72%	9.2
+ rowcol-lasso pruning	34.3	26.4	40.7	35.1	60%	59%	12.0
+ 8bit quantisation	32.7	25.5	34.2	29.1	60%	59%	7.5
+ finetuning	33.3	25.9	35.2	30.5	60%	59%	7.5

Table 2: 8-bit model performance. BLEU score is calculated from WMT20. Speed is measured on a single core CPU with a mini-batch of 32. We experimented with two types of pruning. Head pruning removes entire heads. Row and column pruning removes entire rows or columns of matrices, resulting in a smaller matrix.

	BLEU		COMET		
	WMT20	WMT21	WMT20	WMT21	Size
12-1.base	37.1	28.3	51.5	45.1	195MB
+ 4bit	36.3	27.7	50.0	43.2	25MB
12-1.tiny	36.0	28.0	47.5	42.5	85MB
+ 4bit	35.0	27.6	42.4	38.3	11MB
8-4.tied.tiny	35.7	27.5	49.4	43.6	69MB
+ 4bit	34.2	26.4	44.4	38.2	9MB

Table 3: 4-bit model performance. BLEU score is calculated from WMT20. All the quantized models include fine-tuning. The inference is done in 32fp, therefore their speed are comparable.

We apply 4-bit quantization solely for size efficiency. Therefore, we quantize non-pruned models since they give better size to quality trade-off, compared to pruned models. The performance of 4-bit models can be seen in Table 3.

4 Software improvements

4.1 CPU

We built our work using the Marian machine translation framework, making some improvements on top of the submission from last year: We used predominantly *intgemm*[3] for our 8-bit GEMM operations, including for the shortlisted output layer. All parameter matrices are quantized to 8-bit offline and the activations get quantized dynamically before a GEMM operation. We only perform the GEMM operation and the following activation in 8-bit integer mode. Right after a GEMM operation, the output is de-quantized back to fp32. More formally we perform $dequantize(\sigma(A * B + bias))$, where the addition of the $bias$, the activation function[4] σ, and the de-quantization are applied in a streaming fashion to prevent a round trip to memory.

Furthermore we make use of Intel's *DNNL*[5] for our pruned models, as it performs better than *intgemm* for irregular sized matrices. Unfortunately, *DNNL* doesn't support streaming de-quantization, bias addition or activation function application.

For the CPU_ALL throughput track, we swept configurations of multiple processes and threads on the platform, settling on 4 processes with 9 threads each. The input text is simply split into 4 pieces and parallelized (Tange, 2011) over processes. The mini-batch sizes did not impact performance substantially and 32 was chosen as the mini-batch size. The Hyperthreads available on the platform were not put into use as the compute on each was saturated by the efficient threads. Each process is bound to 9 cores assigned sequentially and to the memory domain corresponding to the socket with those cores using numactl. Output from the data parallel run is then stitched together to produce the final translation.

[3] https://github.com/kpu/intgemm

[4] We only support ReLU activation for now

[5] https://github.com/oneapi-src/oneDNN

mini-batch	master fp32	master fp16	ours fp32	ours fp16	ours 8-bit
32	1160s	1151s	740s	731s	732s
64	696s	636s	426s	400s	416s
128	475s	430s	261s	246s	261s
256	320s	296s	181s	160s	169s
512	282s	241s	147s	127s	133s
768	285s	225s	139s	120s	123s
1024	277s	218s	136s	117s	120s
1132	277s	216s	135s	**116s**	119s
BLEU	33.47	33.43	33.48	33.42	33.26

Table 4: Comparison between the **master** branch of marian-dev, **our** branch and **our** best 8-bit integer tensorcore work for GPU decoding. For grid search we used last year's submission model and tested on 1 million sentences from last year's WNGT competition (Heafield et al., 2020).

4.2 GPU

For our GPU submission we built up on top of last year's submission, applying experimental GPU optimisations on top of the marian-dev master tree[6] and exploring tensorcore[7] applicability using CUTLASS.[8]

Tensorcores can in theory drastically increase the performance of our computations and were enabled for all of our fp16 experiments. Tensorcores can also improve speed when doing 8-bit integer operations, so we implemented 8-bit integer GPU decoding similar to our CPU scheme. We found that shortlisting doesn't improve the performance, so we didn't use it.

We found that while fp16 decoding works fairly well and delivers good performance improvements for decoding, especially when using a really large mini-batch size. We performed a large parameter sweep on a RTX 3090, as shown on Table 4. Unfortunately, we found no setting in which tensorcore 8-bit integer decoding outperforms the fp16 baseline, likely due to the overhead of quantisating the activations beforehand.

5 Conclusion

We participated in all tracks of the WMT 2021 efficiency tracks and we submitted multiple systems that have different trade-offs between speed and translation quality. We performed ample hyperparameter tuning and exploration in order to take advantage of GPU tensorcores for decoding, but unfortunately we couldn't beat our optimised fp16 baseline. For the CPU submission we used 8bit integer decoding and a combination of pruned and non-pruned system, together with a lexical shortlist in order to reduce the computational cost of the largest GEMM in decoding – the output layer.

[6] https://github.com/marian-nmt/marian-dev/pull/743

[7] https://developer.nvidia.com/blog/programming-tensor-cores-cuda-9/

[8] https://github.com/NVIDIA/cutlass

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council (grant EP/L01503X/1), EPSRC Centre for Doctoral Training in Pervasive Parallelism at the University of Edinburgh, School of Informatics.

This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/P020259/1.

References

Alham Fikri Aji and Kenneth Heafield. 2020a. Compressing neural machine translation models with 4-bit precision. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 35–42.

Alham Fikri Aji and Kenneth Heafield. 2020b. Fully synthetic data improves neural machine translation with knowledge distillation. *CoRR*, abs/2012.15455.

Maximiliana Behnke and Kenneth Heafield. 2021. Pruning neural machine translation for speed using group lasso. In *Proceedings of the Six Conference on Machine Translation*, Online. Association for Computational Linguistics.

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model.

Nikolay Bogoychev, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 machine translation efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nicolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. The University of Edinburgh's English-German and English-Hausa submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

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 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. 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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Hai Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–48, Montréal, Canada. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Andres Rodriguez, Eden Segal, Etay Meiri, Evarist Fomenko, Young Jin Kim, Haihao Shen, and Barukh Ziv. 2018. Lower numerical precision deep learning inference and training.

Holger Schwenk, Marta R. Costa-jussà, and Jose A. R. Fonollosa. 2007. Smooth bilingual n-gram translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 430–438, Prague, Czech Republic. Association for Computational Linguistics.

O. Tange. 2011. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

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*, volume 30. Curran Associates, Inc.

HW-TSC's Participation in the WMT 2021 Efficiency Shared Task

Hengchao Shang[1], **Ting Hu**[2], **Daimeng Wei**[1], **Zongyao Li**[1],
Jianfei Feng[2], **Zhengzhe Yu**[1], **Jiaxin Guo**[1], **Shaojun Li**[1],
Lizhi Lei[1], **Shimin Tao**[1], **Hao Yang**[1], **Jun Yao**[2], **Ying Qin**[1],
[1]Huawei Translation Service Center, Beijing, China
[2]Huawei Noah's Ark Lab, Hong Kong, China
{shanghengchao,huting35,weidaimeng,lizongyao,
fengjianfei1,yuzhengzhe,guojiaxin1,lishaojun15,
leilizhi,taoshimin,yanghao30,yaojun97,qinying}@huawei.com

Abstract

This paper presents the submission of Huawei Translation Services Center (HW-TSC) to WMT 2021 Efficiency Shared Task. We explore the sentence-level teacher-student distillation technique and train several small-size models that find a balance between efficiency and quality. Our models feature deep encoder, shallow decoder and light-weight RNN with SSRU layer. We use Huawei Noah's Bolt[1], an efficient and light-weight library for on-device inference. Leveraging INT8 quantization, self-defined General Matrix Multiplication (GEMM) operator, shortlist, greedy search and caching, we submit four small-size and efficient translation models with high translation quality for the one CPU core latency track.

1 Introduction

Transformer and its variants (Vaswani et al., 2017; Shaw et al., 2018; So et al., 2019; Dehghani et al., 2019) have become benchmark models in the domain of machine translation. A lot of innovations and engineering optimizations (Tay et al., 2020) in this area are based on Transformer. In general, to train a high-quality translation model, a large amount of data is required. Expensive training and deployment costs pose great challenges to scenarios where hardware are limited or the deployment environment is complex. This task aims to explore a solution that balances efficient decoding and high-quality translation. We focus on the one CPU latency track, which can better demonstrate the capability of our model and inference framework. We explore a balance between speed and quality, and ensure efficient memory usage and light-weight inference framework capability at the same time. We finally submit four models of different sizes.

We use knowledge distillation (Hinton et al., 2015) to train small models. The teacher models come from our WMT 2021 News Shared Task. For the sake of efficient decoding, our models have only 1-layer decoder. However, the number of encoding layers vary. Such settings lead to a great increase of inference efficiency while ensuring the translation quality (Wang et al., 2019).

All of our experiments are conducted based on fariseq (Ott et al., 2019), including the training of teacher and student models, as well as the generation of distillation data.

We use Huawei Noah's Bolt as the inference library. Bolt is a universal deep learning library featuring light weight and high speed. For the CPU task, we realize INT8 quantization inference and efficient GEMM operator, which is faster than Intel oneDNN[2]. With other engineering optimization strategies, we achieve a significant improvement in terms of inference efficiency.

Section 2 describes the teacher-student knowledge distillation process. Section 3 introduces how we optimize inference for this task. Section 4 presents the final result of our submissions.

2 Teacher to Student Knowledge Distillation

Sentence-level distillation (Kim and Rush, 2016; Freitag et al., 2017) have been demonstrated effective for machine translation tasks. First of all, we train a large teacher model that emphasizes translation quality. Then, we translate the source side of the training data and generate a synthetic parallel corpus, as synthetic data is easier for model fitting than real parallel data. Finally, we train student models using the synthetic data, hoping to minimize model sizes while ensuring equal translation quality as the teacher model. We use KD refer to knowledge distillation.

[1]https://github.com/huawei-noah/bolt

[2]https://github.com/oneapi-src/oneDNN

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 781–786
November 10–11, 2021.

2.1 Teacher Model

As suggested in the task description, we select four iteration models for the third round and also the models before the final round of fine-tuning. All the models adopt back translation (Edunov et al., 2018) and forward translation (Wu et al., 2019) techniques. Our final ensembled model gained 39.7 BLEU on the WMT 2020 test set. The settings of the four models vary. We make sure that the model sizes are similar by adjusting hyperparameters such size of embedding, encoder layers, decoder layers, ffn size, etc. For more details about our teacher model, please refer to our system report for WMT 2021 News Shared Task.

2.2 Traning data

We comply with the constrained condition and use only data from the WMT 2021 En-De News Task. The size of parallel data after filtering is around 80M. In terms of monolingual data, we only use the news-crawl corpus with 230M sentences. So the size of English data we obtained is around 310M. In our teacher-student distillation experiment, we translate all English sentences from the parallel corpus and only 80M sentences sampled from the English monolingual data. Thus, the ratio of real parallel data to synthetic parallel data is 1:2.

When translating English sentences from the parallel corpus, we generate four candidates using the teacher model. Then we calculate the TER scores between those candidates and the corresponding German reference from the parallel corpus and select the candidate with the lowest TER score as the translation result. When translating monolingual data, the beam size is set to 4. For all translation results, we conduct data filtering with language identification using FastText (Joulin et al., 2017). We also delete sentences of which the source side has less than 5 tokens and those with repeated translated segments. The final sizes of our training data are as follow: 79M real parallel data, 73M synthetic data generated from the source side of parallel data, and 76M synthetic data generated from monolingual sentences. Table 1 summarizes the details of data we use.

Type	Corpora	Size
Parallel	Europarl v10	1.8M
	News Commentary v16	0.4M
	Tilde Rapid corpus	1.6M
	Wiki Titles v3	1.4M
	Common Crawl	2.4M
	ParaCrawl v7.1	82.6M
	WikiMatrix	6.2M
	Totle	96.5M
	Filtered	79.4M
Mono	news-crawl	230M
	For KD Translate	80M
KD	Parallel	79.4M
	Parallel En Translated	73.8M
	Mono En Translated	76.8M
	Total	230M

Table 1: Our training data details. The training data consists of three parts: filtered Parallel corpus, Parallel En translated then filtered and part of news-crawl En translated then filtered.

2.3 Vocabulary

We build a joint subword segmentation model from the synthesized parallel data using SentencePiece (Kudo and Richardson, 2018). The vocabulary size is set to 25,000 tokens. We employ SentencePiece regularization (Kudo, 2018) during data processing. We integrate SentencePiece into the training code and perform subword segmentation on the source side via sampling. Such strategy can improve model quality and robustness.

2.4 Student Model

The standard Transformer with self-attention in decoder has a drawback: decoding complexity increase as the decoding length increases. To address this issue, we refer to some light-weight RNNs, such as SRU (Lei et al., 2018) and SSRU (Kim et al., 2019). Based on our previous experiments and experience, we find that under the teacher-student distillation setting, SSRU models cam basically satisfies the translation quality requirements. As a result, all our student models replace the self-attention layer with SSRU layer on the decoder side. The encoder is still the standard Transformer architecture (Vaswani et al., 2017). We train three sizes of model during our experiment: base, small, and tiny, with different hidden sizes and filter sizes. They all have deep encoders/shallow decoders an architecture capable of increasing speed and maintain quality (Kasai et al., 2021; Wang et al., 2019). All models share the source and target word embeddings and softmax weights.

2.5 Traning

Our distillation experiments are based on fairseq. We also integrate SentencePiece into training. The sampling size is set to 64 and smoothing parameter to 0.1 for subword regularization. All our models are trained using 8 Nvidia Tesla V100 for two days with a batch size of 4096. Because the student models have relatively small capacities, regularization techniques such as dropout and label smoothing are not used. The other parameters use the default fairseq parameters. We save models every 1000 steps and average the last 10 checkpoints to produce the final models.

2.6 Evaluation

We use WMT 2019 and 2020 News Task test sets to measure our models with SacreBLEU (Post, 2018). We use the 12-1 base configuration model as our baseline model, which achieves 38.02 BLEU on 2020 test set, 1.7 BLEU lower than our teacher model. In general, more parameters means better translation quality. The BLEU score of the small.12 model is about 2-2.5 lower than that of the base.12 model, and the BLEU score of the tiny.2 model is also about 2-2.5 lower than that of the small.6 model. For details about parameter settings and BLEU results, see Table 2.

3 Inference Optimizations

For CPU optimization, we use Bolt v1.3.0. Bolt is a standalone open-source deep learning acceleration library. v1.3.0 will be available in September 2021.

3.1 Bolt technical overview

As a universal deployment tool for neural networks, Bolt aims to be faster and lighter. Key features of Bolt include extremely high performance, low-bit inference, widely compatible model converter and low memory usage. Bolt has a standalone C++ runtime, therefore Bolt can perform fast inference without any third-party dependencies. Bolt supports most of the NLP and CV models inference on x86 and ARM CPU as well as MALI GPU. We apply assembly-level optimizations to ensure computing performance and memory accessing efficiency. The operators of Bolt are capable of achieving high throughput near the peak of hardware.

3.2 8-bit Quantization

To accelerate translation tasks on Intel CPU and reduce the model size, Bolt uses linear symmetric quantization (Bhandare et al., 2019) to quantize the weights and part of the activations to 8-bit signed integers. Then Bolt converts the activations to 8-bit unsigned integers by adding 128 because of the limitation of Intel SIMD instructions. To ensure the correctness of matrix multiplication, Bolt applies extra integer offsets which can be obtained offline to the results.

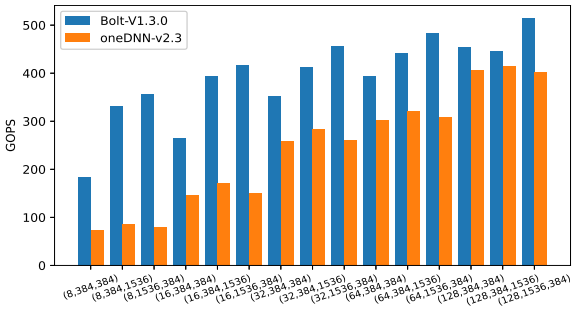


Figure 1: single thread u8s8s32 gemm performance of Bolt v1.3.0 and oneDNN-v2.3 tested on Intel Xeon Gold 6266C CPU, the reported sizes are frequently used in translation task.

The most time-consuming operation of translation tasks is GEMM, Bolt has implemented u8s8s32 gemm kernel, which is faster than Intel oneDNN (MKL-DNN). The u8s8s32 gemm performance of Bolt and oneDNN are shown in Figure 1. We present two key-points of the implementation:

Weights Offline Packing. Assuming the layout of GEMM weights is [N, K], Bolt chunks the weights in the K direction first, and then rearranges the data as NKNxK4 layout, where x is the chunk-size of N direction, x is in {8, 16, 32, 48}. The one that can divide N will be selected.

Highly Efficient Computation. Bolt quantizes bias to 32-bit signed integers, and then adds the offset value obtained offline to bias as the new bias, which is used to initialize the accumulation register of computation for saving addition operations. Bolt uses AVX512 VNNI instructions to perform u8s8s32 matrix multiplication. We have highly optimized the assembly to well utilize the register sources and ensure the instruction efficiency, and we also use memory optimization techniques such as cache-blocking, prefetching and memory alignment. All elements of the product of matrices are 32-bit signed integers. These intermediate data could be efficiently quantized to 8-bit unsigned integers or de-quantized to floating point numbers in registers for the next layer.

Model	Emb.	FFN	Head	Depth	Params(M)	Size(MB)	wmt19	wmt20
Teacher*4	1024	4096	16	25/6	514	2000	46.71	39.70
Base.12	512	2048	8	12/1	53	210	44.65	38.02
Small.12	384	1536	6	12/1	33	132	42.56	35.77
Small.9	384	1536	6	9/1	28	112	42.17	35.73
Small.6	384	1536	6	6/1	22	88	41.15	34.49
Tiny.2	256	1024	4	6/2	13	52	38.92	31.93
Tiny	256	1024	4	6/1	12	48	37.22	30.54

Table 2: Results of Distillation Training. The translation quality deteriorates as the model size decreases: the BLEU score of small model is 2-2.5 lower than that of the base model, and the BLEU score of the tiny model is also about 2-2.5 lower than that of the small model.

3.3 Greedy decoding and Caching

To maximize speed and reduce memory usage, we use greedy search instead of beam search. During decoding, we also skip the final softmax layer and simply get the maximum from the output logits.

Due to the autoregressive model, we also cache the linear transformations for keys and values before the self-attention and cross attention layers.

3.4 Shortlist and Online Quantification

When decoding, we also use the mapping relationship between source and target tokens, a.k.a shortlist, which finds the best matched target token via the source input. Such strategy decreases the dimensions of softmax_weight, which can significantly improve the decoding efficiency while ensuring that the quality is only slightly influenced.

We use fastalign[3] (Dyer et al., 2013) to construct the mapping relationship. During inference, a small target token set is obtained via querying the mapping dynamically, which conflicts with our offline quantization matrix technique. As a result, we try two schemes: a) abandon shortlist and use offline quantification instead; b) keep shortlist and quantify the reduced matrix online. In our experiments, we find that the efficiency of the two versions depends on the size of the target tokens. Because the larger the matrix, the greater the cost of online quantization, and the multiplicative benefits of dimension reduction are offset. For example, on a model we tested, when the input is fixed to 47 tokens, the time cost of a) is 46 ms; the time cost of b) is 68 ms when the size is set to 25000; and the time cost is 48 ms when the size is set to 2000.

Since we focus on the one CPU core latency track, the model processes only one input at a time, and the maximum size of target tokens is less than 2000, so we choose b).

3.5 Submitted Docker images

We choose the base image of ubuntu:18.04. Following the task requirements, our startup script is /run.sh. We use C++ to encapsulate our calls to Bolt and models, SentencePiece, as well as our simple pre- and post-processing. Our model is stored in the /model directory, which contains the converted Bolt model, vocabulary, and shortlist files. The compressed file is provided. Due to the simple runtime environment of Bolt, the final SO package is about 2 MB without any third-party dependency. After quantization, the maximum size of our model is less than 60 MB and the minimum size is about 13 MB. Therefore, the final submitted image is about 100 MB.

4 Optimization results

The latency track we participated in is defined as providing one sentence on standard input and flushing then waiting for your system to provide a translation on its standard output (and flush) before providing the next sentence. So we don't use techniques such as batch. We believe such strategy can better demonstrate the capability our model and inference framework.

After the preceding optimizations, the inference speed is significantly improved. Table 3 lists the results. In general, INT8 inference greatly improves performance. Especially, when the model is relatively large, INT8 improves performance by about three times when comparing with FP32. As the model size becomes smaller, the speed improvement becomes less obvious. But our smallest model also has at least a 2x or above speed improvement. Comparing with that of FP32, the average transla-

[3]http://github.com/clab/fast align

Model	Precious	Size	WPS	BLEU
Base.12	FP32	212	237	38.26
	INT8	53	815	38.02
Small.12	FP32	133	411	36.15
	INT8	33	1158	35.77
Small.9	FP32	112	473	35.90
	INT8	28	1295	35.73
Small.6	FP32	88	550	34.53
	INT8	22	1467	34.49
Tiny.2	FP32	52	759	32.06
	INT8	13	1515	31.93
Tiny	FP32	48	1000	31.01
	INT8	12	2096	30.54

Table 3: Optimization results. The test set is WMT 2020 News Task. The unit of size is MB. WPS refers to the source side. The test environment is Intel(R) Xeon(R) Gold 6278C CPU @ 2.60GH. We submit four models: Base.12, Small.9, Small.6 and Tiny.

tion quality of INT8 models decreases less than 0.1 BLEU.

In our small model setting, the translation quality of the 12-1 model is not significantly improved compared with the 9-1 model, but the inference speed decreases by about 25%. Perhaps under the small model setting, the addition of three encoding layers does not bring significant changes to the model quality. Compared with the Tiny.2 model, the size of our Small.6 model doubles, resulting in an increase of 2.5 BLEU. However, the inference speed are almost the same. In addition, the speed of our Tiny model is 30% faster than our Tiny.2 model by dropping a decoder layer. Our result demonstrates that the number of decoding layers has greater impacts on decoding efficiency. As a result, we submit four models: Base.12, Small.9, Small.6 and Tiny.

The above tests on inference speed are performed with Intel(R) Xeon(R) Gold 6278C CPU @ 2.60GHz.

5 Conclusion

In order to produce a translation system with high inference efficiency, we explore sentence-level distillation techniques and train student models with a trade-off between speed and quality by leveraging Deep-Encoder and Shallow-Decoder models. In terms of inference, we use Huawei Noah's Bolt library. Using a series of optimization techniques, such as INT8 inference and custom efficient GEMM operators, we accelerate inference speed by 2 to 3 times. By using shortlist, greedy search and caching, we submit four models with different settings to the efficiency one CPU core latency task, realizing efficiency improvement under different circumstances.

References

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers.

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.

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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

David R. So, Chen Liang, and Quoc V. Le. 2019. The evolved transformer.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

The NiuTrans System for the WMT21 Efficiency Task

Chenglong Wang[1], Chi Hu[1], Yongyu Mu[1], Zhongxiang Yan[1], Siming Wu[1], Yimin Hu[1], Hang Cao[1], Bei Li[1], Ye Lin[1], Tong Xiao[1,2] and Jingbo Zhu[1,2]
[1]NLP Lab, School of Computer Science and Engineering, Northeastern University
[2]NiuTrans Research, Shenyang, China
clwang1119@gmail.com,huchinlp@gmail.com
{xiaotong,zhujingbo}@mail.neu.edu.cn

Abstract

This paper describes the NiuTrans system for the WMT21 translation efficiency task[1]. Following last year's work, we explore various techniques to improve the efficiency while maintaining translation quality. We investigate the combinations of lightweight Transformer architectures and knowledge distillation strategies. Also, we improve the translation efficiency with graph optimization, low precision, dynamic batching, and parallel pre/post-processing. Putting these together, our system can translate 247,000 words per second on an NVIDIA A100, being 3× faster than our last year's system. Our system is the fastest and has the lowest memory consumption on the GPU-throughput track. The code, model, and pipeline will be available at NiuTrans.NMT[2].

1 Introduction

Large and deep Transformer models have dominated machine translation (MT) tasks in recent years (Vaswani et al., 2017; Edunov et al., 2018; Wang et al., 2019; Raffel et al., 2020). Despite their high accuracy, these models are inefficient and difficult to deploy (Wang et al., 2020a; Hu et al., 2021; Lin et al., 2021b). Many efforts have been made to improve the translation efficiency, including efficient architectures (Li et al., 2021a,b), quantization (Bhandare et al., 2019; Lin et al., 2020), and knowledge distillation (Li et al., 2020; Lin et al., 2021a).

This work investigates efficient Transformers architectures and optimizations specialized for different hardware platforms. In particular, we study deep encoder and shallow decoder Transformer models and optimize them for both GPUs and CPUs. Starting from an ensemble of three deep Transformer teacher models, we train various student models via sequence-level knowledge distillation (SKD) (Hinton et al., 2015; Li et al., 2021a; Kim and Rush, 2016) and data augmentation (Shen et al., 2020). We find that using a deep encoder (6 layers) and a shallow decoder (1 layer) gives reasonable improvements in speed while maintaining high translation quality. We improve the student model's efficiency by removing unimportant components, including the FFN sub-layers and multi-head mechanism. We also explore other model-agnostic optimizations, including graph optimization, dynamic batching, parallel pre/post-processing, 8-bit matrix multiplication on CPUs, and 16-bit computation on GPUs.

Section 2 describes the training procedures of the deep teacher models. Then, Section 3 presents various optimizations for reducing the model size, improving model performance and efficiency. Finally, Section 4 details the accuracy and efficiency results of our submissions for the shared efficiency task.

2 Model Overview

Following Hu et al. (2020), Li et al. (2021a) and Lin et al. (2021a), we use the SKD method to train our models. Our experiments also show that the SKD method can obtain better performance than the word-level knowledge distillation (WKD) method, similar to Kim and Rush (2016). Therefore, all of student models are optimized by using the interpolated SKD method (Kim and Rush, 2016), and trained on data generated from the teacher models.

2.1 Deep Transformer Teacher Models

Recently, researchers have explored deeper models to improve the translation quality (Wang et al., 2019; Li et al., 2020; Dehghani et al., 2019; Wang et al., 2020b). Inspired by them, we employ deep Transformers as the teacher models. More specifically, we train three teachers with different configurations, including *Deep-30*, *Deep-12-768*, and *Skipping Sublayer-40*. We also utilize Li et al. (2019)'s

[1]http://statmt.org/wmt21/efficiency-task.html
[2]https://github.com/NiuTrans/NiuTrans.NMT

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 787–794
November 10–11, 2021.

Student Model	Param.	BLEU
Student-6-6-8	96M	33.2
Student-6-1-8	42M	33.0
Student-6-1-1	42M	32.9

Table 1: Reference BLEU scores for the student models on *newstest20*. 6-6-8 means that the model contains 6 encoder layers and 6 decoder layers with 8 attention heads. Other hyper-parameters are the same as the vanilla Transformer.

ensemble strategy to boost the teachers.

Deep-30 Transformer Model: We set the number of encoder layers to 30 in the Transformer model. Other hyper-parameters are identical to the vanilla Transformer.

Deep-12-768 Transformer Model: This model modifies the number of encoder layers, hidden sizes and embedding sizes to 12, 3072 and 768. Such a setting makes the Transformer model deeper and wider. Other hyper-parameters are the same as vanilla Transformer.

Skipping Sublayer-40 Transformer Model: This model uses a simple training procedure that samples one streaming configuration in each iteration (Li et al., 2021a). The number of encoder layers is 40 and model's other setups are same as Li et al. (2021a).

We adopt the relative position representation (RPR) (Shaw et al., 2018) to further improve the teacher models and set the key's relative length to 8.

2.2 Lightweight Transformer Student Models

Although the ensemble teacher model delivers excellent performance, our goal is to learn lightweight models. The natural idea is to compress knowledge from an ensemble into the lightweight model using knowledge distillation (Hinton et al., 2015). We employ sequence-level knowledge distillation on the ensemble teacher model described in Section 2.1.

Seqence-level Knowledge Distillation The SKD will make a student model mimic the teacher's behaviors at the sequence level. Moreover, the method considers the sequence-level distribution specified by the model over all possible sequences $\mathbf{t} \in T$. Following Kim and Rush (2016), the loss function of SKD method for training students is

$$\mathcal{L}_{\text{SKD}} \approx -\sum_{\mathbf{t}\in\mathcal{T}} \mathbf{1}\{\mathbf{t}=\hat{\mathbf{y}}\}\log p(\mathbf{t}\mid\mathbf{s}) \quad (1)$$

$$= -\log p(\mathbf{t}=\hat{\mathbf{y}}\mid\mathbf{s}) \quad (2)$$

where $\mathbf{1}\{\cdot\}$is the indicator function, $\hat{\mathbf{y}}$ is the output of teacher model using beam search, $\mathbf{s}$ symbolizes the source sentence and $p(\cdot|\cdot)$ denotes the conditional probability. We use the ensemble teacher model to generate multiple translations of the raw English sentences. In particular, we collect the 5-best list for each sentence against the original target to create the synthetic training data. However, we select only 12 million synthetic data to train our student models to reduce training costs. We find that student models will not have better performance when increasing the number of training data.

Fast Student Models As suggested in Hu et al. (2020), the bottleneck of translation efficiency is the decoder part. Hence, we accelerate the decoding by reducing the number of decoder layers and removing multi-head mechanism[3]. Inspired by Hu et al. (2021), we design the lightweight Transformer student model with one decoder layer. We further remove the multi-head mechanism in the decoder's attention modules. Table 1 shows that the Transformer student model with one decoder layer and one decoder attention head can achieve similar translation quality to the baseline. Therefore, we train four different student models based on the Transformer architecture with one decoder layer and one decoder attention head. Those student models are described in detail in the Table 2. Besides, experiments show that adding more encoder layers cannot improve the performance when the student model has 12 encoder layers. Therefore, our submissions have 12 encoder layers at most.

2.3 Data and Training Details

Our data is constrained by the condition of the WMT 2021 English-German news translation task[4], and we use the same data filtering method as Zhang et al. (2020). We select 20 million pairs to train our teacher models after filtering all official released parallel datasets (without official synthetic datasets). The data is tokenized with Moses tokenizer (Koehn et al., 2007), and jointly Byte-Pair

[3] Although the multi-head mechanism does not increase the parameter of the model, it brings non-negligible computational costs.

[4] https://www.statmt.org/wmt21/translation-task.html

Student Model	N-Enc	Dim-FFN	Param.	Speedup	newstest18	newstest19	newstest20
Student-12-1-512	12	512	56M	2.0×	45.3	41.7	33.2
Student-6-1-512	6	512	38M	2.3×	44.5	41.0	32.7
Student-6-1-0	6	0	37M	2.4×	43.9	40.6	32.4
Student-3-1-512	3	512	28M	2.6×	42.8	40.0	31.5

Table 2: N-Enc is the number of encoder layers and Dim-FFN denotes the feed-forward network (FFN) size. The Speedup and BLEU results are measured on a TITAN V GPU. The Speedup is calculated comparing with our ensemble teacher model. The student model has not FFN component in the decoder when the Dim-FFN is 0. Evaluation is performed without inference optimizations and with a beam size of 1.

Teacher Model	Param.	BLEU
Deep-30	138M	32.8
Deep-12-768	170M	33.3
Skipping Sublayer-40	171M	33.1
Ensemble	479M	33.4

Table 3: Results on *newstest20*-Teacher Models. We train our teacher models with the RPR and back-translation.

Encoded (BPE) (Sennrich et al., 2016) with 32K merge operations using a shared vocabulary. After decoding, we remove the BPE separators and detokenize all tokens with Moses detokenizer (Koehn et al., 2007).

Teacher Models Training We train three teacher models using *newstest19* as the development set with `Fairseq` (Ott et al., 2019). We share the source-side and target-side embeddings with the decoder output weights. We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.997$ and $\epsilon = 10^{-8}$ as well as gradient accumulation due to the high GPU memory footprints. Each model is trained on 8 TITAN V GPUs for up to 11 epochs. The learning rate is decayed based on the inverse square root of the update number after 1,6000 warm-up steps, and the maximum learning rate is 0.002. After training, we average the last five checkpoints in the training process for all models. Similar to Zhang et al. (2020), we train our teacher models with a round of back-translation with 12 million monolingual data selected from the News crawl and News Commentary. We train three De→En models with the same method and model setup to generate pseudo-data. Table 3 shows the results of all teacher models and their ensemble, where we report SacreBLEU (Post, 2018) and the model size. Our final ensemble teacher model can achieve a BLEU score of 33.4 on *newstest20*.

Student Models Training The training settings for student models are the same for the teacher models, except its learning rate is 0.0007 and warmup-updates is 8000. In addition, we also use the cutoff method (Shen et al., 2020) to boost our student models[5] and we train our student model with 21 epochs. Table 2 shows the results of all student models. Our student model yields a significant speedup (2×-2.6×) with modest sacrifice in terms of BLEU (0.2-0.9 on *newstest20*).

2.4 Interpretation of Results

After training the final student models, we evaluate their BLEU scores on the English-German *newstest20*, *newstest19*, and *newstest18* before any inference optimization. Results show that the student models can achieve very similar performance to the teachers. For instance, the *Student-12-1-512* model delivers a loss of 0.2 BLEU score compared to the ensemble of teacher models.

3 Optimizations for Decoding

Our optimizations for decoding are implemented with NiuTensor [6]. The optimizations can be divided into three parts, including optimizations for CPUs, GPUs, and device-independent techniques.

3.1 Optimizations for GPUs

For the GPU-based decoding, we mainly explore dynamic batching and FP16 inference.

Dynamic Batching Unlike the CPU version, the easiest way to reduce the translation time on GPUs is to increase the batch size within a specific range. We implement a dynamic batching scheme that maximizes the number of sentences in the batch while limiting the number of tokens. This strategy

[5] https://github.com/stevezheng23/fairseq_extension/tree/master/examples/translation/augmentation

[6] https://github.com/NiuTrans/NiuTensor

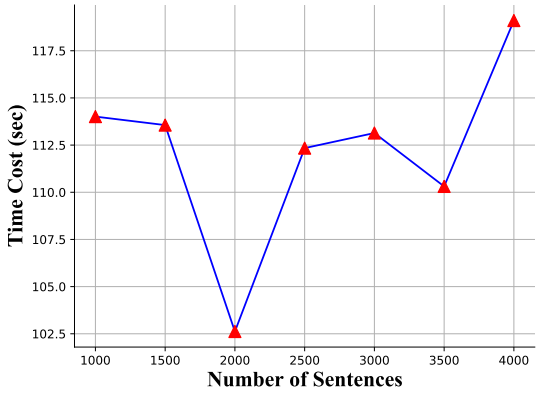


Figure 1: Results on Student-6-1-512 model. The time cost is measured on an Intel Xeon Gold 6240 CPU with 100,000 lines of raw English sentences with an averaged length of 18 words.

significantly accelerates the inference compared to a fixed batch size when the sequence length is short.

FP16 Inference Since the Tesla A100 GPU supports calculations under FP16, our systems execute almost all operations in 16-bit floating-point. To escape overflow, we convert the data type before and after the softmax operation in the attention modules. We also reorder some operations for numerical stability. For instance, we apply the scaling operation (dived by $\sqrt{d_k}$) to the query instead of the attention weights. To accelerate our systems further, we replace the vanilla layer normalization with the L1-norm (Lin et al., 2020). Also, we find that removing the multi-head mechanism (by setting the head to 1) in the student models significantly improves the throughput without performance loss.

3.2 Optimizations for CPUs

We employ the *Student-6-1-512* and *Student-3-1-512* models as our CPU submissions. Two methods are discussed to speed up the decoding for our CPU systems.

The Use of MKL We use the Intel Math Kernel Library (Wang et al., 2014) to optimize our NiuTensor framework, which helps our systems to make the full use of the Intel architecture and to extract the maximum performance.

8-bit Matrix Multiplication with Packing We implement 8-bit matrix multiplication using the open-source library FBGEMM (Khudia et al., 2021). Following Kim et al. (2019), we quantize each column of the weight matrix separately with different scales and offsets. Scale and offsets for weight matrix are calculated by:

$$b_{scale}[j] = \frac{14\sigma_j}{255} \tag{3}$$

$$b_{zeropoint}[j] = \frac{127 - (\bar{x}_j + 7\sigma_j)}{b_{scale}[j]} \tag{4}$$

where σ_j and $\bar{x}_j$ refers to average and standard deviation for the j-th column. The quantization parameters for the input matrix is calculated by:

$$a_{scale} = \frac{x_{max} - x_{min}}{255} \tag{5}$$

$$a_{zeropoint} = \frac{255 - x_{max}}{a_{scale}} \tag{6}$$

where x_{max} and x_{min} are the maximum and minimum values of the matrix respectively. With FBGEMM API, we also execute the packing operation to change the layout of the matrices into a form that uses the CPU more efficiently. We pre-quantize and pre-pack all the weight matrices to avoid repeated operation during inference.

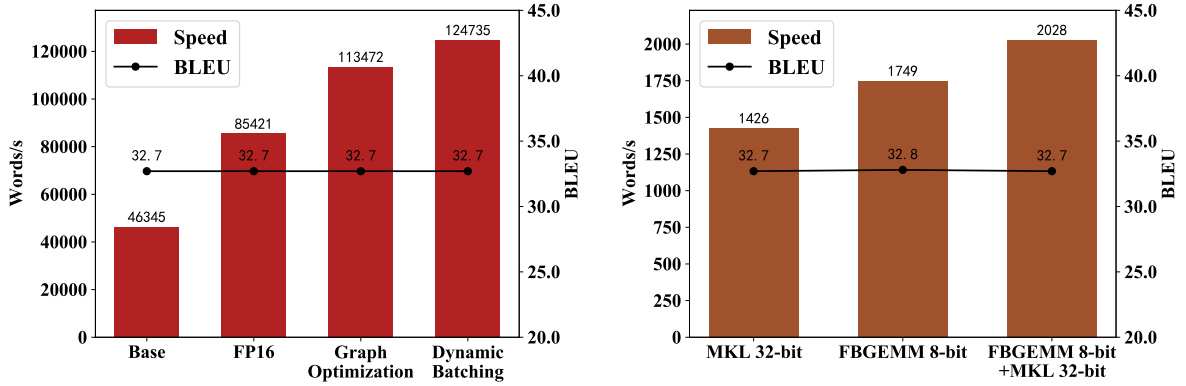
where x_{max} and x_{min} are the maximum and minimum values of the matrix, respectively. We also execute the packing operation to change the layout of the matrices into a form that uses the CPU more efficiently. We pre-quantize and pre-pack all the weight matrices to avoid repeated operation during inference.

3.3 Other Optimizations

Furthermore, we explore other device-independent methods to optimize our systems. Those methods help our systems to achieve obvious speed-up without translation precision loss.

Graph Optimization A neural net can be represented by a directed acyclic graph (DAG), where the nodes represent tensors and the connections represent operations. We optimize our system by simplifying the computational graph of the models. The optimizations for the graph are detailed as follows:

- Computation optimization. We prune all redundant operations and reorder some operations in the computational graph. For instance, we remove the *log-softmax* operation in the output layer when using greedy search. We also extract the *transpose* operations from matrix multiplications to the begin of decoding.



(a) Performance of our GPU system

(b) Performance of our CPU system

Figure 2: BLEU on *newstest20* versus words per second (Words/s) with different optimizations on a TITAN V GPU and Intel Xeon Gold 5118 CPUs. Result of decoding speed is measured with 0.1 M sentences (average length is 18). When the GPU system is running, it will use all free CPUs on the device.

- Memory optimization. We reuse all possible nodes to minimize the memory consumption. We also reduce the memory allocation or movement with an efficient memory pool. Moreover, we sort the source sentences in descending order of length and detect the peak memory footprint before decoding.

Parallel Execution We use the GNU Parallel (Tange, 2011) for our systems to perform tasks in parallel. More specifically, we split the standard input into several lines and deliver them via the pipeline. The method is used to accelerate pre-processing, post-processing, and decoding on CPUs. We also find that the system decoding speed/memory is strongly correlated with the number of lines per task. To find the best number of lines for each run, we measure the time cost in different setups against the number of lines. Figure 1 shows that 2000 is a relatively good choice, and the *Student-6-1-512* model can translate 100,000 sentences in 102.6s on CPUs under this setup.

Better Decoding Configurations As aforementioned, our GPU versions use a large batch size, but the batch size on the CPU is much smaller. To be more clear, there is sentence batch (*sbatch*) and word batch (*wbatch*) in our systems, and they restrict the number of sentences and number of words in a mini-batch to not be greater than *sbatch* and *wbatch*, respectively. In our GPU systems, we set the *sbatch*/*wbatch* to 3072/64000. For our CPU systems, the number of processes is managed by the Parallel tool, which is more efficient and accurate. Moreover, We use one MKL thread for each process and set the *sbatch*/*wbatch* to 128/2048.

Greedy Search In the practice of knowledge distillation, we find that our systems are insensitive to the beam size. It means that the translation quality is good enough even using greedy search in all submissions.

Fast Data Preparation We use the fastBPE[7], a faster C++ version of subword-nmt[8], to speed the BPE process. Moreover, we also use the fast-mosestokenizer[9] for tokenization.

3.4 Results after Optimizations

Figure 2 plots the *Student-6-1-512* model's performance with different decoding optimizations. All results show that our optimizations can significantly speed up our system without losing BLEU. What is interesting about the BLEU is that we can achieve additional improvements of 0.4/0.1 BLEU points on the GPU/CPU through decoding optimizations in all our experiments. We also measure other models after decoding optimizations and find their performance is similar to the *Student-6-1-512* model.

4 Submissions and Results

4.1 Submissions

For the GPU track submissions, our GPU systems are compiled with CUDA 11.2. We set the num-

[7] https://github.com/glample/fastBPE

[8] https://github.com/rsennrich/subword-nmt

[9] https://github.com/mingruimingrui/fast-mosestokenizer

ber of decoder layers and the number of our decoder attention head to 1 as described in Section 2.2 for all our GPU systems. We see a speedup of more than 6× on the GPU system created by *Student-12-1-512* model and a slight decrease of only 0.2 BLEU on the *newstest20* compared to the deep ensemble model. The system is named as ***Base-GPU-System*** in following part. We continue to reduce the number of encoder layers for more accelerations, and the GPU system with *Student-6-1-512* model reduces the translation time by one-four with only six encoder layers compared to the *Base-GPU-System*. Our fastest GPU system consists of three encoder layers and one decoder layer, which achieves 31.5 BLEU on the *newstest20* with GPU and 1.6× speedup compared to the *Base-GPU-System*. We also employ the *Student-6-1-0* model to create a GPU system that can achieve the 1.3× speedup compared to *Base-GPU-System*. Our systems are compiled in the 11.2.1-devel-centos7 docker image, an NVIDIA open-source image[10]. We copy the executables, dependence tools, and model files to the 11.2.1-base-centos7 docker image (final submission). In this way, we ensure all of our system docker images can be executed by the organizers successfully and reduce the docker images size.

For the CPU track submissions, we use the test machine, which has 18 virtual cores. Our CPU version is compiled with MKL static library, and the executable file is 23MiB. Also, we use the 8-bit matrix multiplication with packing to speed the matrix multiplication in the network. We use the *Student-3-1-512* and *Student-6-1-512* models in our CPU systems, and they respectively achieve 31.5 and 32.8 BLEU on *newstest20*. For our CPU docker images, we use the base-centos7 docker image[11] to deploy our CPU MT systems.

Furthermore, all submissions are tested with different cases, including dirty data, empty input, and very long sentences. The test results show that our systems can run successfully with exceptional inputs.

4.2 Results

Our systems for the GPU-throughput track are the fastest overall submissions. Specifically, the *Student-3-1-512* system can translate about 250 thousand words per second and achieve 25.5 BLEU on *newstest21*. We attribute this to the comparison of the performance of our teacher model on WMT21. In the CPU track, our system also has competitive performance. Our fastest CPU system created by *Student-3-1-512* model can translate about 48 thousand words per real second via 36 CPU cores and can achieve 25.5 BLEU. We find that reducing the number of encoder layers for student model achieves lower BLEU scores at a similar speed for our CPU systems. Moreover, we compare the cost-effective of GPU and CPU decoding in terms of millions of words translated per dollar according to the official evaluation results. We find that highly-effective GPU decoding is about to out-compete CPU-bound decoding in terms of cost-effective. Noteworthy, our GPU system with *Student-3-1-512* model can translate 300M words per dollar with acceptable quality. Also, all of our GPU systems have the lowest RAM consumption (about 4 GB) to official test compared with the submissions of other participants.

5 Conclusion

We have described our systems for the WMT21 shared efficiency task. We have explored various efficient Transformer architectures and optimizations specialized for both CPUs and GPUs. We have shown that a lightweight decoder and proper optimizations for different hardware can significantly accelerate the translation process with slight or no loss of translation quality. Our fastest GPU system with three encoder layers and one decoder layer is 11× faster than the deep ensemble model and lose 1.9 BLEU points.

Acknowledgements

This work was supported in part by the National Science Foundation of China (Nos. 61876035 and 61732005), the National Key R&D Program of China (No.2019QY1801), and the Ministry of Science and Technology of the PRC (Nos. 2019YFF0303002 and 2020AAA0107900). The authors would like to thank the anonymous reviewers for their comments and suggestions.

References

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model.

[10] https://hub.docker.com/r/nvidia/cuda
[11] https://hub.docker.com/_/centos

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Chi Hu, Bei Li, Yinqiao Li, Ye Lin, Yanyang Li, Chenglong Wang, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans system for WNGT 2020 efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 204–210, Online. Association for Computational Linguistics.

Chi Hu, Chenglong Wang, Xiangnan Ma, Xia Meng, Yinqiao Li, Tong Xiao, Jingbo Zhu, and Changliang Li. 2021. Ranknas: Efficient neural architecture search by pairwise ranking.

Daya Khudia, Jianyu Huang, Protonu Basu, Summer Deng, Haixin Liu, Jongsoo Park, and Mikhail Smelyanskiy. 2021. Fbgemm: Enabling high-performance low-precision deep learning inference. *arXiv preprint arXiv:2101.05615*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Bei Li, Ziyang Wang, Hui Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2021a. Learning light-weight translation models from deep transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13217–13225. AAAI Press.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 995–1005, Online. Association for Computational Linguistics.

Yanyang Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2021b. An efficient transformer decoder with compressed sub-layers. *CoRR*, abs/2101.00542.

Ye Lin, Yanyang Li, Tengbo Liu, Tong Xiao, Tongran Liu, and Jingbo Zhu. 2020. Towards fully 8-bit integer inference for the transformer model. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3759–3765. ijcai.org.

Ye Lin, Yanyang Li, Ziyang Wang, Bei Li, Quan Du, Tong Xiao, and Jingbo Zhu. 2021a. Weight distillation: Transferring the knowledge in neural network parameters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2076–2088. Association for Computational Linguistics.

Ye Lin, Yanyang Li, Tong Xiao, and Jingbo Zhu. 2021b. Bag of tricks for optimizing transformer efficiency.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

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.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

O. Tange. 2011. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

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 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Endong Wang, Qing Zhang, Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. 2014. Intel math kernel library. In *High-Performance Computing on the Intel® Xeon Phi™*, pages 167–188. Springer.

Hanrui Wang, Zhanghao Wu, Zhijian Liu, Han Cai, Ligeng Zhu, Chuang Gan, and Song Han. 2020a. Hat: Hardware-aware transformers for efficient natural language processing.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Qiang Wang, Tong Xiao, and Jingbo Zhu. 2020b. Training flexible depth model by multi-task learning for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4307–4312, Online. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, et al. 2020. The niutrans machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345.

TenTrans High-Performance Inference Toolkit for WMT2021 Efficiency Task

Kaixin Wu, Bojie Hu, Qi Ju*
TencentMT Oteam
{danielkxwu, bojiehu, damonju}@tencent.com

Abstract

The paper describes the TenTrans's submissions to the WMT 2021 Efficiency Shared Task. We explore training a variety of smaller compact transformer models using the teacher-student setup. Our model is trained by our self-developed open-source multilingual training platform TenTrans-Py[1]. We also release an open-source high-performance inference toolkit[2] for transformer models and the code is written in C++ completely. All additional optimizations are built on top of the inference engine including attention caching, kernel fusion, early-stop, and several other optimizations. In our submissions, the fastest system can translate more than 22,000 tokens per second with a single Tesla P4 while maintaining 38.36 BLEU on En-De *newstest2019*. Our trained models and more details are available in TenTrans-Decoding competition examples[3].

1 Introduction

We participate in the GPU throughput track of the Workshop on Machine Translation (WMT) 2021 Efficiency Shared Task. The efficiency task aims at exploring the different techniques for training and optimizing GPU models for high throughput while preserving the highest possible accuracy. While we do not pay more attention to training techniques, we apply a variety of optimizations to improve the computation efficiency of our GPU models in the inference phase.

In terms of the training phase, we trained a variety of smaller compact student models using the common teacher-student training approach (Hinton et al., 2015; Kim and Rush, 2016) on our open-source multilingual training platform TenTrans-Py. All of them are based on the deep transformer which has proven more effective and has lower training costs than the wide transformer models (Wang et al., 2019). For the inference phase, our strategy for the shared task includes attention caching, kernel fusion, early-stop, and several other optimizations. All of these optimizations are employed in a high-optimized and C++-based inference engine TenTrans-Decoding.

The paper is structured as follows: Section 2 describes the data preparation and the training details, then Section 3 presents the variety of ours optimizations to improve decoding efficiency. The detailed accuracy and efficiency results are shown in Section 4. Finally, we conclude our work in Section 5.

2 Teacher-student Training

To train smaller compact student models, the teacher-student training approach (Hinton et al., 2015; Kim and Rush, 2016) is adopted. First, a large model (the teacher) is trained on all available bilingual data, included synthetic data generated by the back-translation (Sennrich et al., 2015a) method. Multiple model ensembles are also typically used to build stronger teacher systems. Then, all our small optimized models (the student) are created using sequence-level knowledge distillation (Kim and Rush, 2016) and trained on data generated from the teacher model. The sequence-level knowledge distillation is a common technique that has proven successful for reducing the size of neural models, especially in NMT tasks.

2.1 Deep Transformer

Transformer networks (Vaswani et al., 2017) are the current state-of-the-art in many machine translation tasks, and the deep transformer (Wang et al., 2019) which simply stacks more encoder layers has been proved to further enhance the accuracy of the model. To stabilize the training of the deep

* Corresponding author.

[1] https://github.com/TenTrans/TenTrans

[2] https://github.com/TenTrans/TenTrans-Decoding

[3] https://github.com/TenTrans/TenTrans-Decoding/blob/master/examples/WMT21-Efficiency.md

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 795–798
November 10–11, 2021.

Transformer	N_{enc}	N_{dec}	h	d_{model}	d_{ff}	param.	BLEU
Teacher-base-20_6 (2xFFN)	20	6	8	512	4096	160M	39.97
Student-base-20_1	20	1	8	512	2048	88M	39.93
Student-base-10_1	10	1	8	512	2048	58M	39.30
Teacher-tiny-20_1	20	1	8	256	1024	28M	38.36

Table 1: Transformer model configurations and SacreBLEU (Post, 2018) scores on *newstest2019*.

model, we use the Pre-Norm strategy (Wang et al., 2019). The layer normalization (Ba et al., 2016) is applied to the input of every sub-layer which the computation sequence could be expressed as: layer normalization $\longrightarrow$ multi-head attention / feed-forward $\longrightarrow$ residual-add. All of our models are based on deep transformer architecture.

2.2 Teacher & Student Models

The different model configurations for both teacher and student models are presented in Table 1. We train a teacher model and three student model variant with a different number of encoder layers N_{enc}, decoder layers N_{dec}, hidden size d_{model}, and feed-forward network size d_{ff}. We adopt a deep encoder and a shallow decoder architecture of all student models, and the number of decoder layers is set to 1 by default. All of our models tie source embedding, target embedding, and softmax weights.

2.3 Data and Training Details

Dataset Following the shared task setup, we limit our training data to the WMT 2021 English-German translation task. The bilingual data used in the English-German task includes all the available corpora provided by WMT 2021: Europarl v10, ParaCrawl v7.1, News Commentary, Wiki Titles v3, Tilde Rapid corpus and WikiMatrix. For monolingual data, we only use NewsCrawl2020, Europarl v10, and News Commentary for back-translation.

Data preprocessing Then, we normalize punctuation and tokenize all data with the Moses tokenizer (Koehn et al., 2007). For the bitext datasets, we remain sentences no longer than 200 words as well as sentence pairs with a source / target length ratio between 0.3 and 2.0. The fast-align tools (Dyer et al., 2013) are applied to further obtain a cleaned and high-quality parallel corpus. For the monolingual dataset, the sentences with words between 4 and 200 are remained. See Table 2 for details on the bitext and monolingual dataset sizes. After that, we use joint byte pair encodings (BPE) with 32K split operations for subword segmentation (Sennrich et al., 2015b).

	En-De	De (**mono.**)
No filter	49.2M	57.0M
+ length filter	46.9M	55.2M
+ fast-align	41.2M	-

Table 2: Number of sentences in bitext and monolingual datasets for different filtering schemes.

Student training First, we train the teacher model on all available bilingual data, including synthetic data through the back-translation method, and we use English-German *newstest2019* as the development set. We ensemble four best models for building a stronger teacher. Then, the English part of the bilingual data is translated by the teacher model and the resulting synthesized parallel data is used to train the student models. Table 1 shows their evaluation scores on *newstest2019* of different models. The results correlate well with the expectation that more model parameters lead to better performance. Our distillation student models show strong competitiveness even when the number of parameters is greatly reduced.

3 GPU Inference Optimizations

3.1 Implementation: TenTrans-Decoding

TenTrans-Decoding is an open-source high-optimized inference engine for transformer models and the code is written in C++. TenTrans-Decoding's goal is to offer a lightweight and rapid deployment of high-performance service solutions for executing models. All additional optimizations are built on top of the inference engine.

3.2 Attention Caching

We apply the common technique of caching linear projections in Transformer decoder layers. More specifically, we cache the linear transformations for keys and values before cross-attention layers and each step of decoder self-attention layers.

3.3 Kernel Fusion

To reduce kernel launching overhead and enhance the GPU computation efficiency, we implement many kernel fusion techniques for our Transformer models.

- ***Add_bias_residual_layerNormalization*** For the layer normalization between two General Matrix Multiplications (GEMMs), we reorganize the *AddBias* kernel, residual network, and *LayerNormalization* kernel into a single one.
- ***Add_bias_ReLU*** In the Feed-Forward network layers of the Transformer model, the *AddBias* kernel and *ReLU* kernel are fused into one.
- ***Add_bias_residual*** For the output of every encoder or decoder layer, we fuse the *AddBias* kernel and residual network.
- ***Fused_multihead_attention*** In addition to the fusion techniques above, we also fuse the attention layer by packing GEMMs and bias to further improve the computation efficiency.

Figure 1 details the kernel fusion techniques of a transformer decoder layer. The computation graph of a transformer can be reorganized into a more compact graph by fusing all the kernels between two GEMMs into a single one.

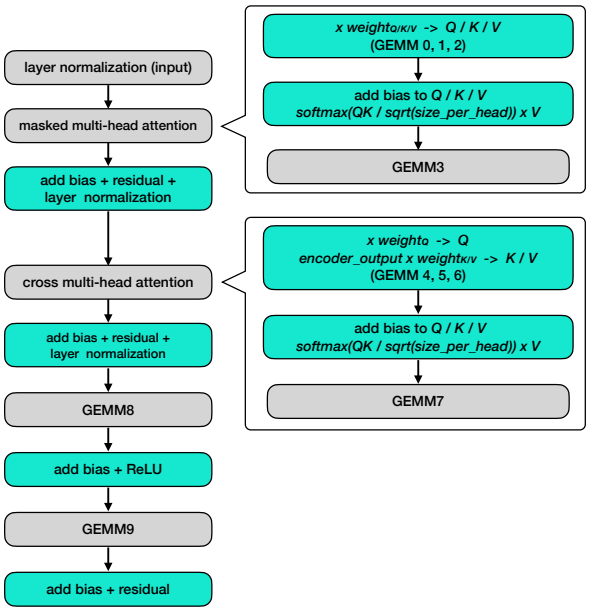


Figure 1: Kernel fusion of a transformer decoder layer. The part in darker color represent using the kernel fusion technique.

	Speed	Ratio	BLEU
TenTrans-Py	696.5	1.00x	38.91
TenTrans-Decoding	1822.4	2.62x	38.91
+ kernel fusion	2565.4	3.68x	38.82
+ early-stop	2682.5	3.85x	38.82
+ sorted batch	5034.8	7.23x	38.98

Table 3: The decoding speed (source tokens per second) and SacreBLEU scores on *newstest2019* for student-tiny-20_1. The speed is measured by a single Tesla P4 GPU and the beam size is 4.

3.4 Early-stop

In batch decoding, the number of decoding ending steps between sentences is different. The early-stop strategy which optimizes kernel function is adopted to avoid redundant computation. For sentences that have been decoded in batch, there is no additional computation for these sentences until the whole batch has been decoded.

3.5 Sorted Batch & Greedy Search

In addition to the methods above, we sort all input sentences from shortest to longest, and the batch size is 128 in our settings. The sorting makes the batches contain sentences of similar sizes which reduces the amount of padding and increases the computation efficiency. During decoding, we use greedy search instead of beam search since we find the distillation model are insensitive to the beam size. We skip the final softmax layer and simply get the maximum from the output logits.

4 Optimization Results

Table 3 shows the impact of different inference optimizations when decoding the Student-tiny-20_1 student transformer model. TenTrans-Decoding leads to a 2.62x speedup than the TenTrans-Py baseline without any inference optimizations. Combine all the inference optimizations mentioned above, it can achieve a 7.23x speedup with no accuracy loss over the baseline.

Table 4 presents all of our submissions and we only participate in the GPU-throughput track. As details in Table 4, we report our model configuration, model size, and metric for translation, including SacreBLEU scores on *newstest2019* and the real translation time cost. All of our systems are tested on a single Tesla P4 GPU. All student models follow a deep encoder and a shallow decoder architecture, the number of decoder lay-

transformer	Model size	Speed (tokens/s)	Ratio	Time Cost (s)	BLEU
Teacher-base-20_6 (2xFFN)	642MB	6274.0	1.00x	9.80	39.97
Student-base-20_1	354MB	12128.1	1.93x	5.07	39.93
Student-base-10_1	234MB	15900.3	2.53x	3.87	39.30
Student-tiny-20_1	113MB	22481.8	3.58x	2.74	38.36

Table 4: Results of all submissions. Time Cost in seconds to translate *newstest2019* and BLEU scores are reported using SacreBLEU. The *newstest2019* contains 1997 sentences. All systems were executed on a single Tesla P4 GPU with greedy search.

ers is 1 by default. All student models training with sequence-level distillation show a competitive performance. The Student-base-20_1 transformer achieves a 1.93x speedup over the teacher baseline with almost no accuracy loss, and the amount of parameters is greatly reduced. Compared with the teacher baseline, the Student-base-10_1 transformer has a speedup of 2.53x times and a slight decrease of only 0.67 BLEU. The Student-tiny-20_1 transformer, our fastest system, which has one-sixth parameters of the teacher model, achieves 38.36 BLEU on *newstest2019* and speeds up the teacher baseline by 3.58x.

In this version, we do not pay more attention to the model size, memory footprint, and low precision inference (e.g., FP16). All operations on the model are based on FP32 floating-point numbers. In the future version, we plan to optimize these points mentioned above.

5 Conclusion

This work presents the TenTrans's submissions to the 2021 Efficiency Shared Task of WMT. We show the deep encoder and shallow decoder student models that training with sequence-level distillation can achieve a competitive performance both in speed and accuracy compared with the teacher baseline. To further improve computation efficiency, we combine several optimizations including attention caching, kernel fusion, early-stop and sorted batch. Finally, our fastest student model achieves a speedup of 3.58x times, while only has one-sixth parameters of the teacher baseline.

In the future, we will apply low-precision inference (e.g., FP16) and more kernel fusion techniques to improve the computation efficiency of our GPU systems. Furthermore, we will continue to explore a more efficient teacher-student training approach to obtain compact student models with competitive performance both in quality and speed.

References

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

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

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 companion volume proceedings of the demo and poster sessions*, pages 177–180.

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

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*.

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.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*.

Lingua Custodia's participation at the WMT 2021 Machine Translation using Terminologies shared task

Melissa Ailem, Jinghsu Liu and Raheel Qader
Lingua Custodia, France
{melissa.ailem,jingshu.liu,raheel.qader}@linguacustodia.com

Abstract

This paper describes Lingua Custodia's submission to the WMT21 shared task on machine translation using terminologies. We consider three directions, namely English to French, Russian, and Chinese. We rely on a Transformer-based architecture as a building block, and we explore a method which introduces two main changes to the standard procedure to handle terminologies. The first one consists in augmenting the training data in such a way as to encourage the model to learn a copy behavior when it encounters terminology constraint terms. The second change is constraint token masking, whose purpose is to ease copy behavior learning and to improve model generalization. Empirical results show that our method satisfies most terminology constraints while maintaining high translation quality.

1 Introduction

Neural-based architectures have become standard for Machine Translation (MT), they are efficient and offer state-of-the-art performance in many scenarios (Vaswani et al., 2017). However, these models often trained on very large corpora turn out to be less adequate in domains that require very careful use of terminology. For instance, consider the following English sentence from a biomedical corpus ***"now for the fever you can take a tachipirina sweet"*** . The term ***"tachipirina sweet"*** refers to ***"paracétamol"*** in French. Unfortunately, a generic English-French Neural MT (NMT) model would translate the above sentence as: ***"maintenant pour la fièvre tu peux prendre un tachipirina bonbon"***, where the term ***"tachipirina sweet"*** is translated ***"tachipirina bonbon"***.

The goal of the WMT21 shared task on machine translation using terminology constraints is to explore methods that can take into account terminology constraints, in order to improve MT models' accuracy and consistency on specific domains. In the literature there are two main families of methods to take into account specific terminologies. One family incorporates terminology constraints at inference (Post and Vilar, 2018; Susanto et al., 2020). Members of this category can guarantee strict enforcement of constraints, however this often comes at the cost of higher decoding time and decreased accuracy (Hokamp and Liu, 2017; Post and Vilar, 2018). The other family of method integrates terminologies at training time (Dinu et al., 2019; Ailem et al., 2021), and they have the benefit of not changing the NMT model as well as of not incurring additional computational overheads at inference time (Crego et al., 2016; Song et al., 2019; Dinu et al., 2019).

We participate in the following three directions: English to French, Russian, and Chinese, and the system we submit falls into the second family of method incorporating terminologies at training time. More precisely, we explore a variant of the models proposed in (Ailem et al., 2021), which we train for each language pair. Following this work, we first annotate our training data with the constraints using tags to distinguish constraints terms from other tokens in the sentences. Second, we further perform constraint-token masking, which improves model robustness/generalization as supported by our experiments.

The rest of the paper is organized as follows: section 2 reviews the details of our system, section 3 describes the training data selection, the development and test sets, as well as the terminologies used for each language pair, and section 4 presents the different experimental settings and results.

2 Method

Our objective is to encourage neural machine translation to satisfy lexical constraints. To this end, we rely on the approch proposed in (Ailem et al., 2021), which introduces two changes to the standard procedure, namely training data augmentation

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 799–803
November 10–11, 2021.

Source	since COVID-19 shows similarities to **SARS-CoV** and MERS-CoV , it is likely that their effect on pregnancy are similar .
Constraints	**SARS-CoV** → **SARS-CoV**
TADA	since COVID-19 shows similarities to <S> **SARS-CoV** <C> **SARS-CoV** </C> and MERS-CoV , it is likely that their effect on pregnancy are similar .
+MASK	since COVID-19 shows similarities to <S> **MASK** <C> **SARS-CoV** </C> and MERS-CoV , it is likely that their effect on pregnancy are similar .

Figure 1: Illustration of TrAining Data Augmentation (TADA) and MASK.

and token masking. In the following we describe these two operations, which are also depicted in Figures 1 and 2.

TrAining Data Augmentation (TADA). The purpose of this step is to encourage the NMT model to exhibit a copy behavior when it encounters constraint terms whose translation should be consistent with some terminology. This step, illustrated in Figures 1 and 2, consists in using tags to annotate our training data with the terminology constraints, i.e., indicate the constraints (if any) in a given source sentence. Note that in the literature, there are other variants that use additional information such as source factors (Dinu et al., 2019). We do not use such information, and we specify terminologies using tags only.

Token MASKing (MASK). We further consider masking the source part of the constraint – tokens in blue – as illustrated in Figure 1 last row. As suggested in (Ailem et al., 2021), this masking strategy provides a more general pattern for the model to learn to perform the copy operation every time it encounters the tag $< S >$ followed by the MASK token. Moreover, this can make the model more apt to support conflicting constraints, i.e., constraints sharing the same source part but which have different target parts. This may be useful in situation in which some tokens must be translated into different targets for some specific documents and contexts at test time.

3 Data

This section provides information and some statistics regarding the datasets for the three language pairs we consider.

Training Data Selection. We consider three language pairs, namely English to French, Russian, and Chinese. Since our method acts at training time, we first perform a training data selection in order to obtain a reasonable number of sentences containing at least one term from the provided terminologies. To do so, we consider both bilingual and monolingual data, provided as part of the shared task. In fact, we observe that bilingual data do not contain many sentences with terminology terms. Thus, we rely on back-translation of monolingual data, which contains more recent news on COVID-19, to obtain more sentence pairs with terminologies. We rely on OpusMT[1] to back translate the Russian monolingual to English. For Chinese and French we use in-house translation engines. Note that we further convert the Chinese data into simplified Chinese using OpenCC. Following previous work on terminology control (Dinu et al., 2019; Ailem et al., 2021), only 10% of the training sentences are annotated in order to maintain the model's performance in terminology free cases. The details about training data selection for the different language pairs are summarized in tables 1, 2 and 3.

Development and Test Sets. For all language pairs, a development and test sets are provided. Note that for the test sets we have access to the source part only. For the dev sets, the terminology constraints associated with each sentence are available, for the test sets this information is not available, and we leverage the terminology files to find constraint terms in these sets. Just like the training data, both test and dev sets are augmented with the terminology constraints as presented in figures 1 and 2. The dev/test sets of the different language pairs share the same English source file containing 971/2100 sentences respectively.

Terminologies. For each language pair, we use the provided terminologies to annotate our train, dev and test sets. The terminologies consist of respectively 670, 925 and 710 unique source-target terms for English → French, Russian and Chinese. We also observe that one source term might be associated with one or more target terms. In that

[1]https://github.com/Helsinki-NLP/Opus-MT

Source	the Canadian government announced CA $ 275 million in funding for 96 research projects on medical countermeasures against COVID-19 , including numerous **vaccine** candidates at Canadian universities , with plans to establish a " vaccine bank " of new vaccines for implementation if another Coronavirus outbreak occurs .
Constraints	**vaccine** → **vaccin**, **vaccines** → **vaccins**, **Coronavirus outbreak** → **épidémie de coronavirus**
TADA	the Canadian government announced CA $ 275 million in funding for 96 research projects on medical countermeasures against COVID-19 , including numerous <S> **vaccine** <C> **vaccin** </C> candidates at Canadian universities , with plans to establish a " <S> **vaccine** <C> **vaccin** </C> bank " of new <S> **vaccines** <C> **vaccins** </C> for implementation if another <S> **Coronavirus outbreak** <C> **épidémie de coronavirus** </C> occurs .
+MASK	the Canadian government announced CA $ 275 million in funding for 96 research projects on medical countermeasures against COVID-19 , including numerous <S> **MASK** <C> **vaccin** </C> candidates at Canadian universities , with plans to establish a " <S> **MASK** <C> **vaccin** </C> bank " of new <S> **MASK** <C> **vaccins** </C> for implementation if another <S> **MASK MASK** <C> **épidémie de coronavirus** </C> occurs .

Figure 2: Illustration of TrAining Data Augmentation (TADA) and MASK (multiple constraints in one sentence).

Data type	#sentences	#term-grounded sentences	Corpora
Monolingual fr	342,941	342,941	News Crawl 2020
Parallel en-fr	3,110,291	110,291	NCv16, UN, Common Crawl, Europarl v10
Parallel en-fr (biomedical)	1,733,757	67,887	EMEA, Medline Titles, Medline abstracts
#Total	5,186,989	521,119	

Table 1: English → French data we use for training.

Data type	#sentences	#term-grounded sentences	Corpora
Monolingual ru	997,889	697,889	News Commentary, News
Parallel en-ru	6,121,064	3,169	News Commentary, Wikititles, ParaCrawl, UN, Wikimatrix, Common Crawl, Yandex
Parallel en-ru (biomedical)	46,782	0	Medline
#Total	7,165,738	701,058	

Table 2: English → Russian data we use for training.

Data type	#sentences	#term-grounded sentences	Corpora
Monolingual zh	899,163	899,163	News Crawl 2020
Parallel en-zh (up-sampled)	12,900	12,900	Wikititles
Parallel en-zh	6,322,275	0	NCv16, ParaCrawl, Wikimatrix, UN, CCMT
#Total	7,234,338	912,063	

Table 3: English → Chinese data we use for training.

case, when annotating the train and dev sets we choose the target term used in the ground truth translation. For the test set, we select one of the possible terms at random.

4 Experimental results

4.1 Settings

For English to French and Russian pairs, we first tokenize the terminology files and the train/test/dev sets before annotating them with the terminology constrains. We use the Moses tokenizer (Koehn et al., 2007) for this step. We then rely on BPE encoding (Sennrich et al., 2015) with 40k merge operations to segment words into subword-units, which results in a joint vocabulary size of 42588 words for English->French, and vocabulary sizes of (44644, 47532) for the (English, Russian) pair. For English->Chinese we rely on sentence piece (Kudo and

Model	BLEU	Exact-Match Accuracy	Window Overlap (2)	Window Overlap (3)	1-TERm	COMET
Transformer	32.12	0.325	0.112	0.114	0.369	0.023
Constrained decoder	40.12	0.856	0.306	0.298	0.535	0.416
TAG+MASK	**44.90**	**0.919**	**0.344**	**0.335**	**0.598**	**0.681**

Table 4: Comparison of different models on the English → French test set.

Language Pair	BLEU	Exact-Match Accuracy	Window Overlap (2)	Window Overlap (3)	1-TERm	COMET
English → French	44.90	0.919	0.344	0.335	0.598	0.681
English → Russian	29.13	0.849	0.247	0.248	0.474	0.604
English → Chinese	29.16	0.829	0.223	0.225	0.437	0.637

Table 5: Results of the investigated system (TAG+MASK) across all the language pairs we consider. Results are obtained using the test set.

Richardson, 2018) for tokenization, which also performs BPE encoding simultaneously and results in a vocabulary size of 52172 for Chinese and 39996 for english. We then annotate the train/test/dev sets with the terminology constraints.

As a building block for our system, we use the transformer architecture (Vaswani et al., 2017) with 6 stacked encoders/decoders and 8 attention heads. For English-French, the source and target embeddings are tied with the softmax layer. We use 512-dimensional embeddings, 2048-dimensional inner layers for the fully connected feed-forward network and a dropout rate of 0.3. The models are trained for a minimum of 50 epochs and a maximum of 100 epochs with a batch size of 2000 tokens per iteration and an initial learning rate of 5×10^{-4}. For each language pair, the validation set is used to compute the stopping criterion. We use a beam size of 5 during inference for all models.

4.2 Results

For all language pairs, the models are evaluated using the standard MT evaluation metrics (BLEU and COMET scores) as well as other terminology-targeted metrics (Anastasopoulos et al., 2021). The latter include the "Exact-Match Accuracy" measure, which simply compute the percentage of constraint terms present in the predicted translations. Although this measure provides an indication of terminology satisfaction, it can only assess whether a term is present in the hypotheses without evaluating whether this target term is correctly placed. To overcome this issue, the authors in (Anastasopoulos et al., 2021) proposed an additional measure, namely "Window Overlap", which computes the percentage of similar tokens surrounding the constraint terms – within a defined window – in the ground truth and the generated hypotheses. Finally, the models are also evaluated in terms of "Terminology-biased TER" score, which is an edit distance based metric (Snover et al., 2006; Anastasopoulos et al., 2021).

We compare the our model TAG+MASK with the traditional transformer baseline (Vaswani et al., 2017) and the constrained decoder approach (Post and Vilar, 2018), which integrates the constraints during inference time. Results on English → French data are presented in table 4. We observe that the TAG+MASK approach significantly improves over baselines in terms of all measures.

Table 5 depicts the results that the submitted system reaches across all the language pairs in terms of different metrics.

5 Conclusion

In this paper, we describe our submission to the WMT21 shared task on machine translation using terminologies. We participate in three language pairs, namely English → French, Russian and Chinese. Our system integrates terminology constraints during training by augmenting the data with terminological terms. Due to the lack of parallel training data containing the terminology terms, we rely on monolingual data for all language pairs to augment the number of sentences containing terminology terms. Empirical results comparing our approach with terminology grounded as well as terminology free baselines show the effectiveness

of the investigated method.

Acknowledgments

This work was partially funded by the French Ministry of Defense.

References

Melissa Ailem, Jingshu Liu, and Raheel Qader. 2021. Encouraging neural machine translation to satisfy terminology constraints. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1450–1455, Online. Association for Computational Linguistics.

Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, Vassilina Nikoulina, et al. 2021. On the evaluation of machine translation for terminology consistency. *arXiv preprint arXiv:2106.11891*.

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*.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3063–3068.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, page 1535–1546.

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.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *Proceedings of NAACL-HLT 2018*, page 1314–1324.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 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: Technical Papers*, pages 223–231.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing nmt with pre-specified translation. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, page 449–459.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with levenshtein transformer. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 3536–3543.

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.

Kakao Enterprise's WMT21 Machine Translation using Terminologies Task Submission

Yunju Bak[1] **Jimin Sun**[2*] **Jay Kim**[1] **Sungwon Lyu**[1] **Changmin Lee**[1]
[1]Kakao Enterprise [2]Carnegie Mellon University
[1]{juliette.y,jay.ka387,james.ryu,louis.cm}@kakaoenterprise.com
[2]jimins2@cs.cmu.edu

Abstract

This paper describes Kakao Enterprise's submission to the WMT21 shared Machine Translation using Terminologies task. We integrate terminology constraints by pre-training with target lemma annotations and fine-tuning with exact target annotations utilizing the given terminology dataset. This approach yields a model that achieves outstanding results in terms of both translation quality and term consistency, ranking first based on COMET in the En→Fr language direction. Furthermore, we explore various methods such as back-translation, explicitly training terminologies as additional parallel data, and in-domain data selection.

1 Introduction

We participate in the WMT21 Machine Translation using Terminologies Task in four language directions, English→French (En→Fr), English→Chinese (En→Zh), English→Korean (En→Ko) and Czech→German (Cs→De).

1.1 Task description

The recent COVID-19 pandemic has raised the urgency to translate and distribute the latest medical information worldwide. However, despite recent advances in neural machine translation (NMT), translation in such emerging domains remains a challenge, as it is unaffordable to collect fair amounts of quality in-domain parallel data in a short time. As an alternative, word- or phrase-level dictionaries of key terms are relatively easier to obtain. These dictionaries are prevalent in commercial settings, where customers specify domain-specific jargon that human translators can attend to. However, incorporating pre-specified dictionaries effectively into NMT models is a non-trivial problem, as NMT is inherently trained without explicit constraints compared to statistical approaches.

In this context, the shared task of Machine Translation using Terminologies is held in five language directions at WMT21. The task assumes a realistic scenario where parallel and monolingual data are abundant in generic domains (e.g., news, web crawl), but only hundreds of word- or phrase-level term dictionaries are available in the domain of interest — COVID-19. Technically, this poses a challenge as we must impose terminology constraints without hurting general translation quality, while only 1.5% of parallel data contain the provided terminologies. Additional issues such as the $1:N$ mapping of term translations further complicate the problem.

Evaluating MT systems in specialized domains diverge from general MT evaluation in that overall translation quality may not ensure the translation accuracy of domain-specific terms. This potential gap calls the need for evaluation metrics that directly assess the consistent use of terms. Concretely, three metrics proposed in Alam et al. (2021) are employed in this task – Exact-Match Accuracy, Window Overlap, and Terminology-biased Translation Edit Rate (TERm). The suggested metrics complement general translation accuracy measured by standard MT metrics (BLEU, chrF, BERTscore, COMET) by validating whether terms are translated faithfully according to the dictionary.

Specifically, human-labeled COVID-19 related term dictionaries are released in four language directions (En→Fr, En→Zh, En→Ko, En→Ru), with around 600 terms for each direction. Exceptionally, the dictionary for Cs→De is constructed automatically and consists of 5,601 parallel terms.

1.2 Related work

Word- or phrase-level constraints have often been introduced to NMT via constrained decoding to reinforce specific tokens in the output sequence.

*Work done during the author's internship at Kakao Enterprise.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 804–812

Combined with terminology dictionaries, constrained decoding integrates the target side terms as decoding-time constraints (Hokamp and Liu, 2017; Anderson et al., 2017; Post and Vilar, 2018).

Subsequent work has shown that adding inline annotations to the source sentence as soft constraints can improve performance and time complexity when employed with additional source factor streams (Dinu et al., 2019; Bergmanis and Pinnis, 2021). Similarly, a merging approach by adding markers without modifying the model has also proven to be effective (Wang et al., 2019).

2 Data

2.1 Cleaning

Both monolingual and parallel corpora of all languages are preprocessed according to the following pipeline. First, we remove non-utf8 or non-printable characters. Second, we unescape HTML characters such as `>`. Finally, we normalize variations in spaces and punctuation marks. All cleaning steps are done with Moses scripts (Koehn et al., 2007). We also use the Moses tokenizer, but only for European languages (En, Fr, Cs, De) since Asian languages (Zh, Ko) require language-specific tokenizers that consider the characteristics of each language.

2.2 Filtering

Web-crawled data are notorious for being noisy. To prevent defective data from undermining performance, we filter both parallel and monolingual data with diverse methods.

Bi-text We filter the provided parallel data with several heuristics. We first eliminate pairs that contain empty lines or identical content in both source and the target side. We filter pairs that contain overly long sentences (250 words) or excessively long words (50 characters). The pairs that have a word count ratio larger than four are also omitted. We refer to previous literature to set statistical thresholds of each rule. Lastly, we only use pairs of which both sides are identified as the correct language with a language identification tool. Specifically, we use fastText (Joulin et al., 2016, 2017).

In addition, for En→Ko, we filter out mislabeled bi-text which we found manually, that seemed as byproducts of web-crawl in the source or target side. For instance, the pattern "*YYYY년 MM월 DD일에 확인함*", which means "*Confirmed in YYYY/MM/DD*", was found instead of the correct labels in 20,909 samples. The final dataset sizes are shown in Table 1.

	En-Fr	En-Zh	En-Ko	Cs-De
Parallel	158M	62M	13M	15M
+ Filter	149M	-	12M	13M

Table 1: Dataset sizes of parallel corpora before and after filtering in each language pair. For En-Zh, we did not apply rule-based filtering.

Mono-text We used monolingual text for two language pairs (En→Ko, En→Fr) to augment existing parallel corpora via back-translation (Sennrich et al., 2016a). The back-translation procedure is described in Section 3.2.

For En→Ko, we do not apply any filtering schemes as the size of the Korean monolingual corpus is small (14M sentences).

On the contrary, for En→Fr, using the entire French monolingual corpora (8.5B) for back-translation is unwieldy, considering the time and computation required to infer all samples. Hence, we filter the corpus and select in-domain, COVID-19 related data to maintain a reasonable size for inference and training.

We filter French monolingual data in three steps. First, we roughly filter the data with rule-based methods that are similar to those of bi-text filtering. Second, we choose sentences that contain terms in the terminology dictionary (8.5B → 725M). Lastly, we use the Moore and Lewis (Moore and Lewis, 2010) method to find samples that are more similar to the term-related samples. Specifically, we train an in-domain language model with sentences that contain terminologies from the En-Fr parallel corpus. A general-domain language model is also trained with samples chosen randomly from the En-Fr parallel corpus. For both models, we use KenLM (Heafield, 2011) to train 5-gram language models with modified Kneser-Ney smoothing. Finally, top-k sentences with the highest scores are chosen (725M → 160M).

3 Approaches

3.1 Baseline

We explore two baseline approaches that differ by their training data. First, models are trained with

solely the parallel data described in 2.2. This baseline does not utilize the terminology dictionary.

Second, we take a naïve approach to leverage the term dictionary – including the provided terms as additional parallel data to train the model. For $1:N$ mappings of term translations, we flatten them into N distinct pairs. We refer to this approach as the "explicit" model in the following sections as we "explicitly" augment the training dataset with terminology dictionaries.

3.2 Back-translation

We incorporate back-translated monolingual data for two language directions: En→Fr and En→Ko.[1] We train reverse translation models (Fr→En, Ko→En) with the same parallel corpora and training configuration used to train our baseline models covered in Section 4.2. Back-translated samples are inferred with beam search of beam size 4, and a length penalty of 0.6.

For En→Fr, we use back-translated corpora for Exact Target Annotation fine-tune. We revisit the details of this procedure in Section 3.4.

For En→Ko, we train the back-translation model from scratch using both parallel and back-translated text. During training, we upsample the parallel corpus twice as frequently as the back-translated text.

3.3 Target Lemma Annotation

To integrate terminology constraints, we employ Target Lemma Annotation (**TLA**) of Bergmanis and Pinnis (2021), which helps the model learn how to copy-and-inflect inline annotations. At training time, we randomly select target lemmas and inject them into the source sentence behind the corresponding source word(s).

Specifically, we adopt a simple approach where we modify the input data but not the model. This differs from the method described in Bergmanis and Pinnis (2021), which uses additional input streams to denote the annotated tokens. In detail, we introduce three special tokens **<b>**, **<t>**, and **</t>** which respectively indicate the start of annotated source tokens, the start of target lemma tokens and the end of target lemma tokens. An example is shown in Table 2.

Following the training data annotation procedure of Bergmanis and Pinnis (2021), we first lemmatize and mark part-of-speech tags of the target sentences, using spaCy (Honnibal et al., 2020) instead of the pre-trained Stanza model (Qi et al., 2020) due to the time complexity. We then obtain word alignments using fast_align (Dyer et al., 2013) and randomly annotate verbs or nouns with their corresponding target lemma. To set annotation thresholds, we refer to Bergmanis and Pinnis (2021) – [0.6, 1.0) for sentence-level and [0.0, 1.0) for word-level. The annotated and original data are fed into the model with a proportion of 1:1.

At test time, we provide soft terminology constraints by annotating source terms with their corresponding target terms retrieved from the terminology dataset. Terminology entries are identified with the longest word-sequence match in the source sentence. If there exist several target terms for one source term, we randomly select one candidate.

Original Source	EN	and are you having any of the following **symptoms** with your chest pain?
Annotated Source	EN	and are you having any of the following **<b> symptoms <t> symptômes </t>** with your chest pain?
Target	FR	et avez-vous l'un de **symptômes** suivants en plus de vos douleurs thoraciques ?

Table 2: An example of using special tokens for inline annotations. Inline annotations are marked in bold. **<b>**, **<t>**, **</t>** denote the start of the annotated source tokens, the start of the target lemma tokens, and the end of the target lemma tokens.

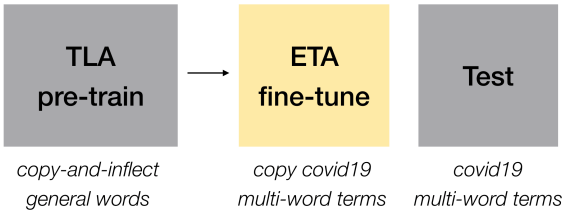


Figure 1: The steps in our TLA pre-train → ETA fine-tune approach and the objective of each phase.

3.4 Exact Target Annotation (Fine-tune)

We adopt Exact Target Annotation (ETA) designed by Dinu et al. (2019) to fine-tune the TLA model pre-trained as in Section 3.3. ETA injects the exact target-side translation of a terminology entry into the source sentence using inline annotations. Note that we utilized the whole terminology dataset during training, unlike Dinu et al. (2019), since the task allows the use of the terminology dataset at

[1] We do not incorporate back-translated corpora of Cs-De and En-Zh due to time constraints.

training time.

While TLA learns to copy-and-inflect general words, our terminology dataset is domain-specific. We aim to fill the domain gap by constructing fine-tuning data in which terminology entries are present on both the source and target sides. As a result, 750K samples from the parallel data and 10M samples from the back-translated data are selected. We upsample the parallel corpus by eight times.

Another discrepancy between training and test time annotation in TLA is that TLA engages a single target word to the corresponding source word(s), whereas many of the actual terms are multi-word expressions in both source and target sides. We expect ETA fine-tune to alleviate the problem since ETA annotates target terms in verbatim. The pretrain-finetune phases are outlined with their motivation in Figure 1.

Specifically, we follow the annotation strategy of Dinu et al. (2019), where we annotate only when both the source side term t_s and the target side term t_t are present. When a sentence contains multiple matches overlapping each other, we keep the longest match.

The difference between Dinu et al. (2019) and our method is that we annotate with three special tokens as described in Section 3.3. Instead of randomly deciding whether to annotate or not, we annotate all matches. We then combine the annotated data with its original data and use it for training with a proportion of 1:1. The annotation procedure at test time is also equivalent to Section 3.3.

4 Experiments

4.1 Evaluation setting

Evaluation of the models is done using the evaluation script [2] and the development dataset, both provided by the task organizers. We select the best models by considering all metrics provided by the evaluation script.

For evaluation, we tokenize our outputs so that they resemble the tokenization setup of the development dataset. For En→Fr and Cs→De, we use the Moses toolkit (Koehn et al., 2007). For En→Zh, we apply the Jieba tokenizer.[3]

Before submitting the test set translations, we handle rare target-side tokens decoded as <unk> by simple substitutions, which we found to work well during evaluation even without incorporating external methods such as word alignments. When the number of <unk> tokens are equal on both sides, we copy the original source-side tokens to the target slots in the same order. After replacing rare tokens, outputs are detokenized using the Moses toolkit (Koehn et al., 2007).

4.2 Experimental details

For En→Fr and Cs→De, we pre-tokenize the data using the Moses toolkit (Koehn et al., 2007). We use sentencepiece (Kudo and Richardson, 2018) to learn a joint byte pair encoding (BPE) with vocabulary size 40K (En→Fr) and 32K (Cs→De). For En→Ko, We pre-tokenize Korean sentences with Mecab (Kudo, 2005) without space tokens as suggested in Park et al. (2021) and use sentencepiece to learn a BPE model with vocabulary size 32K for each language side. For En→Zh, we first convert characters possibly in traditional Chinese to simplified Chinese text using hanziconv [4] and. Then, we pre-tokenize the data using the Jieba tokenizer [3]. We then use subword-nmt (Sennrich et al., 2016b) to train BPE on combined Chinese and English corpus and build separated vocabularies. The final vocabulary size is 44K for Chinese and 32K for English.

For all language directions, we employ the Transformer architecture (Vaswani et al., 2017) implemented in fairseq (Ott et al., 2019). The specific training and generation configurations can be found in Appendix A.

Since TLA relies on the word-aligner's performance, we did not apply TLA pre-training and ETA fine-tuning for En→Ko and En→Zh. Given that both are linguistically distant language pairs, we assumed that the word-aligner's performance would not be sufficient enough to guarantee improvements from TLA.

We start ETA fine-tuning from the TLA checkpoint saved at 750,000 steps for En→Fr and 200,000 steps for Cs→De, chosen based on BLEU scores and Exact Match Accuracy. To evaluate the TLA and ETA fine-tuned models, we run annotation using the terminology tags provided with the development dataset, which is different from the test annotations described in 3.3.

For En→Ko and Cs→De, we use an ensemble of models that utilize back-translation, explicit training, and data augmentation. The exact ensemble

[2] https://github.com/mahfuzibnalam/terminology_evaluation

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

[4] https://github.com/berniey/hanziconv

System	BLEU	Exact Match	Window Overlap (Window 2/3)	1-TERm
En-Fr				
Baseline	47.83	0.882	**0.31/0.301**	0.628
TLA	47.07	0.915	0.282/0.275	0.611
TLA w/o annotation	47.84	0.881	0.305/0.297	0.617
TLA + ETA fine-tune (bi-text only)	47.47	**0.932**	0.298/0.289	0.615
TLA + ETA fine-tune	**48.16**	0.929	0.307/0.30	**0.631**
En-Zh				
Baseline	29.08	0.803	**0.192**/0.194	0.418
Explicit	**29.81**	**0.805**	**0.192/0.197**	**0.431**
En-Ko				
Baseline	12.04	0.412	0.038/0.037	0.129
Baseline + BT	14.14	0.417	0.039/0.037	0.172
Explicit	12.27	0.42	0.034/0.032	0.151
Explicit + BT	14.24	**0.464**	0.04/0.04	**0.184**
Ensemble	**14.56**	0.454	**0.043/0.042**	0.178
Cs-De				
Baseline	30.95	0.832	0.41/0.398	0.434
Explicit	30.77	0.833	0.408/0.396	0.433
Ensemble	**32.47**	0.848	**0.429/0.416**	**0.445**
TLA	28.46	**0.924**	0.281/0.272	0.395
TLA + ETA fine-tune (bi-text only)	30.14	0.889	0.353/0.342	0.417

Table 3: Evaluation results for each task language pair. Highest scores are **boldfaced**. Rows in gray indicate our submitted systems for test evaluation.

configurations are detailed in Appendix B.

5 Results

Table 3 reports the evaluation results of the four language pairs that we participated in.

5.1 English→French

The TLA model improves Exact Match Accuracy but shows deteriorated performance on all other metrics compared to the baseline. Notably, the degradation stems from the test-annotation method – test scores are comparable to the baseline when tested with raw text (without test-annotation) on the same TLA model.

On the other hand, under the same test-annotation condition, the ETA fine-tuned model recovers the performance loss and even boosts the BLEU score, Exact Match Accuracy, and the 1-TERm score compared to both the baseline and the TLA model. TLA + ETA fine-tune outperforms the baseline by 0.33 points, 4.65%, and 0.24% on BLEU, Exact Match, and 1-TERm, respectively.

In addition, we run a simple ablation experiment by using only bi-text data during ETA fine-tuning: TLA + ETA fine-tune (bi-text only). The results are indistinguishable from the original TLA + ETA fine-tune, which is fine-tuned with data from both bi-text and mono-text. This result supports that the performance gain stems not only from the use of monolingual data, which was unseen during TLA pre-training.

5.2 English→Chinese

We compare two approaches – baseline and explicit, and observe that adding the term pairs explicitly to training improves both general translation performance (+0.73 BLEU) and term consistency (+2.29% 1-TERm) compared to the baseline.

5.3 English→Korean

Back-translation yields performance gains across all metrics with considerable improvements, particularly in BLEU and 1-TERm. The explicit model also brings modest improvements to Exact Match

Language	COMET			Exact Match Accuracy			Number of
	Ours	Best	Rank	Ours	Best	Rank	Submissions
En-Fr	**0.781**	-	1	0.95	0.974	4-6	22
En-Zh	0.229	0.716	8	0.645	0.886	7-8	8
En-Ko	**0.581**	-	1	**0.569**	-	1	1
Cs-De	**0.694**	-	1	0.866	0.871	1-2	2

Table 4: Official task results of our submitted systems. Scores, where our system ranked 1st, are bold-faced. In other cases, the best scores from other submissions are shown for comparison.

Accuracy and 1-TERm. Finally, our ensemble model that combines these approaches demonstrates the best performance across all metrics, raising the BLEU score by 2.52 points, Exact Match Accuracy by 4.2%, Window Overlap by 0.43% and 0.54% for windows 2 and 3 respectively, and 1-TERm by 4.88 points.

5.4 Czech→German

We discover that the explicit model does not bring significant gains compared to the baseline model. This trend contradicts other language directions, where we observed at least modest improvements over their respective baselines. We suspect the differences lie in how the terminologies are generated; Cs→De terminologies are constructed automatically, whereas, for other language directions, the terminologies were annotated manually.

Our ensemble model improves upon the baseline model by 1.5 BLEU points, 1.6% Exact Match Accuracy, 1.84% and 1.74% Window Overlap for window sizes 2 and 3, and 1.1 points in 1-TERm.

We also attempted to apply TLA pre-training + ETA fine-tuning to Cs→De as done in En→Fr. In our preliminary experiments, while some metrics improved, we observed Exact Match Accuracy deteriorate after 1,000 steps of TLA training, unlike En→Fr, possibly due to the automatic creation pipeline of Cs→De terminologies. Therefore, we did not further explore this direction during our task participation. However, subsequent experiments after the deadline revealed that TLA, when followed by ETA fine-tuning, has its advantages in finding a balance between BLEU and Exact Match Accuracy, supporting our findings in En→Fr.

5.5 Official task results

We present our official submission results in Table 4. Despite the trade-off between general translation quality (COMET) and term consistency (Exact Match Accuracy), our approach strikes at the right balance between the two criteria for En→Fr. Out of 22 submissions in this direction, our system ranks 1st in COMET. According to Exact Match Accuracy, our system performs roughly comparable to the best system, ranking 4-6th. For En→Zh, our system ranks 8th in both metrics out of 8 submissions. For En→Ko, our submission is the only submission. For Cs→De, our submission ranks 1st in terms of COMET and 1st-2nd for Exact Match Accuracy out of 2 submissions.

6 Conclusion

We participate in four language directions for the shared task WMT21 Machine Translation Terminologies. To this end, we explore various techniques, including back-translation, explicitly training with term pairs along with other parallel data, and in-domain data selection to improve translation performance in the COVID-19 domain.

In particular, for En→Fr and Cs→De, we find that TLA outperforms the baseline in terms of Exact Match Accuracy by leveraging terminology constraints. However, all other metric scores (BLEU, 1-TERm) plummeted, implying that the overall translation quality was compromised. We recover this performance loss by introducing a new technique – fine-tuning with ETA, and achieve significant improvements in both general translation quality and terminology consistency. We leave it to future work to validate our approach in other languages and reveal the factors behind the benefits of ETA fine-tuning precisely, hopefully, to discover a more suitable design to impose terminology constraints.

References

Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp

Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency. *CoRR*, abs/2106.11891.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. 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.

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.

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. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás 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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. 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 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Takumitsu Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Sungjoon Park, Jihyung Moon, Sung-Dong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Young kuk Jeong, Inkwon Lee, Sang gyu Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Alice H. Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021. Klue: Korean language understanding evaluation. *ArXiv*, abs/2105.09680.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer 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. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tao Wang, Shaohui Kuang, Deyi Xiong, and António Branco. 2019. Merging external bilingual pairs into neural machine translation.

A Training configuration

```
fairseq-train
  task : translation
  arch : transformer_wmt_en_de_big
  lr : 0.0005
  lr-scheduler : inverse_sqrt
  warmup-updates : 4000
  warmup-init-lr : 1e-07
  optimizer : adam
  adam-betas : (0.9, 0.98)
  update-freq : 8
  dropout : 0.1
  weight-decay : 0
  criterion : label_smoothed_cross_entropy
  label-smoothing : 0.1
  fp16 : True

fairseq-train (ETA fine-tune)
  lr : 1e-06
  lr-scheduler : fixed
  warmup-updates : 0

fairseq-generate
  beam : 4
  lenpen : 0.6
```

B Ensemble Configuration

For En→Ko, we use an ensemble of four models trained with different configurations:

- Baseline + Back-translation
- Baseline + Back-translation + Rule-based filtering
- Baseline + Back-translation + Explicit
- Baseline + Back-translation + Explicit (Parallel corpus upsampling with ratio 2)

For Cs→De, we use an ensemble of four models trained with different configurations. The third model concatenates the previous and next sentence for additional context with probability of 0.1:

- Baseline
- Baseline + Rule-based filtering
- Baseline + Two sentences concatenation (0.1)
- Baseline + Explicit

The SPECTRANS System Description for the WMT21 Terminology Task

Nicolas Ballier[1] **Dahn Cho**[1] **Bilal Faye**[1] **Zong-You Ke**[1] **Hanna Martikainen**[2] **Mojca Pecman**[1]
Jean-Baptiste Yunès[3] **Guillaume Wisniewski**[4] **Lichao Zhu**[4] **Maria Zimina-Poirot**[1]
[1] CLILLAC-ARP / [3] IRIT / [4] LLF, Université de Paris, F-75013 Paris, France
[2] CLESTHIA, Université Sorbonne Nouvelle - Paris 3, F-75005 Paris, France
{nicolas.ballier, guillaume.wisniewski,jean-baptiste.yunes,
maria.zimina-poirot}@u-paris.fr, mpecman@eila.univ-paris-diderot.fr
{dcho0501, biljolefa, zongyou.ke.fr}@gmail.com
hanna-julia.martikainen@sorbonne-nouvelle.fr

Abstract

This paper discusses the WMT 2021 terminology shared task from a "meta" perspective. We present the results of our experiments using the terminology dataset and the OpenNMT (Klein et al., 2017) and JoeyNMT (Kreutzer et al., 2019) toolkits for the language direction English to French. Our experiment 1 compares the predictions of the two toolkits. Experiment 2 uses OpenNMT to fine-tune the model. We report our results for the task with the evaluation script but mostly discuss the linguistic properties of the terminology dataset provided for the task. We provide evidence of the importance of text genres across scores, having replicated the evaluation scripts.

1 Introduction

In our (traditional) sense, terminological databases are the collection of specialised lexical resources that are generally compiled from corpora, in collaboration with experts from the field, then analysed and structured according to the type of information recorded in term records: terms, equivalents, definitions, synonyms, contexts of use, and related terms (hyperonyms, hyponyms, meronyms, holonyms, etc.). The data thus created are empirical and provide knowledge-based representations of the domain (especially in the case of an ontological approach), underlining conceptual links between terms that can be observed (like meronomy: "X is a part of Y") and potentially represented in conceptual graphs.

For instance, the ARTES database (Pecman and Kübler, 2011), used at Université de Paris in Masters studies for teaching terminology management to translation students (Kübler et al., 2018), adopts such a comprehensive approach to terminology, with specific attention to emerging terminology and complex noun phrases (CNPs) (Kübler et al., 2021). In recent works combining studies on terminology, specialised translation and corpus linguistics, attention has been drawn to CNPs in English which have been demonstrated to cause major difficulties during translation, both human and machine (Kübler et al., 2021; Maniez, 2017). Moreover, studies have demonstrated an increase of complex compounding in specialised texts in English over the last few decades, with, for instance, an overwhelming use of patterns with adjectival and participial compound pre-modifiers (e.g. *receptor-binding activity, electron-dense aggregates*) (Mestivier-Volanschi, 2015).

For this WMT21 Terminology workshop, we focused on the linguistic properties of the terminological dataset provided. We selected what we believe to be the two best models we produced for the EN-FR track with two different neural toolkits but we mostly took the opportunity to discuss the addition of terminology to neural machine translation. The rest of the paper is organised as follows: section 2 summarises our approaches to the task, section 3 presents the tools we used and how we used the constrained data. Section 4 presents our experiments and the best models we used for the translation challenge. Section 5 discusses our results.

2 Our Approaches to the Task

This section presents our various strategies for the terminology task.

2.1 Toolkit Comparison

We compared the predictions of two toolkits. We trained two systems, JoeyNMT (Kreutzer et al., 2019) and OpennMT (Klein et al., 2017) with comparable parameters, using Europarl as baseline, later supplemented with the terminology resource provided for the task.

2.2 Model selection and fine-tuning

With OpenNMT only, we selected the training data, comparing the performance with and without the

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 813–820

terminological data for CommonCrawl and Europarl and applied fine-tuning to the model based on Europarl enriched with the terminological data.

2.3 Comparing with pre-trained models

We were curious to see how pre-trained models fared on this tasK. We produced two translations, one based on `mBART-50` (Tang et al., 2020) and the other one on the `Hugging Face` (Wolf et al., 2019) baseline. We finalised them after the evaluation deadline, so that we report our findings on the sacreBLEU score we calculated with the Systran translation used as reference. Debatable as it may sound to use an MT-generated reference translation, this enabled us to run comparisons.

2.4 A linguistic analysis of the terminology resource and evaluation script

We focused our analysis on the linguistic properties of the terminology provided and tested. We also tried to test other models we produced after the competition deadline, which is why we detail the evaluation script we tried to replicate and the terminology in the next section.

3 Data and Tools Used

This section presents the datasets used to build our system as well as our replication of the evaluation script to analyse the models we did not have the time to submit.

3.1 Training Data

The first challenge lies in the data selection for the training corpora among the possibilities of the challenge. We did not resort to specific texts such as the TICO-19 data (Anastasopoulos et al., 2020) but used the Europarl corpus as baseline.

3.2 Terminological Data

This subsection provides a linguistic qualitative approach to the provided terminology dataset.

A potential problem with the terminology dataset is variation. While some variants are probably interchangeable in most texts (e.g. 220 *hand sanitizer: gel hydroalcoolique* | *désinfectant pour les mains*), others present different degrees of specialization (e.g. 345 *multi-organ failure: défaillance multi-viscérale* | *défaillance de plusieurs organes*). For yet other variants, both forms are possible, but not within the same text for coherence (e.g. 286 *SARS-CoV-2: SRAS-CoV-2* | *SARS-CoV-2*, where the first variant is the translated acronym and stands for *syndrome respiratoire aigu sévère*).

The French-English terminological resource included 595 "terms" out of which only 181 were tested in the script so that the achievement rate as testd by the evaluation scripts only relies on 30.42% of the resource provided. Many entries in the dataset are not actually terms, but rather out-of-context strings or keyword combinations that are impossible to translate since, in translation, context truly is everything. Strings such as (154) *covid-19 WHO* and (158) *covid19 CDC* are not actual NPs and are rarely found as such, on their own, in real texts. In context, these n-grams are always followed by additional information that needs to be taken into account in their translation (e.g. *Covid-19 WHO Situation Report* or *Covid-19 CDC Info*). [1] Therefore, the proposed translations (respectively, *OMS et covid-19* and *CDC et covid19*), where the different elements are simply linked with the conjuntion *et* cannot work in context since the components of the actual NP would need to be reorganized in translation when unpacking the informational content in these CNPs. Other examples of out-of-context keyword combinations in the dataset are entries 112 *covid-19 dangerous*, 113 *covid-19 deadly*, 116 *covid19 domestic travel*, and 128 *covid19 international travel*. The role of Complex Noun Phrases seems to be underestimated in the terminology resource, as well as collocations. Nouns are more frequent than adjectives and verbs in the provided resource. 143 adjective + noun collocations are proposed (such as *deadly virus*) for 13 adjectives. Only 19 verbal collocations are proposed for eight verbs.

Beyond the immediate textual context, lack of real-world context is also a potential source for incorrect translation. For entries 245 *n95*, 246 *n95 mask*, and 247 *n95 respirator*, the proposed translations all use the N95 classification, which is the US NIOSH standard. For real texts, functionally adequate translation might require, for instance, using the equivalent European classification (*FFP2*). Dataset entry 246 presents an additional real-word related issue: *N95 respirators* should not be referred to as "masks", as their airborne-particle filtration capacity is far superior to those of surgical

[1] With hindsight, setting values at n=2 or n=3 for Window Overlap Accuracy was consistent with "truncated" sequences such as *covid-19 WHO* but *Covid-19 WHO Situation Report* and similar embedding structures would only be captured by the Window Overlap Accuracy metric when n=4 or more.

masks which serve a different purpose (reducing outward particle emission).

3.3 Data for Fine-Tuning

We re-trained our generic model by selecting the presumed best candidates for training sets. To specialize the model and make it more efficient, after having trained it on Europarl, we chose a method to select texts that are closer to the terminological data. Several similarity measurement methods are possible. In this case we worked with cosine similarity, which is more sensitive to the number of occurrences of terms in each corpus. After having carried out the similarity measurement of all the texts with the test data, we retained 1/4 of the files, corresponding to 22,741,561 sentences. These selected texts served as a corpus of re-training of our model for its specialization. Compared to the constrained corpora proposed for training, our optimised selection of texts based on the cosine similarity with the testing set corresponded to the following subsampling of the proposed corpora: 1 % of News Commentary v16 and 99 % of 10^9 French-English Corpus. From a purely machine learning perspective, using testing sets to figure out training sets may sound unusual, but it should be borne in mind that we do not aim here at generalisability but at performing a specific task (translating biomedical texts).

3.4 Replicating the evaluation script

We did not have the time to submit our translations based on fine-tuning and pre-trained systems, so that we tried to replicate the evaluation script[2]. Our script[3] is a modification of the procedure described in (ibn Alam et al., 2021) that includes 1-TER but not COMET. It allows the calculation of the following scores: Exact-Match accuracy, Window Overlap (2), Window Overlap (3), sacreBLEU, TER and TERm. The calculation, unlike the evaluation made by the competition, is done here segment by segment and the average of the set of results makes it possible to detail scores per segments. The preprocessing is the same as on the reference script (tokenization and lemmatization) and the removal of parentheses on the corpus is necessary to run it. It is limited to 1,371 segments, for which the term to be translated was identified with certainty. As a result, one section of the testing data was not considered (the email sent to the wikipedia collaborators, referred to as "email" in our text genre analysis).

[2]https://github.com/mahfuzibnalam/terminology$_evaluation$

[3]To be found on https://github.com/nballier/SPECTRANS/tree/main/WMT21

4 Experiments and Results

4.1 Training with JoeyNMT

For comparison purposes, we used the baseline of JoeyNMT which is based on TRANSFORMER (Vaswani et al., 2017) and requires lighter implementations. It took the `Europarl 7` parallel corpus as data set, split as follows: training (341,554 sentences), dev (50,000 sentences) and test (100,000). The data set has been preprocessed with a two-level tokenization: standard tokenization (Spacy) segments data into words and BPE tokenization (*SentencePiece* (Kudo, 2018)) into sub-words. Our model was trained with the following parameters: vocabulary size: 32, 000, maximum sentence length: 50, maximum output length: 100, training optimizer: `ADAM`, normalization: tokens, training model initializer: `XAVIER`, encoder embedding dimension: 512, decoder embedding dimension: 512, hidden size: 512. The best BLEU score from English to French (Figure 1) was achieved at 32.04 at step 41 000 with a training rate of 18 seconds per 100 steps, whereas the best French to English BLEU score was 31.35. By comparing JoeyNMT translation with OpenNMT translation, we notice that JoeyNMT had poor results in translating dates, numbers, proper nouns, acronyms and symbols. Sentences which have several of those may have been translated into a string of characters of repeated sub-words. The translation submitted could not be scored but for BLEU (5.29). The result came as a surprise to us since JoeyNMT has the same model architecture as OpenNMT (Transformer). Because of these issues, we only conducted the other experiments with OpenNMT.

4.2 Training and Fine-tuning with OpenNMT

We used the baseline of OpenNMT-tf 2.20.1 based on `TRANSFORMER` (Vaswani et al., 2017). The parallel data Europarl v10 (Koehn et al., 2005) containing 1,911,202 aligned sentences pairs was used as a dataset, which was divided into two subsets: training set (1,906,202 sentences) and evaluation set (5,000 sentences). The dataset was preprocessed with a BPE tokenization using SentencePiece into subword units (32,000 subword units as training vo-

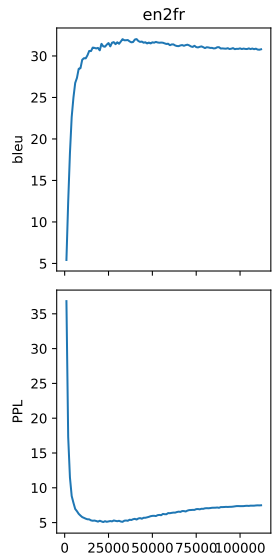


Figure 1: JoeyNMT : BLEU score and PPL score (en-fr)

cabulary). The model was trained with the following parameters: vocabulary size: 31,000, learning optimizer: LazyAdam. The best BLEU score from English to French was 43.90 after 70,000 steps with a training rate of 1.18 steps per second.

We then produced a model with Europarl adding the terminology to the training data with the same evaluation data. As a comparison, we also tried to produce a model with Common Crawl corpus [4] using the same parameters of SentencePiece and training. The dataset consists of 3,244,152 aligned sentences pairs split into training set (3,239,152 sentences) and evaluation set (5,000 sentences). This model produced the best scores in among our submissions (0.871 for Exact-Match Accuracy).

For fine-tuning, we used the Europarl model enriched with the terminological data. We were not able to use the onmt-update-vocab command, so that instead we directly replaced the dictionary file in the configuration with the dictionary based on the files described in section 3.3. Contrary to our expectations, the fine-tuning did less well for scores, according to our estimations (see Table 1). Not being able to update the dictionary in fine-tuning might be responsible for worsening the quality of our results.

4.3 Pre-trained Systems

For a point of comparison, we considered two Transformer-based models available in the Hugging Face library (Wolf et al., 2019). The first one is the standard *pipeline*[5] for English to French translation. The second one is based on the multilingual language model `mBART-50` (Tang et al., 2020), fine-tuned for multilingual machine translation as described in (Tang et al., 2020). The two models were applied on the raw sentences extracted from the SGM files of the test data. The sole preprocessing that was applied consisted in replacing XML entities by their corresponding characters and applying the tokenizer considered by the model. While the translation for the PUBMED section is satisfactory, the translation of the CMU section revealed issues in the use of subjunctive (ie segment 20). It should be noted that, according to our homemade evaluations, these models did much better for sacreBLEU scores (+3.7) and Hugging Face is slightly higher than the Corpus Crawl data trained with the terminology resources (the two models are superimposed on Figure 3.

4.4 Replicating the scoring system with the different translations

Because we could not submit all our translations in time, we resorted to a proxy for evaluation by adapting the available scripts to produce our own evaluation scripts. Our sacreBLEU (Post, 2018) score was based on the SYSTRAN translation used as a reference text. We used the SYSTRAN generic Pure Neural Server (Crego et al., 2016). We show how our scoring system (dots) compares to the official evaluation system (crosses) in Figure 2. We tend to be less generous for Exact-Match Accuracy and more optimistic for Window Overlap Accuracy (with n=3). It should be noted that our reference translation, although mostly accurate, also presents some problems. These occur mainly in the incomplete out-of-context segments related to patient symptom descriptions, many of which are also ungrammatical (ie segment 4). Table 1 recaps the scores we obtained for all the models we produced. For the models we submitted in time, as could be expected, the model trained with Common Crawl and the terminological resources (+ Term in our table) got better scores than Europarl supplemented with the terminological resources. For our in-house evaluation, we tested the translations produced by these models as well, so that we could

[4] https://commoncrawl.org/

[5] Pipelines are Hugging Face abstractions for NLP tasks that automatically select the 'correct' model architecture and all the related components (such as the tokenizer) required to make a prediction

submitted model	BLEU (truecased)	Exact-Match Accuracy	Window Overlap Accuracy (n=2)	Window Overlap Accuracy (n=3)	1-TERm Score	COMET
Common Crawl + Term	40.02	0.871	0.296	0.296	0.507	0.596
Europarl + Term	34.93	0.795	0.275	0.267	0.495	0.296
Europarl (baseline)	33.59	0.640	0.248	0.241	0.480	0.212
in-house scores	**sacreBLEU**	**Exact-Match Accuracy**	**Window Overlap Accuracy (n=2)**	**Window Overlap Accuracy (n=3)**	**1-TERm Score**	**1-TER score**
Hugging Face	**32.21**	0.73	0.32	**0.324**	0.36	0.37
mBART	30.46	0.707	0.296	0.294	0.35	0.36
Common Crawl + Term	28.50	**0.77**	0.299	0.306	0.30	0.308
Europarl + Term	23.74	0.68	0.258	0.256	0.293	0.303
Europarl (baseline)	17.98	0.53	0.18	0.17	0.24	0.25
Europarl (fine-tuning)	26.19	0.68	0.279	0.278	0.278	0.287
joeyNMT (Europarl)	4.67	0.16	0.039	0.034	0.045	0.064

Table 1: Summary of our official and home-made scores for our models

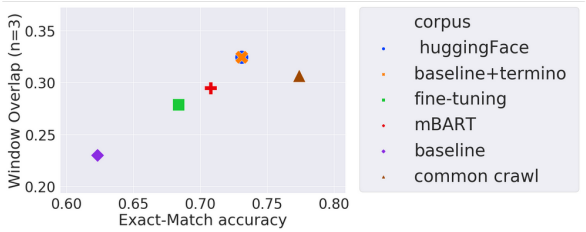


Figure 2: Comparison of the scores for the three SPECTRANS models submitted)

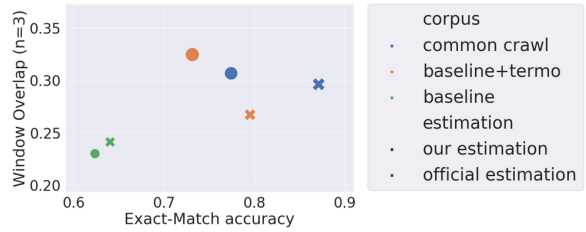


Figure 3: Comparison of the best models acccording to our scores)

compare them to the translations produced by the pre-trained models (`Hugging Face` and `mBART`. The latter did better for sacreBLEU and Window Overlap Accuracy (n=3) but probably having seen the terminological resources in the training data gave an edge for Exact-Match Accuracy to our model trained with Common Crawl and the terminological resources.

5 Discussion

5.1 Variability Across Text Genres

The benefit of our recreation of the evaluation script is that it allowed us to compute the terminology scores for 1,430 segments. We grouped the different sections of the test data according to text genres, in fashion similar to (Anastasopoulos et al., 2020). We distinguished 5 groups of texts and the variability of the BLEU scores across these text genres can be seen on Figure 4. This variability across text genres can also be seen for some other metrics, such as Window Overlap accuracy (with n=3) (see Figure 5).

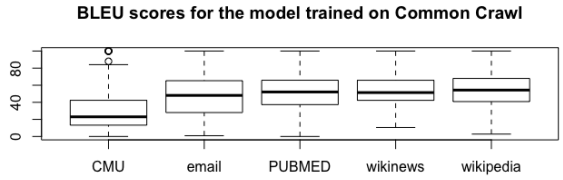


Figure 4: BLEU scores and text genres (Common Crawl training)

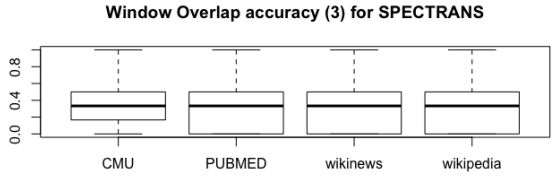


Figure 5: Variability of Window Overlap accuracy (n=3) across text genres

Overall, it is likely that our results could have been better if we had used alternative testing sets rather than using part of the reference corpora as testing sets.

5.2 Alternative qualitative terminological analysis

This subsection discusses the error analysis in terminology from a qualitative point of view.

For CNPs not included in the terminology dataset such as *chest pain*, the system deploys various avoidance strategies ranging from anatomic approximations (segment 20: *mal de coeur*) to omission (segment 8: *Et cette douleur est-elle bien réelle?*) to unlucky guesses (segment 2: *maux de mer*) to idiomatic expressions (segment 18: *C'est bien là que le bât blesse*). For less formal descriptions of similar symptoms where the actual term does not appear in the source text, the output

type	segments	acronyms	terms	acronym/segment	terms/segment
CMU	104	0	71	0.000	0.683
PUBMED	676	465	622	0.688	0.920
wikinews	67	11	25	0.164	0.373
email	98	7	7	0.071	0.071
wikipedia	1,155	315	929	0.273	0.804

Table 2: Distribution of acronyms in the text data

ranges from gibberish and hallucinations (segment 25: *c'est comme si la grenna est écrasée* to soaring lyricism (segment 97: *C'est la peine que j'ai sur le cœur*). When confronted with an unorganized list of terms such as the one in segment 30 (*anyone in the family have a heart problem heart disease heart attack high cholesterol high blood pressure*), most of which are not included in the dataset, the system valiantly tries to make sense of it by turning it into a complete sentence: *Quiconque au sein de la famille est confronté à un problème cardiaque, s'attaque à la pression sanguine élevée en raison de la forte pression sanguine*.

For key terminology around Covid-19, the preferred option in the output is the masculine form (*le/du/au Covid*: 127 occurrences) that is also massively present in the terminology dataset, whereas the feminine *la Covid* only appears 9 times in the output. Interestingly, in only one of these occurrences (segment 2124) does the feminine form appear within the CNP recorded in the dataset (*virus de la COVID-19*). In the other segments, it appears as a translation for the simple term COVID-19, which is in the dataset invariably associated with the masculine form when the gender is specified.

For the compound key term (56) *coronavirus disease*, different solutions appear in the output alongside the proposed translation from the dataset (*maladie du coronavirus*). One erroneous solution in our output is *maladie des coronavirus*. The plural form is problematic, as several coronaviruses exist indeed and most of them are linked with the common cold, with presents a very different picture from the illness provoked by the new coronavirus having emerged in 2019. An interesting solution appears in our output for segment 186:

[EN] The outbreak of Coronavirus disease 2019 (COVID-19), caused by severe acute respiratory syndrome (SARS) coronavirus 2 (SARS-CoV-2)

[FR] L'apparition de la maladie liée au coronavirus 2019 (COVID-19), causée par les syndromes respiratoires aigus sévères (SARS) Le coronavirus 2 (SARS-CoV-2)

The proposed translation, i.e. *la maladie liée au coronavirus 2019 (COVID-19)*, is actually a better choice than the one included in the dataset. The system seems to have achieved this translation by linking *disease* and *illness*, as the translation for *coronavirus disease* appears to draw from that given for *covid19 illness* in the terminology dataset. For the second CNP in this segment, *severe acute respiratory syndrome (SARS) coronavirus 2 (SARS-CoV-2)*, however, the proposed translation is less accurate, specifically in terms of syntax. This example also contains one of the few occurrences of the short form *SARS-CoV-2* in our output (16 in total, most with no article). The preferred option in our output is the translated acronym *SRAS-CoV-2*, with 153 occurrences, of which 143 also have a definite article (*le/du/au*).

5.3 Presence of acronyms in the terminological data

Medical terms in each segment involve two forms: acronyms and fully spelled form. The semantic fields covered by these terms include medical products ("face masks," "vaccine"), biochemical elements ("virus"), diseases ("COVID", "SARS"), as well as public health practices ("quarantine"), organizations ("WHO") and phenomena ("outbreak"). For any segment that contains at least one medical term of either form, the term count of the corresponding form is set to 1 for the segment. Counts and ratios per segment for each of the five types of documents are calculated. It can be observed from the above table that type PUBMED has the highest ratio per segment for either form of medical terms (0.688 for acronym and 0.920 for normal form), while type EMAIL has very low ratios especially for normally spelled form (0.071). In terms of medical term density, differences among these types of documents are therefore distinct. Table 2 sums up our findings in terms of the presence of acronyms

5.4 Terminology better at inference time?

We entered the challenge following the track for using the terminological resources at training time. We nevertheless did a background check on the possibilities of using the provided dataset at inference time. We plan to experiment the SYSTRAN Model studio functionalities to test the performance of using the terminology resource at inference time.

5.5 A Case for Onto-terminology?

The terminology provided for this task was unstructured, contrary to existing ontologies for medical English. Taking advantage of ontology-oriented programming in Python as implemented in Owlready (Lamy, 2017), it is tempting to consider potential implementations of onto-terminology in python-based neural translation toolkits. Biomedical ontologies have a record of established terminologies. One of the added benefits of this line of investigation is that we could not only test the gains of a structured ontology at training time but we could try to implement sanity checks at inference time to ensure the quality of the terminology by making sure the position of the terms in the output is consistent with the hierarchy in the ontology.

6 Conclusion

This paper presents the SPECTRANS system description for the WMT21 Terminology Shared Task. We participated in the English-to-French task, using the terminology resources at training time. Though English–French is a language-pair with many linguistic resources, we only used the data provided by the organisers. Given the novel evaluation of terminology provided for this task, we not only aimed to build a translation system for the competition, but also to provide a critical angle on the task and on its evaluation. For the MT system, we applied a variety of strategies, toolkit comparison, data augmentation and fine-tuning. Though we did not experience catastrophic forgetting, our fine-tuning did less well in the terminology metrics, probably because we were not able to update the dictionary. We obtained the best scores for the models we submitted with a model trained with Common Crawl supplemented with the terminology resource. The translations produced with pretrained models competed in terms of terminology scores, did better for sacreBLEU, especially for the translation of PUBMED, but proved less robust for the translation of the patient-doctor interactions of the CMU section of the testing data.

For the analysis of the terminology, we discussed the role of complex noun phrases and initialisms. Our contribution mostly lies in the critical analysis of the terminological input and of the evaluation script. This allowed us to raise the issue of the role of acronyms in the terminology, the importance of complex NPs (and the correlative interest of the Window Overlap Accuracy with n=3, more likely to capture complex NPs than Window Overlap Accuracy when n= 2) as well as the importance of text genres.

Acknowledgements

The SPECTRANS project is funded under the 2020 émergence research project, under the ANR grant (ANR-18-IDEX-0001, Financement IdEx Université de Paris). Lichao Zhu and Guillaume Wisniewski collaborate in this paper in the ambit of a Regional innovation programme, under the Ile-de-France DIM RFSI 2020 funding programme NeuroViz. This publication has emanated from research supported in part by a 2021 research equipment grant from the Scientific Platforms and Equipment Committee (PAPTAN project) and Masters students Internship grants to Bilal Faye and Zong-You Ke with the financial support of data intelligence institute of Paris (diip), both under ANR Grant Number ANR-18-IDEX-0001 (Financement IdEx Université de Paris).

References

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Md Mahfuz ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn et al. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Natalie Kübler, Alexandra Mestivier, and Mojca Pecman. 2018. Teaching specialised translation through corpus linguistics: quality assessment and methodology evaluation by experimental approach. *META : Journal des traducteurs / Meta: Translators' Journal*, 63(3):806–824.

Natalie Kübler, Alexandra Mestivier, and Mojca Pecman. 2021. Using comparable corpora for translating and post-editing complex noun phrases in specialised texts: Insights from english-to-french in specialised translation. *S. Granger and M-A. Lefer (eds.), Extending the scope of corpus-based translation studies*.

Jean-Baptiste Lamy. 2017. Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*, 80:11–28.

François Maniez. 2017. Evaluation des récentes avancées de la traduction automatique : le cas des adjectifs composés formés à partir d'un participe passé en anglais de spécialité. *Asp*, 72:29–48.

Alexandra Mestivier-Volanschi. 2015. Productivity and diachronic evolution of adjectival and participial compound pre-modifiers in english for specific purposes. *Fachsprache*, XXXVII(1-2):2–23.

Mojca Pecman and Natalie Kübler. 2011. Artes: an online lexical database for research and teaching in specialized translation and communication. In *Proceedings of the First International Workshop on Lexical Resources*.

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

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

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 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Dynamic Terminology Integration for COVID-19 and other Emerging Domains

Toms Bergmanis[†‡] and **Mārcis Pinnis**[†‡]

[†]Tilde / Vienības gatve 75A, Riga, Latvia
[‡]Faculty of Computing, University of Latvia / Raiņa bulv. 19, Riga, Latvia
{firstname.lastname}@tilde.lv

Abstract

The majority of language domains require prudent use of terminology to ensure clarity and adequacy of information conveyed. While the correct use of terminology for some languages and domains can be achieved by adapting general-purpose MT systems on large volumes of in-domain parallel data, such quantities of domain-specific data are seldom available for less-resourced languages and niche domains. Furthermore, as exemplified by COVID-19 recently, no domain-specific parallel data is readily available for emerging domains. However, the gravity of this recent calamity created a high demand for reliable translation of critical information regarding pandemic and infection prevention. This work is part of WMT2021 Shared Task: *Machine Translation using Terminologies*, where we describe Tilde MT systems that are capable of dynamic terminology integration at the time of translation. Our systems achieve up to 94% COVID-19 term use accuracy on the test set of the EN-FR language pair without having access to any form of in-domain information during system training. We conclude our work with a broader discussion considering the Shared Task itself and terminology translation in MT.

1 Introduction

This work is part of WMT2021 Shared Task: *Machine Translation using Terminologies*, which is concerned with improving machine translation (MT) accuracy and consistency on *newly developed domains* by utilising word and phrase-level terms. We describe Tilde MT systems that are capable of dynamic terminology integration at inference time. Our submissions consist of translations by terminology-enabled general-purpose MT systems for EN-RU, EN-FR, and CS-DE translation directions. Our systems are deliberately trained without consideration for the test domain to follow the spirit of the Shared Task—MT for emerging domains. Despite term collections being noisy, our MT systems with dynamic terminology integration improve term translation accuracy proving their usefulness in dynamic adaptation for novel domains, where training-time domain adaptation methods are not feasible.

The remainder of this work describes the methods used for dynamic terminology integration (Section 2) by describing the tasks of terminology filtering, term recognition, and dynamic terminology integration in the translation process. The bulk of Section 2 describes problems due to the low-quality term collections and terminology mismanagement and our solutions to them. We hope that the examples provided will not only illustrate the self-imposed problems by the Shared Task but also will motive reconsidering the purpose and the desired qualities of a term collection in the context of MT. We then briefly describe the experimental setting and results in Section 3 and Section 4 respectively. We conclude our work with a broader discussion considering the Shared Task and terminology translation in MT in Section 5.

2 Methods

This section describes the three tasks necessary for successful MT with terminology: terminology filtering, term recognition, and finally, integration of terminology constraints in the translation process.

2.1 Terminology Filtering

To guarantee terminology translation correctness and consistency, which are two quality aspects of terminology translation, term collections must provide unambiguous information about the preferred translation equivalent for each source term's type (full form, short form, or acronym) when listing multiple possible translation equivalents in a tar-

*Both authors have contributed equally.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 821–827
November 10–11, 2021.

	#	Source Term	Target Side of a Term Entry
EN-RU	1	hand-washing	мытье рук | мытья рук | мытья
	2	sneeze	чихании | чихания | чихнуть
	3	flu epidemic	эпидемия гриппа | эпидемия свиного гриппа
EN-FR	4	World Health Organization	Organisation mondiale de la Santé | *Organisation Mondiale de la Santé* | *Organisation mondiale de la santé*
	5	Coronavirus outbreak	épidémie de coronavirus
	6	*coronavirus outbreak*	épidémies de coronavirus
CS-DE	7	zimní paralympijské hry	Paralympische Spiele | Sommer-Paralympics | Paralympische Winterspiele | Paralympische Sommerspiele | Paralympics
	8	letní paralympijské hry	Paralympische Spiele | **Sommer-Paralympics** | Paralympische Winterspiele | Paralympische Sommerspiele | Paralympics
	9	narcista	Narzissmus | *Narzißmus* | narzisstisch

Table 1: Examples of noisy term entries in the provided term collections. Inflected forms are blue, wrong translations of the source term are red, alternative spelling variants are *italic*, other possible translations are **bold**, other terms within one term entry are underlined.

get language. Typically preparing such term collections that are useful for MT (or the translation process in general) is the task of a professional translator or a terminologist. In this Shared Task, however, term collections sometimes contain entries that do not feature reciprocal translations of terms (e.g., see examples 1, 3, 7, and 8 in Table 1), have multiple translations per term (example 8 in Table 1), have multiple different terms merged into a single term entry (examples 2 and 9 in Table 1), have different spelling forms listed on the target side (examples 4 and 9 in Table 1), and have different inflections or spelling variants of source terms separated in different term entries (examples 5 and 6 in Table 1). Besides, unlike the common custom of providing terms and their translations in their dictionary forms (e.g., see examples of terms in EuroTermBank[1], the InterActive Terminology for Europe[2], the United Nations Terminology Database[3], and other authoritative term banks), the terminologies provided for the Shared Task often contain entries where the source or the target language form is already inflected (examples 1, 2, and 6 in Table 1).

To reduce the noise present in the provided term collections, we performed filtering by discarding:

1. Term pairs that feature terms consisting of symbols other than digits, letters, apostrophes, white-spaces, and hyphen symbols. This filter allows to identify and discard expressions that do not represent terminology (e.g., full sentences, complete clauses, formulas, expressions consisting of terms and their acronyms within one term entry, etc.; see examples 1-6 and 8 in Table 2).

2. Term pairs where the source term is longer than the target term and the source term contains the target term as a sub-string (or vice versa). This filter is intended to discard term entries representing named entities that are written identically in both source and target languages, but for which one of the sides is incomplete (see examples 7 and 9 in Table 2).

3. Term pairs that represent general language (i.e., are too common). General language phrases are typically ambiguous and may require different translations based on surrounding context as well as external knowledge, which may not be available when translating. Therefore, it may be safer to allow the NMT model to handle the translation of general language phrases. We also do not want to burden the MT model too much with excessively annotated input data since longer segments are typically handled worse by NMT models than shorter segments (Neishi and Yoshinaga, 2019). To identify term entries that are too general, we apply an inverse document frequency (IDF) (Jones, 1972) filter (Pinnis, 2015a). As an example, this filter discarded from the term collections all term entries of the English term "spread" as it is a highly ambiguous word and according to the Collins EN-FR dictionary[4] it may have at least 20 distinct translations . Since the term collection

[1] https://eurotermbank.com/
[2] https://iate.europa.eu/
[3] https://unterm.un.org/
[4] https://www.collinsdictionary.com/dictionary/EN-FR/spread

features just one possible translation without any added meta-data, it is safer to not use such terms (considering also the limitations of term recognition when working with emerging domains with scarce or no parallel data).

Besides filtering out the noisy term entries, the type and the quality of the term collections provided in the Shared Task also require selecting one among potentially many term translation equivalents. As noted before, this is typically done by a human, possibly, a domain expert. Nevertheless, we opt for two different strategies. If more than one translation equivalent is provided, it is fair to assume that they are all equally applicable. Thus we propose to select the first translation equivalent in the list. We refer to it as **1st Trg Term** term selection strategy. After analyzing the term collections, however, we conclude that it is not the case that all translation equivalents provided in the Shared Task are of equal quality. Therefore, we employ a statistical word alignment-based strategy to select the translation equivalent with the highest alignment score. To compute word alignments we use *eflomal*[5] (Östling and Tiedemann, 2016). We refer to it as **Alignment-based** term selection strategy.

Table 3 gives examples of terms selected by either of the term selection strategy. Examples illustrate that some translation equivalents are of equal quality (examples 1, 3, 6, and 7). However, selecting the first translation equivalent can sometimes give long (examples 6 and 8) or inadequate (example 4) translation equivalents. The Alignment-based term selection strategy also tends to select translation equivalents that are dictionary forms (example 2) instead of inflections.

2.2 Term Recognition

Having a term collection, the next task in the MT workflow is term recognition in a running text. Term recognition depending on the morphological typology of the source language and the nature of the domain can prove to be a complex task. Recognition involves term identification in its surface form, which for morphologically complex languages may be hindered by many surface forms a single word can take or by the level of form ambiguity in the case of morphologically impoverished languages (Bergmanis and Goldwater, 2018). To overcome issues posed by the morphology of the natural language, one can use one of the many off-the-shelf morphological taggers to obtain contextually correct part-of-speech and lemma pairs for each token and perform term recognition on lemmatized collections and texts. We, however, opt for an alternative, a more rudimentary method utilizing language-specific stemmers to normalise the surface forms and do the term recognition on stemmed running text and term collections (Pinnis, 2015a,b). We opt for the stemmer-based approach because, in the production setting, stemming is faster than morphological tagging and has broader coverage for low-resource languages. Besides, to take full advantage of the morpho-syntactic information provided by morphological taggers, similar information must be provided by the term collection. However, as term collections of this Shared Task exemplify, expecting any meta-data is naive.

Last but not least, recognised forms must be word-sense disambiguated if more than one translation sense (i.e., term entry per source-side lexical form) is available. Word-sense disambiguation tools typically are lexicalized classifiers that are trained using large amounts of parallel data. However, the spirit of this Shared Task is MT using terminology for emerging domains where "*parallel data are hard to come by*"[6]. Thus we skip word-sense disambiguation and use just one word-sense per word form.

2.3 Integration of Terminology Constraints

In the day-to-day work of professional translators, terminologies are glossaries containing source language terms and their corresponding target language translations in their dictionary forms. Some of the previous work on terminology translation assumes that term entries are given in forms already inflected as required by the target morpho-syntactic context. Thus, such work focuses either on morphologically impoverished languages or is concerned with terminology translation in unrealistic scenarios. Either way, such methods are not relevant for the languages of the Shared Task because all of the target languages, with the exception of Chinese, are to some degree inflective languages.

There is, however, another body of work that addresses translation with terminology while accounting for morphological complexity of the target language (Exel et al., 2020; Niehues, 2021;

[5] https://github.com/robertostling/eflomal

[6] http://www.statmt.org/wmt21/terminology-task.html

	#	Source Term	Target Term
EN-FR	1	shortness of breath	essoufflé (e)
	2	Do the Five	"5 gestes" "barrière" ""
	3	Coronavirus (COVID-19)	coronavirus (COVID-19)
EN-RU	4	CDC	центры по контролю и профилактике заболеваний США (CDC)
	5	active cases	активные случаи [заболевания]
	6	contagious	заразный: передающийся при контакте
CS-DE	7	Lions Clubs International	Lions Clubs
	8	VIII. hlavový nerv	Nervus vestibulocochlearis
	9	Eli Lilly	Eli Lilly and Company

Table 2: Examples of terms removed by basic filtering.

Bergmanis and Pinnis, 2021). We base our submission on Bergmanis and Pinnis (2021) and employ target lemma annotations (TLA) to augment MT training data. An example of a sentence fragment annotated with TLA is "*infections*|**s** инфекция|**t** *result*|w *in*|**w** *mild*|**w** *symptoms*|**w**", where |**s**, |**t**, |**w** are factors indicating whether token is a source language term, a lemma (the dictionary form) of a target language term, or an ordinary source language word respectively. Systems trained on such data are equipped with a mechanism for passing soft terminology constraints at inference time. An essential property of MT systems trained using TLA is that they learn not just to copy but also to inflect the provided terminology constraints according to the target morphosyntactic context. Therefore, the translation of the sentence above, for example, "инфекций приводят к легким симптомам", contains the plural noun "инфекций" and not just the annotated singular form "инфекция".

3 Experiments

Data. We use all parallel data provided for the Shared Task for training, except for development data which we use to choose the best model for final submission. Although back-translated monolingual data could, in theory, improve the overall translation quality, we do not use it to train our systems because typically, the monolingual target data is selected based on its similarity with the target domain data. However, the scenario proposed for the Shared Task assumes that the domain is novel; thus, we aim to explore the merits of terminology translation and do not look for extra synthetic target domain data.

MT Model and Training. For system training, we use the *Marian toolkit* (Junczys-Dowmunt et al., 2018) because of its factored model functionality developed within the scope of the *User-Focused Marian project* [7][8]. In this Shared Task, we train standard MT systems that mostly follow the Transformer (Vaswani et al., 2017) *base* model configuration. The only deviations from the standard configuration are 1) the use of source-side factors (we use factor embeddings of dimensionality 8 and concatenate them with word embeddings), 2) increased *–optimizer-dealy* (from 16 to 24), and 3) an increased maximum sequence length (from 128 to 196 tokens). These changes are necessary purely for TLA support during training and inference: increased sequence length accounts for longer input sequences due to TLA and terminology constraints, while increased optimizer delay compensates for fewer sentences fitting in workspace memory-based batch due to their increased maximum length.

4 Results

We trained one NMT system per translation direction and evaluated translation quality on the development sets using the terminology translation evaluation tool provided by the Shared Task[9] (ibn Alam et al., 2021). We compare the baseline translation scenario where no terms are annotated in the source text with improved scenarios where terms are annotated using term collections acquired using the different filtering and term translation equivalent selection strategies.

When analysing the lemmatized exact match accuracy, we must bear in mind that the evaluation data similarly to the term collections feature term

[7] https://marian-project.eu/

[8] https://github.com/marian-cef/marian-examples/blob/forced-translation/forced-translation/docs/Experiments.md

[9] https://github.com/mahfuzibnalam/terminology_evaluation

	#	Source Term	1st Trg Term	Alignment-based
EN-FR	1	disease outbreak	apparition de maladie	épidémie
	2	epidemiologist	épidémiologistes	épidémiologiste
	3	wash your hands	lavez-vous les mains	laver les mains
EN-RU	4	Center for Disease Control	центры по контролю и профилактике заболеваний США	Центр контроля и профилактики заболеваний
	5	COVID-19 crisis	кризиса, связанного с COVID-19	кризис из-за COVID-19
	6	health crisis	кризисной ситуации, которая сложилась сегодня в сфере общественного здравоохранения	кризис здравоохранения
CS-DE	6	benzoylperoxid	Dibenzoylperoxid	Benzoylperoxid
	7	alternativní medicína	Alternativmedizin	Alternative Medizin
	8	AV ČR	Akademie der Wissenschaften der Tschechischen Republik	AV ČR

Table 3: Examples of terms selected by either **1st Trg Term** or **Alignment-based** term selection strategy for term entries with multiple translation equivalents.

EN-FR	**BLEU**	**Accuracy**	**Window 2**	**Window 3**	**1 - TERm**
Baseline	44.5	0.885	0.286	0.278	0.575
1st Trg Term	44.2	0.922	0.284	0.278	0.572
1st Trg Term, IDF≥5	44.3	0.905	0.281	0.275	0.574
1st Trg Term, IDF≥7	44.4	0.890	0.285	0.280	0.574
Alignment-based	44.6	**0.936**	0.291	0.285	0.576
Alignment-based, IDF≥5	44.6	0.918	0.287	0.281	0.576
Alignment-based, IDF≥7	44.4	0.896	0.284	0.278	0.574
EN-RU	**BLEU**	**Accuracy**	**Window 2**	**Window 3**	**1 - TERm**
Baseline	24.9	0.760	0.163	0.164	0.398
1st Trg Term	24.6	0.751	0.163	0.164	0.394
1st Trg Term, IDF≥5	25.0	0.815	0.176	0.176	0.401
1st Trg Term, IDF≥7	25.1	0.800	0.174	0.175	0.401
Alignment-based	25.0	**0.833**	0.174	0.175	0.403
Alignment-based, IDF≥5	25.1	0.821	0.176	0.177	0.402
Alignment-based, IDF≥7	25.2	0.792	0.171	0.172	0.401
CS-DE	**BLEU**	**Accuracy**	**Window 2**	**Window 3**	**1 - TERm**
Baseline	29.9	0.824	0.376	0.368	0.390
1st Trg Term	26.3	0.851	0.338	0.328	0.340
1st Trg Term, IDF≥5	29.5	0.852	0.371	0.361	0.378
1st Trg Term, IDF≥7	29.6	0.837	0.373	0.364	0.381
Alignment-based	26.8	0.849	0.346	0.337	0.348
Alignment-based, IDF≥5	29.6	**0.853**	0.373	0.364	0.386
Alignment-based, IDF≥7	29.7	0.837	0.375	0.366	0.389

Table 4: Development set results for baseline MT systems and systems using terminology integration. MT systems using terminology integration are named after the method used for terminology filtering. The numerically highest scores according the Lemmatized Exact-Match Accuracy (Accuracy) in data setting are **bold**. For detailed description of other metrics consult (ibn Alam et al., 2021).

entries 1) with more than one allowed synonymous translation equivalent (not counting different inflected forms), and 2) where different terms are merged into one entry (see examples in Section 2.1). This consequently means that the evaluation procedure 1) allows terminological ambiguity on the target side, 2) does not allow analysing terminology translation consistency, and 3) may depict a rough estimate of terminology translation accuracy. Therefore, we believe that the lemmatized exact match accuracy results should be analysed with a grain of salt.

That being said, the results in Table 4 show that the metric improves when using a term collection in all but one experiment. The fact that in overall the Alignment-based term collections show better overall translation results (in terms of BLEU) and also allow reaching the highest terminology translation accuracy results shows that relying on the first translation equivalent in a term entry is not a good idea.

We also see that the overall terminology translation quality is already relatively high for the baseline systems ranging from 76% for EN-RU till 88.5% for EN-FR. This makes us wonder whether the evaluated domain can be considered emerging as it features few novel terms and the majority are well handled by the baseline systems. To investigate further, we analysed whether the bilingual terminology that can be found in the development sets is also present in the training data of the NMT systems. We found that for CS-DE, 97.9% and 92.5% of such (unique) bilingual terms are featured at least once or 10 times in the training data respectively. The numbers are even higher if we analyse running terms (tokens) – 99.8% and 98.7% respectively. Since the terminology for EN-RU was human-created and not extracted from parallel data, it shows slightly lower results when analysing unique terms – 93.5% and 88.3% respectively. However, the situation is similar to CS-DE when analysing running terms – 99.1% and 97.8% of bilingual terms found in the EN-RU development data are also found in the training data at least once and 10 times respectively. Based on these findings, we believe that the validation data does not depict an emerging domain and does not help analysing terminology translation quality for emerging domains.

When analysing the overall translation quality (in terms of BLEU), we see that term filtering using the IDF-based filter is crucial when relying on very noisy and automatically acquired term collections (as was the case with the CS-DE term collection). The results show that translation quality drops by 3 BLEU points when using the unfiltered term collections. This shows that too general (and ambiguous) terminology can be harmful and lower translation quality. The overall translation quality change is marginal for the translation directions that featured human-created term collections (EN-FR and EN-RU), however we do see an increase in terminology translation accuracy.

Our **final submission** consists of machine translations of Shared Task test sets provided by general-purpose MT systems that use dynamic terminology integration using TLA (Bergmanis and Pinnis, 2021). To translate our final submissions, term collections are filtered by basic filters (see Section 2.1) for EN-FR and EN-RU language pairs, while for CS-DE language pair, we also use IDF>5 filtering. We use the statistical word alignment term selection strategy for term entries with multiple translation equivalents for all language pairs. The development set results for the corresponding systems are marked bold in Table 4.

5 Discussion

Shared Task. Results of automatic metrics show that our baseline systems are already well equipped to translate the development and test sets regardless of their seemingly novel domain. Indeed – we found no statistically significant differences in scores measuring general translation quality between the baseline systems and systems with terminology integration. Preliminary test results suggest a similar pattern in other submissions (c.f. results of submissions by Prompt). The only seemingly meaningful differences are in metrics specifically targeting terminology integration. These results are in stark contrast with previous work (Exel et al., 2020; Niehues, 2021; Bergmanis and Pinnis, 2021) which report significant improvements not only on terminology use targeted metrics but also on metrics measuring general translation quality. This disparity suggests that test data is *not* from an emerging or novel domain (at least as far MT systems trained on the training data provided are concerned). Considering this shortcoming, together with the visibility of WMT Shared Tasks, these results pose a risk of misrepresenting the problem the Shared Task was set out to research. The outcome might be

unintended downplaying of the role of terminology translation for technical domains, which could lead to diminishing interest in terminology translation from the MT research community.

Term Collections in MT. Tables 1, 2, and 3 of Section 2 provide numerous examples of problems present in the provided term collections. We believe that these examples illustrate the understanding of the purpose and desired qualities of a term collection not just of those individuals involved in preparing term collections for the Shared Task but also to a broader community of translation professionals. Many of the problematic examples suggest that the shift from human-readable to machine-readable term collections is not there yet, or that it has happened rather formally by merely reformatting the for-human-made term collections into neater TSV-formatted files. While having TSV-formatted files helps for the file to be machine-readable, it does not make the content machine-usable. The standards for-machine-made term collections have to be higher than those made for humans. At least as long as there is no sophisticated intelligence in the MT workflow that is on par with humans to recover from the irregularities and noise present in the term collections typically made for humans. Likewise, the encyclopedia-style entries explaining a concisely coined concept of the source language using a whole sentence to define it in the target language are still present in for-human-made term collections, but they are of no use for current MT systems.

If translation with terminology is supposed to improve MT for novel domains, the term collections, being the supposed source of the expected improvement, have to be of a higher quality than the MT systems they are intended to improve.

Acknowledgements

This research has been supported by the User-focused Marian project which is co-financed by the European Union's Connecting Europe Facility programme (Action number: 2019-EU-IA-0045).

References

Md Mahfuz ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency.

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with lematus. 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 1391–1400.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111.

Miriam Exel, Bianka Buschbeck, Lauritz Brandt, and Simona Doneva. 2020. Terminology-constrained neural machine translation at SAP. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 271–280, Lisboa, Portugal. European Association for Machine Translation.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

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.

Masato Neishi and Naoki Yoshinaga. 2019. On the relation between position information and sentence length in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338.

Jan Niehues. 2021. Continuous learning in neural machine translation using bilingual dictionaries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 830–840.

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

Mārcis Pinnis. 2015a. Dynamic terminology integration methods in statistical machine translation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 89–96, Antalya, Turkey.

Mārcis Pinnis. 2015b. *Terminology Integration in Statistical Machine Translation*. Ph.D. thesis, PhD thesis, Faculty of Computing, University of Latvia, Riga, Latvia.

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 systems for WMT21: Terminology translation Shared Task

Josef Jon and **Michal Novák** and **João Paulo Aires** and **Dušan Variš** and **Ondřej Bojar**
Charles University
{jon,aires,varis,bojar}@ufal.mff.cuni.cz

Abstract

This paper describes Charles University submission for Terminology translation Shared Task at WMT21. The objective of this task is to design a system which translates certain terms based on a provided terminology database, while preserving high overall translation quality. We competed in English-French language pair. Our approach is based on providing the desired translations alongside the input sentence and training the model to use these provided terms. We lemmatize the terms both during the training and inference, to allow the model to learn how to produce correct surface forms of the words, when they differ from the forms provided in the terminology database. Our submission ranked second in Exact Match metric which evaluates the ability of the model to produce desired terms in the translation.

1 Introduction

Terminology integration, or, more generally, constrained translation in NMT was extensively studied in recent years. Lexically constrained translation means that aside from the source sentence, we have available some additional knowledge of what tokens or expressions should appear in the translation and we want to force the system to include them in the generated output. Three main ways of enforcing these constraints have been studied.

First, replacing the source part of the constraint that is found in the source sentence with a placeholder which is then copied by the model into the output. There it gets replaced by the target part of the constraint (Luong et al. (2015); Crego et al. (2016)).

Second way is to modify the decoding search algorithm in a way that only allows hypotheses containing the constraints to be marked as finished (Anderson et al. (2017); Hasler et al. (2018); Chatterjee et al. (2017); Hokamp and Liu (2017); Post and Vilar (2018); Hu et al. (2019))

Finally, some works focus on providing the constraints directly to the model as part of the input sequence. The model is trained to incorporate these constraints into the output, for example Dinu et al. (2019); Chen et al. (2020); Song et al. (2019) or Bergmanis and Pinnis (2021).

As apparent from previous paragraphs, the problem of integrating lexical constraints into NMT is well studied, but one issue was largely ignored. In inflected languages, the surface form of the constraint in the output cannot be known beforehand, as there are usually many possible ways to translate a sentence and many of them need different surface forms of the constraint to be fluent and grammatically correct. For example, let's say we have a terminology database containing term pair *influenza -> grippe* and this source sentence:

During the 2018-2019 **influenza** season.

Possible correct translation is:

Pendant la saison **grippale** 2018-2019.

Where the term base noun form *grippe* is inflected into adjective *grippale*. Traditional constraint integration methods will try to enforce the term DB form *grippe* instead.

We have studied this problem in our recent work (Jon et al., 2021) concurrently with Bergmanis and Pinnis (2021), who used a very similar approach. Both works use different languages and evaluation pipelines and both show that the proposed approach is feasible.

2 Method

NMT models are known to produce fluent, consistent and grammatically correct outputs (Popel et al., 2020). Thus, it makes sense to utilize this ability of the model to inflect the constraint into correct form, instead of trying to disambiguate the form externally. Our approach is based on annotating

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 828–834
November 10–11, 2021.

source sentences with the desired target constraints and training the model to incorporate these constraints into the output. We publish our preprocessing scripts at https://github.com/ufal/bergamot/wmt21-terminology

2.1 Term annotation

There are multiple possibilities in how to exactly annotate the source sentence. For example, let's say the terminology database contains entries:

runny nose -> nez qui coule
fever -> fièvre

and we have a sentence:

*And are you having a **runny nose** or **fever**?*

One way is to replace the part of the source sentence containing the source constraint with the target part of the constraint:

*And are you having a **nez qui coule** or **fièvre**?*

Another option is to insert the translation tokens after the source part of the constraint and use factors to mark which tokens of a sentence belong to source constraint, which tokens are part of the target constraint and which are neither. For example, if factor with value 2 denotes that the token is part of the translation, value 1 means that the token is part of a source constraint and 0 means that it is just ordinary token, we get:

And_0 are_0 you_0 $having_0$ a_0 ***runny***$_1$ ***nose***$_1$ ***nez***$_2$ ***qui***$_2$ ***coule***$_2$ or_0 ***fever***$_1$ ***fièvre***$_2$ $?_0$

We use simpler method to integrate the constraints in our systems: we append them to the source sentence as a suffix, separated by a special token (<sep>) and in case of multiple constraints for a single sentence, we separate them by a different token (<c>):

*And are you having a **runny nose** or **fever**? <sep> **nez qui coule** <c> **fièvre***

For more details about the possible modifications of our method, comparisons with other approaches and detailed evaluation and analysis, we refer the reader to our previous work (Jon et al., 2021).

2.2 Training data generation

We prepare synthetic constraints for parallel training data by sampling random token subsequences from the target sentence. These subsequences are used as a suffix for the source sentence as described earlier. There is a number of parameters guiding this process. Every token in a sentence can become a start of a constraint with probability s. Unless stated otherwise, we set $s = 0.1$. Any subsequent token in an open constraint can end the constraint with probability $e = 0.75$. We permit multiple non-overlapping constraints for a sentence. We skip the sentence for constraint generation (i.e. leave it without any constraints) with probability $n = 0.1$. In pseudocode:

```
s=0.1
e=0.75
n=0.1
for sent in text:
  r=random()
  constraints=[]
  if r > n:
    open=False
    constraint=""
    for t in tokens(sent):
      r=random()
      if open:
        if r < e:
          constraints.append(constraint)
          open=False
        else:
          constraint+=t
      else:
        if r < s:
          constraint+=t
          open=True
  print(sent, constraints)
```

Since the task allows for multiple target variants for a single source term, we have to account for such possibility in our training data generation. We assume that each generated constraint can have a variant with probability $v = 0.1$. This variant is sampled randomly (with no relation to the source sentence) from n-grams extracted from the target training corpus (so it is not a part of a current target sentence, but it is still a plausible subsequence in the target language). The variant has the same number of tokens as the original constraint with probability $l = 0.9$, otherwise the length of the variant is taken from triangular distribution between 1 and 9 with mode 2. The variants of a single constraint are delimeted with another special token <v>. None of the probabilities were tuned for improving results, we chose them based on manual inspection of the generated data. We use values that produced similar counts and lengths of the constraints as in

the validation set.

2.3 Lemmatization

The training data generation method described above works, but suffers from the issues described in the introduction – the system learns to generate only the exact tokens supplied as constraints in the suffix, but doesn't account for different possible inflections of the constraints in different contexts. To overcome this issue, we lemmatize the constraints both during the training and during test time. This way, the model learns to not only generate the correct words in the output, but also to correctly inflect them.

2.4 Source-side terminology matching

To find term pairs from terminology database in the input text, we lemmatize both the database source side and input sentences and search for the terms that appear either on lemma or surface form level. Since our lemmatizer works with context, we lemmatize both the text and the database word by word to ensure consistent lemmas. For the models trained with lemmatized constraints, we lemmatize also the target side of the terminology database and anntote the source sentence with lemmas of the target terms, instead of the surface forms.

3 Experiments

3.1 Data

We used all English-French corpora allowed by the organizers, aside from Paracrawl (with the exception of one model, which is marked). Namely this means Europarl v10, Common Crawl, UN Parallel Corpus v1.0, News Commentary v16 and Gigaword. We used WMT15 news test set as our validation set. After deduplication and filtering, the resulting training set consists of 24.6M sentences without Paracrawl and 125.9M including Paracrawl.

3.2 Tools

We use MarianNMT (Junczys-Dowmunt et al., 2018) to train Transformer-big models with standard parameters (Vaswani et al., 2017). The corpora are filtered using Moses cleaning script[1] and `fasttext` langid (Joulin et al., 2016). We split the text into subwords using FactoredSegmenter[2] based on SentencePiece (Kudo and Richardson, 2018) and lemmatize using UDPipe (Straka and Straková, 2017). BLEU scores are computed using SacreBLEU (Post, 2018), other metric are obtained by an evaluation script provided by the organizers[3](ibn Alam et al., 2021).

3.3 Evaluation

The script provided by the task organizers computes multiple metrics: BLEU, (Lemmatized) Exact Match, Window overlap and 1-TERm.

Exact match is a fraction of constraints which were produced in the outputs (the output sentences are lemmatized and the search is performed on both lemma and surface form level). This metric can be cheated in two ways – first, the system can place the target constraint at arbitrary place in the output, e.g. we can just translate with a non-constrained MT model, append the constraints at the end and obtain a perfect score. Second way is related to lemmatization – the system can produce any valid surface form of the constraint and even though this form is not grammatically correct in context of the output sentence, it still gets counted as matching. On the other hand, without lemmatization, only the word forms listed in the terminology database would get accepted, which would not cover all the possible correct forms.

Window overlap aims to overcome the first shortcoming of EM by evaluating placement of the constraint in the output. For each constraint in the translation and in the reference, windows of n tokens are extracted and compared with each other to see if the system places the constraint in similar context as in the reference. 2 and 3 token windows are used.

TERm metric is weighted TER which uses higher weights for tokens which are part of a term from terminology database to increase sensitivity to differences in the terminology. In the experiments, we observed that 1-TERm score is influenced mainly by the overall translation quality and less so by the term integration. We believe that this metric alone is also not sufficient for comparing ability to integrate constraints in different models, as the results seem to rely mainly on the "baseline" model performance, i.e. big general NMT model, trained on more data, which provides better overall translation quality, but does not explicitly

[1] https://github.com/marian-nmt/moses-scripts

[2] https://github.com/microsoft/factored-segmenter

[3] https://github.com/mahfuzibnalam/terminology_evaluation

Constraints	Corpus	Variants	BLEU	EM	window 2	window 3	1-TERm
None	Base	-	43.976	0.862	0.289	0.283	0.584
None	Base+paracrawl	-	45.084	0.851	0.283	0.279	0.587
None	Base+bt	-	42.319	0.834	0.282	0.275	0.575
SF	Base	no	43.771	0.953	0.297	0.290	0.581
SF	Base	yes	41.656	0.982	0.253	0.255	0.555
Lemm	Base	yes	42.317	0.919	0.278	0.274	0.552
Lemm	Base	no	44.959	0.961	0.302	0.296	0.591
Lemm*	Base	no	44.623	0.909	0.292	0.288	0.588
Final combined	-	-	45.590	0.989	0.309	0.304	0.600

Table 1: Results of our models on official validation set. The first column specifies whether the constraint were lemmatized (*Lemm*) or not SF (*SF*), second one shows which part of copora we used. Base means all parallel data allowed by the organizers with exception of Paracrawl. Third column says whether we provided all possible variants of the target term from terminology database to the model, on we only the first one. Asterisk in *Constraints* column means that the model was trained with these form of constraints, but no constraints were provided during the test time.

integrate constraints, may obtain higher scores than a smaller constrained model with perfect constraint integration ability.

3.4 Results

We trained our models by techniques described earlier and we present metrics computed by the official evaluation script in Table 1. Due to time and computing constraints, most of the models were trained without Paracrawl corpus and we only trained one baseline on dataset including Paracrawl for comparison. We compared integrating constraints in the surface form (so the model needs to produce exactly the same token as provided in the input) and constraints in lemmatized form (the model can produce different inflection of the provided constraint). We also compared providing all possible variants of the target constraint from terminology database (delimeted by <v>, as described earlier), or just the first possible translation.

We see that in most metrics, the model which is trained with lemmatized constraints and uses only one variant performs the best. Systems trained with multiple variants of the target term show large degradation in BLEU scores. We suppose one of the problems in our method is that during training, only the true constraint variant from the target is plausible translation of the source, others are n-grams sampled randomly from the whole corpus. Thus, the negative samples are very easy to distinguish during the training, but during the test time, the variants are provided by the term base and they are all plausible in the context. We will analyse these results further in the future.

Our final primary submission is a combination of all the models. They are ranked by their respective BLEU scores on validation set and we check if the produced translation contains the desired term either at lemma level. We use the best ranking systems' translation that does, or, in case none of the systems produced the term, we use the translation of baseline system.

The task organizers provide test set results.[4] Two metrics were considered for the ranking. First, COMET (Rei et al., 2020), which evaluates general translation quality without special regard for specific terminology. Secondly, exact match, which measures how many of the desired constraints were actually produced in the output, but suffers from the issues described earlier. Our primary submission was ranked on joint 6th-10th place out of 21 systems according to COMET and 1st-3rd according to exact match.

3.5 Error analysis

Our submitted system did not cover 10 out of 872 term occurrences in the validation set. We analyse these ten errors manually. Six of these errors are related to casing, notably by translating *SARS-CoV* as *Sars-CoV*, instead of keeping the original casing (five occurrences). This is caused by our lemmatization pipeline, which produces *Sars* as lemma of *SARS*. We confirmed that after manually fixing the input and restoring the original casing, the system produces correct output. Other five examples classified as errors are presented in Table 2.

[4] https://docs.google.com/spreadsheets/d/13-1kwDq9yerehSF4No6ZTLqPXjSaL7HOsksnZDjjO-Y/

i	Source	Target terms	MT output
1	Many human **Coronavirus** have their origin in bats.	coronavirus	Beaucoup de **Coronavirus** humains ont leur origine dans les chauves-souris .
2	Data from these practices are reported online in a weekly return, which includes monitoring weekly rates of influenza-like illness (ILI) and other communicable and **respiratory diseases** in England.	maladies respiratoires / maladies communes des voies respiratoires / maladie respiratoire	Les données relatives à ces pratiques sont communiquées en ligne dans une déclaration hebdomadaire, qui comprend le suivi des taux hebdomadaires de maladies grippales(SG) et d'autres **maladies** transmissibles et **respiratoires** en Angleterre.
3-4	We will share the protocol with UK colleagues and the I-MOVE consortium who have recently obtained EU Horizon 2020 funding from the stream "Advancing knowledge for the clinical and public health response to the **novel coronavirus epidemic**"	coronavirus nouveau; epidémie / épidémies / épidémique	Nous partagerons le protocole avec nos collègues du Royaume-Uni et le consortium I-MOVE , qui ont récemment obtenu un financement de l'OMS horizon 2020 dans le cadre du projet «Advancing knowledge for the clinical and public health response to the **novel coronavirus epidemic**»
5	The statistical methodology is in support of a policy approach to widespread disease **outbreak**, where so-called nonpharmaceutical interventions (NPIs) are used to respond to an emerging pandemic to produce disease suppression.	épidémie / épidémies / épidémique	La méthodologie statistique est à l'appui d'une approche politique face à l'apparition de **maladies à grande échelle**, où les interventions dites non pharmaceutiques (ISP) sont utilisées pour répondre à une pandémie émergente afin d'éliminer les maladies.

Table 2: Rest of the examples with uncovered terms. *Target terms* column shows possible translations of the source terms (bold) as provided in the terminology database.

Another casing error occurs in translation of the sentence (1) in the table. The model keeps the original source casing, but the evaluation script only checks for lower-case *coronavirus*. This sentence is also actually part of unsplit and wrongly tokenized source line *The large number of host bat and avian species, and their global range, has enabled extensive evolution and dissemination of coronaviruses.Many human coronavirus have their origin in bats.* This may be a source of further confusion for the model.

In example (2), the related terminology DB pair is *respiratory diseases -> maladies respiratoires*. In the model output, the adjective *transmissibles* is interjected between the terms, which is probably not an error from human point of view, but is hard to evaluate automatically.

In example (3-4), the model does not translate the name of the project in quotes, thus it does not produce the desired translations of both *epidemic -> épidemie* and *novel coronavirus -> coronavirus nouveau* .

Finally, (5) is a true failure of the model to use the provided term. The sentence produced by the model is a plausible and semantically correct translation, but it is not using the desired term. For further analysis, we manually replaced the produced translation of the term (*maladies à grande échelle*) with the term from the terminology database (*épidémie*). We computed cross-entropy scores for the modified sentence both with and without providing the constraint to the model. We saw that when provided with the constraint, the modified translation is more probable than without the constraint (but still slightly less probable than the translation that was actually produced.) This shows that the method still partially works in this case, but the bias towards producing the term in the output needs to be stronger – we plan to explore this further using contrastive learning.

4 Conclusion

We describe our submission to Terminology translation Shared Task at WMT21. We show our method can effectively incorporate the terminology without negative effects on overall translation quality. We analysed all ten examples in the validation set where our model did not cover the desired term constraint and we show that most of them can be explained by preprocessing issues.

Acknowledgements

Our work is supported by the Bergamot project (European Union's Horizon 2020 research and innovation programme under grant agreement No 825303) aiming for fast and private user-side browser translation, GA ČR NEUREM3 grant (Neural Repre-

sentations in Multi-modal and Multi-lingual Modelling, 19-26934X (RIV: GX19-26934X)) and by SVV 260 575 grant.

The work described herein has also been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure, supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization. Main track.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Eva Hasler, Adrià de Gispert, 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, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

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. Association for Computational Linguistics.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Md Mahfuz ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency.

Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. End-to-end lexically constrained machine translation for morphologically rich languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4019–4033, Online. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. 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, Beijing, China. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

PROMT Systems for WMT21 Terminology Translation Task

Alexander Molchanov, Vladislav Kovalenko & Fedor Bykov
PROMT LLC
17E Uralskaya str. building 3, 199155,
St. Petersburg, Russia
First.Last@promt.ru

Abstract

This paper describes the PROMT submissions for the WMT21 Terminology Translation Task. We participate in two directions: English to French and English to Russian. Our final submissions are MarianNMT-based neural systems. We present two technologies for terminology translation: a modification of the Dinu et al. (2019) soft-constrained approach and our own approach called PROMT Smart Neural Dictionary (SmartND). We achieve good results in both directions.

1 Introduction

The currently state-of-the-art approach of neural machine translation (NMT) does not inherently allow for explicit control over the system's output. That is why terminology translation has always been a problem for NMT systems. There are several approaches to solving this problem. One common paradigm is constrained decoding (Hokamp and Liu, 2017; Anderson et al., 2017; Post and Vilar, 2018), where the terminology matches are presented as hard constraints that the beam search must satisfy. Constrained decoding has its disadvantages: it is computationally expensive and can deteriorate the translation quality (Dinu et al., 2019). Another common approach is the one introduced by (Dinu et al., 2019): the terminological constraints are provided as input to the NMT as additional annotations inline with the source sentence. These can be considered 'soft' constraints, as there is no guarantee that the NMT system will indeed produce an output containing them.

In this paper we describe two approaches to terminology translation. First, we propose a modification of the (Dinu et al., 2019) approach. Second, we introduce our own technology PROMT Smart Neural Dictionary (SmartND) aimed at handling terminology translation.

The paper is organized as follows: in Section 2 we describe the systems built for the Task and the data we used. In Section 3 we describe our technologies for terminology translation. In Section 4 we present and discuss the results. We conclude the paper with discussion for possible future work in Section 5.

2 Systems overview

We submitted two single baseline transformer-based (Vaswani et al., 2017) systems trained with the `MarianNMT` (Junczys-Dowmunt et al., 2018) toolkit: English-Russian and English-French. We use all parallel data allowed by the organizers. The final systems have the same architecture: we use a shared vocabulary of sizes 16k and 32k for the English-French and English-Russian systems respectively. We use the `OpenNMT` toolkit (Klein et al., 2017) version of byte pair encoding (BPE) (Sennrich et al., 2016b) for subword segmentation. We use the devsets provided by the organizers as our development sets.

We build intermediate models to obtain back-translations (Sennrich et al., 2016a) for our final systems. We use iterative back-translation for the English-Russian system. The intermediate models are trained using SentencePiece (Kudo and Richardson, 2018) for subword segmentation as we noticed that SentencePiece-based models are more robust in low and middle-resource

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 835–841

conditions. We also tag all our synthetic data with special tokens at the beginning of the source sentences as described in (Caswell et al., 2019). Our final models use three types of synthetic data for training: back-translations, data with terminology and special data with placeholders for processing named entities during translation (see Molchanov, 2019 for details). The models are trained with guided alignment which is used at translation time by our SmartND technology. We obtain alignments using the `fast-align` (Dyer et al., 2013) tool. Both final models were trained for approximately 1.2M steps on two RTX 2080 GPUs.

We also perform fine-tuning for our final systems. There are two reasons for that. First, due to time constraints the initial final systems were trained with only one term in each sentence with terminology markup. After testing these systems we realized that they couldn't handle sentences containing multiple terms. The second reason is that we only processed parallel data and not back-translations for the initial final systems, whereas the 2020 news contain a lot of information about COVID etc. For fine-tuning we processed both the back-translated news and parallel data only using the glossary provided by organizers. This data was mixed with general parallel data.

2.1 Data preparation

There are several stages in our data preparation pipeline. These are mostly common filtering techniques. The statistics for the training data are shown in Table 1. The main stages of the pipeline are:

- Basic filtering
  This includes some simple length-based and source-target length ratio-based heuristics, removing tags, lines with low amount of alphabetic symbols etc. We also remove lines which appear to be emails or web-addresses and duplicates.

- Language identification
  The algorithm is a fairly simple ensemble of three tools: `pycld2` [1], `langid` (Lui and Baldwin, 2012), `langdetect` [2]. We only use pycld2 for large monolingual corpora.

- Bicleaner filtering
  We use the bicleaner (Ramírez-Sánchez et al., 2020) tool to filter parallel data. We discard all sentence pairs with the score threshold <= 0.3.

- Scoring with NMT models
  We finally score all parallel data and back-translations with our intermediate models to discard non-parallel sentence pairs and bad synthetic translations.

2.2 English-French

Due to time constraints and relatively large amounts of training data for the English-French pair we only build one intermediate model. We use all parallel data that we suppose to be of good quality (i.e. all data except the paracrawl, commoncrawl and giga corpora; we also randomly select only 2.5M sentence pairs from the United Nations corpus) and build a joint system trained to translate both from English into French and back. Basic filtering is applied to this data. We use a shared vocabulary of size 8k obtained with SentencePiece. We also tag the source side of the training data with language tokens. The model is trained for approximately 1M steps on two RTX 2080 GPUs. We then use this system to 1) score all parallel data in both directions; 2) translate the monolingual French news corpora into English. We translate the 2020, 2019 and 2018 news corpora. The final model is built using all allowed filtered parallel data, back-translated news and additional synthetic data for terminology markup (see Section 3 for details).

2.3 English-Russian

The English-Russian was a surprise pair announced roughly three weeks before the submission deadline. That is why despite the relatively small amounts of parallel data we only make two iterations of training intermediate systems. We first build an English-Russian system using all parallel data (except commoncrawl which we believe to be of bad quality; basic filtering is applied) including the Edinborough corpus of Russian news translated into English and separate SentencePiece-based vocabularies of

[1] https://pypi.org/project/pycld2/

[2] https://pypi.org/project/langdetect/

size 16k each. As there are approximately 25M parallel sentences pairs, we randomly select 25.5M from the back-translated Russian news corpus. We then use this model to translate the English monolingual 2020 news corpus into Russian. Then we build a Russian-English system with the same vocabularies using all completely filtered parallel data and the obtained English-Russian translations. After that we translate the Russian monolingual news (2020, 2019 and 2018) into English. The final English-Russian model is trained on all filtered parallel data (also scored with the two intermediate systems) and the back-translations of the Russian monolingual news. Despite the fact that it is better to use separate vocabularies for models with different alphabets we use a shared vocabulary because this is necessary for our terminology handling approach.

3 Terminology translation

In this section we describe our two approaches to terminology handling in detail.

3.1 SmartND

Our SmartND systems work on the backbone of the PROMT RBMT technology. The technology doesn't need any specific pretraining or fine-tuning. The entire process can be divided into three steps: dictionary creation, terminology search and output modification.

First off we create a PROMT dictionary in specific format based on the provided glossary. The dictionary is highly optimized for speed and

English-Russian						
	#sent	**#sent clean**	**#tok EN**	**#tok EN clean**	**#tok RU**	**#tok RU clean**
News-commentary	331,508	263,674	8,940,220	6,833,693	8,483,220	6,490,441
Paracrawl	5,377,911	3,384,721	122,008,867	72,171,449	100,966,255	63,635,823
UN	23,239,280	12,875,296	61,3108,270	401,818,416	578,849,401	375,889,876
WikiMatrix	1,661,908	896,209	39,460,867	22,136,958	36,102,154	19,909,749
Yandex corpus	1,000,000	770,424	24,685,829	18,849,831	22,613,143	17,341,207
Commoncrawl	878,386	309,378	22,000,613	7,375,812	21,152,629	6,712,507
WikiTitles	1,189,058	195,653	3,403,009	839,231	3,515,590	836,091
Total	33,678,051	18,695,355	833,607,675	530,025,390	771,682,392	490,815,694
English-French						
	#sent	**#sent clean**	**#tok EN**	**#tok EN clean**	**#tok FR**	**#tok FR clean**
Europarl	1,915,930	1,387,120	53,588,034	38,751,350	59,215,266	43,076,733
News-commentary	365,510	318,811	94,442,52	8,328,207	11,312,937	10,044,909
UN	25,805,088	15,076,117	681,718,544	457,225,777	790,583,218	535,347,782
Commoncrawl	3,244,152	1,832,936	82,530,944	43,761,355	92,685,758	50,241,069
Giga	22,520,376	11,559,142	685,336,581	304,757,715	826,389,803	362,087,754
Paracrawl	104,351,522	45,673,561	2,274,818,705	961,613,380	2,604,498,787	1,078,787,397
Total	158,202,578	75,847,687	3,787,437,060	1,814,437,784	4,384,685,769	2,079,585,644

Table 1: Statistics for the initial and filtered parallel data in sentences (#sent) and tokens (#tok); 'clean' stands for the final filtered versions of the corpora.

contains POS information for each lexeme along with the complete inflectional paradigm. If a term is present in any of our existing dictionaries, we copy the information from there. In other cases, we try to guess the POS and possible paradigm based on how the term ends. Currently, this system only works with nouns, so we omitted any verbs and adjectives present in the provided terminology glossary, as well as any ambiguous terms that could belong to different parts of speech (like 'quarantine'). If a term has multiple translations we either choose one (if the translations are interchangeable, like 'World Health Organization' – 'Organisation mondiale de la Santé' or 'Organisation Mondiale de la Santé') or omit the term entirely. We also drop any common terms that would likely be translated correctly by our NMT models (like 'coronavirus') or, in case of the English-Russian language pair, terms with translations that are incorrect ('Coronavirus crisis' – 'коронавирус кризис'). This ensures that SmartND will not interfere with a perfectly valid NMT output. We remove 30 entries from the original glossary.

The translation process is organized as follows. If a term is present in the input text, we search the NMT output for the expected term translation. If that translation is not present in the NMT output, our RBMT systems analyze it and determine the grammatical information (case and number) of the word which our NMT model used to translate the term. We use the word-level alignment provided by the NMT model to find the term-translation pair. Then we can substitute that word for the correct translation taken from the RBMT dictionary using the same case and number. The entire process depends of the NMT model providing good quality word-level alignment. We do not substitute the term translation if the alignment is incomplete in that part of the segment.

3.2 Soft-constrained Terminology Translation

Our second approach is based on (Dinu et al., 2019) with slight modifications. The general idea is quite simple: terminology is identified and tagged on the source side, and each term is appended by its translation (also tagged). The work of (Dinu et al., 2019) and (Bergmanis and Pinnis, 2021) is based on the `Sockeye` (Hieber et al., 2017) toolkit. Each part of the term and its translation is marked with a special source feature. Whereas `MarianNMT` doesn't support source features (and our systems are `MarianNMT`-based), we propose a 'trick' similar to the one described in (Tamchyna et al., 2017). We add special tokens after the term and its translation in the input string. In the first version of our systems we added special tokens after each part of the term and each part of its translation to 'mimic' a source feature behavior. But we noticed that the resulting strings are often too long, especially if the source line contains several multi-word terms. So we decided to simplify the algorithm and mark each term with three special tokens which indicate the beginning and end of the term itself and the end of its translation: <term_start>, <term_end> and <term_trans>.

We use the glossary provided for the Task to tag our parallel data. We also use the parallel WikiTitles corpora to create more synthetic data with terminology markup. For the English-Russian pair we use the provided WikiTitles corpus. For the English-French pair we use the `wikipedia-parallel-titles`[3] tool to extract the English-French Wikipedia titles. Note that we only use this corpus to indentify and tag terms in the provided constrained data. We apply the basic filtering to the titles corpora and then randomly select 10k parallel entries for data markup. The English-French glossary remains as is, whereas we generate all possible forms for the translations of the English-Russian glossary using our parser to be able to find them in the parallel data and process more sentences for training.

The data preprocessing is simple: we go through the parallel data line by line and identify the terms (either from the provided glossary or from the WikiTitles) on the source side. If a term is found, we look for any of its translations on the target side. If a translation is found, we tag the term and append the found translation as described above. We obtain about 2.1M sentence pairs for the initial system training and around 0.8M pairs for fine-tuning (using only the provided glossary) for the English-Russian pair, For the English-French pair we have around 0.8 sentence pairs for the initial system and 0.2M pairs for tuning.

[3] https://github.com/clab/wikipedia-parallel-titles

At translation time both glossaries remain as is because we don't use the lemmatized approach, so each term is appended by the initial form of the translation. The motivation for this is that we think that having seen different forms of words and expressions at training time the model can 'guess' that it should transform the initial form to the one necessary in this context (i.e. copy and inflect).

4 Results and discussion

In this Section we present the results on the dev and test sets both in terms of automatic and human evaluation and discuss advantages and drawbacks of our approaches. The results of automatic evaluation on the dev sets according to the tool (Mahfuz ibn Alam et al., 2021) provided by the organizers are presented in Table 2. We can see that both our approaches clearly outperform the baseline according to the terminology-related metrics. As for the BLEU (Papineni et al., 2002) scores, they slightly rise for both approaches which indicates positive results of the application of our approaches. We also present the final results for our submitted systems on the test sets in Table 3. They are generally consistent with the results we obtained on the dev sets.

4.1 Tuned models with SmartND

We observe minor decrease in the exact match scores for the tuned models with the SmartND technology. Surprisingly, our final English-Russian tuned system was ranked last on the test set according to the Exact Match and Window Overlap metrics. We performed human evaluation for these translations. The results show that the exact match scores decrease because of the translation of the term *COVID-19* which is translated as *Covid-19* by the tuned model. This is a perfectly fine translation, but the evaluation metric handles all term translations in case-sensitive mode. The tuned model outperforms the baseline model in all other aspects. This is probably a reason to 1) slightly modify our SmartND algorithm; 2) make the scoring metrics more robust regarding the case aspects.

English-French					
	BLEU	**Exact match**	**Window overlap (2)**	**Window overlap (3)**	**1-TERm**
Intermediate	38.45	0.82	0.255	0.253	0.53
Final	45.44	0.87	0.3	0.29	0.61
Final+Soft	45.86	0.966	**0.314**	**0.309**	0.613
Final+SmartND	45.51	0.922	0.307	0.303	0.613
Final tuned	45.29	0.867	0.297	0.289	0.61
Final tuned+Soft	**46.04**	**0.973**	0.309	0.306	**0.614**
Final tuned+SmartND	45.31	0.87	0.299	0.29	0.611
English-Russian					
	BLEU	**Exact match**	**Window overlap (2)**	**Window overlap (3)**	**1-TERm**
Intermediate	23.92	0.707	0.165	0.163	0.395
Final	27.05	0.84	0.205	0.203	0.439
Final+Soft	26.94	0.86	0.2	0.198	0.44
Final+SmartND	**27.22**	0.867	0.208	0.207	**0.44**
Final tuned	26.75	0.742	0.193	0.19	0.433
Final tuned+Soft	26.9	**0.914**	**0.215**	**0.214**	0.438
Final tuned+SmartND	26.91	0.765	0.195	0.191	0.434

Table 2: Results of the Terminolgy Translation Task on the dev sets.

English-French								
	BLEU	**EM**	**WO (2)**	**WO (3)**	**1-TERm**	**COMET**	**Rank (EM)**	**Rank (COMET)**
Final tuned+Soft	47.69	0.974	0.359	0.352	0.625	0.752	**1-3**	3
Final tuned+SmartND	47.89	0.966	0.357	0.348	0.626	0.746	**1-3**	4-5
Best Scores	49.60	**0.974**	**0.359**	**0.352**	0.632	0.781	**-**	-
English-Russian								
	BLEU	**EM**	**WO (2)**	**WO (3)**	**1-TERm**	**COMET**	**Rank (EM)**	**Rank (COMET)**
Final tuned+Soft	31.06	0.909	0.254	0.255	0.482	0.631	**1**	**1-2**
Final tuned+SmartND	31.92	0.788	0.243	0.241	0.487	0.634	10	**1-2**
Final+SmartND	31.52	0.857	0.251	0.250	0.482	0.624	2-5	3
Best Scores	**31.92**	**0.909**	**0.254**	**0.255**	**0.487**	**0.634**	-	-

Table 3: Final results of the Terminolgy Translation Task on the test sets.
EM stands for *Exact Match*, WO stands for *Window Overlap*. The Best Scores row shows the best scores on the test set for each metric from all participants, the PROMT systems are in bold.

4.2 SmartND and Soft-constrained translation

We compared the two approaches to handling terminology during our experiments. They both have advantages and drawbacks originating from their architecture.

The SmartND technology is more reliable as it almost always produces the right translation given the input from the glossary. However, a noisy glossary is a great problem for SmartND as in this case it needs to be carefully handled and filtered by linguists. The second problem with SmartND is that it sometimes (rarely) produces incorrect translations putting words in the wrong form in the output. This concerns morphologically rich languages, and the reason for it is that it is sometimes hard to parse the output and define the correct form for the term translation.

The soft-constrained approach is more robust to noise in terminology glossaries. The NMT output is more fluent as the system tends to put the terms in the right forms or generate its own translation. However, as we noticed, this technology cannot handle very noisy glossaries or entries either. The soft-constrained systems also require specific training and fine-tuning and data for it, which can be costly.

4.3 General translation quality

We also observe the fact that better baseline models receive better scores according to all metrics. We paid more attention to the English-Russian direction in this task and contributed more work to it. As a result, we obtain generally higher scores on the English-Russian direction compared to the English-French direction according to all metrics.

5 Conclusions and future work

In this paper we presented our submissions for the WMT21 Shared Terminology Translation Task. We show good results in both directions we participate (English-French and English-Russian). We are planning to make more thorough analysis of the results of our work on both the dev and test sets. We are also planning to try the lemmatized approach as described in (Bergmanis and Pinnis, 2021).

References

Md Mahfuz ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the Evaluation of Machine Translation for Terminology Consistency. *Computing Research Repository*, arXiv:2106.11891.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark.

Toms Bergmanis and Marcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 53–63, Florence, Italy.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of NAACL HLT 2013*, pages 644–648, Atlanta, USA.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *Computing Research Repository*, arXiv:1701.02810.

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.

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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of ACL 2012, System Demonstrations*, pages 25–30, Jeju, Republic of Korea.

Alexander Molchanov. 2019. PROMT Systems for WMT 2019 Shared Translation Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 302–307, Florence, Italy.

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.

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, New Orleans, Louisiana.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón and Sergio Ortiz Rojas. 2020. Bifixer and Bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal.

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 2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 2016b. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 35–40, Berlin, Germany.

Aleš Tamchyna, Marion Weller-Di Marco and Alexander Fraser. 2017. Modeling Target-Side Inflection in Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 32–42, Copenhagen, Denmark.

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.

SYSTRAN @ WMT 2021: Terminology Task

MinhQuang Pham[†‡], **Antoine Senellart**[†], **Dan Berrebbi**[†], **Josep Crego**[†], **Jean Senellart**[†]
[†] SYSTRAN, 5 rue Feydeau, 75005 Paris, France
firstname.lastname@systrangroup.com
[‡] LISN, Université Paris-Saclay 91405 Orsay, France
firstname.lastname@limsi.fr

Abstract

This paper describes SYSTRAN submissions to the WMT 2021 terminology shared task. We participate in the English-to-French translation direction with a standard Transformer neural machine translation network that we enhance with the ability to dynamically include terminology constraints, a very common industrial practice. Two state-of-the-art terminology insertion methods are evaluated based (i) on the use of placeholders complemented with morphosyntactic annotation and (ii) on the use of target constraints injected in the source stream. Results show the suitability of the presented approaches in the evaluated scenario where terminology is used in a system trained on generic data only.

1 Introduction

The high quality obtained by out-of-the-box neural machine translation (NMT) systems (Bojar et al., 2016) has boosted the adoption of automatic translation by the industry and invigorated the research and development on domain adaption and integration of technology in human translation workflows. For instance, combination with translation memories (Bulte and Tezcan, 2019; Xu et al., 2020; Pham et al., 2020), terminology handling (Hasler et al., 2018; Dinu et al., 2019; Michon et al., 2020; Bergmanis and Pinnis, 2021), interactive translation (Peris and Casacuberta, 2019), post-editing modelling (Chatterjee et al., 2020) or dynamic adaptation (Farajian et al., 2017) are all different techniques to make machine translation part of real-life localization workflow.

Terminology resources with all their sophistication have been the core building bricks and a continuous challenge to acquire in volume (Senellart et al., 2003) for rule-based engines. At the other extreme, they have been reduced to corpus or aligned "phrase pairs" (Schwenk et al., 2008) for Statistical Machine Translation approaches, missing most of their intrinsic linguistic properties. In contrast, neural machine translation operates on word and sentence representations in a continuous space so, on the one hand, it has access to deep actual linguistic knowledge (Conneau et al., 2018) and demonstrates a huge ability to generalize. But on the other hand, results are more difficult to interpret (Koehn and Knowles, 2017), and subsequently the translation process is far more complicated to control. Therefore, as for several other linguistic annotations, the challenge is how terminological information can be "passed" to the model. From a human perspective, even though presentation and usage of dictionaries have evolved from ontology (as found in paper dictionary) to corpus-based presentation, looking up terms in a dictionary is the ultimate point of reference for validating the correct term for a specific domain and context.

Inline with the conditions of the WMT 2021 terminology shared task, we present English-to-French NMT engines built from abundant generic (out-of-domain) training data. We evaluate several methods to enhance translation engines with the ability to integrate terminology as a quick way to dynamically specialize a translation to a particular domain, which in this case considers the new COVID-19 domain and the large efforts for translation of critical information regarding pandemic handling and infection prevention strategies. In-domain resources are limited to word- and phrase-level terminology entries created to guide professional translators to ensure both accuracy and consistency in translations. Our generic systems make only use of terminologies at inference time.

The remainder of the paper is organized as follows: Section 2 gives details of several terminology injection approaches considered in this work. In addition, we outline a grammatical error correction network that is applied over French translation hypotheses. The experimental framework is presented in Section 3. Results are discussed in Section 4. Finally, we draw conclusions in Section 5.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 842–850
November 10–11, 2021.

2 Terminology Injection

Terminology is typically defined as the technical or special terms used in a business, art, science, or special subject. A high quality asset maintained by language specialists as part of a translation project. It allows to guarantee language consistency, certify translation accuracy and define constraints to human translation.

In recent years there has been significant work proposing methods to integrate such external specialized terminologies into NMT models, each showing different levels of performance when facing terminology injection issues, mainly inference overhead and generalization power.

In this section we describe the two main methods employed for this shared task and illustrate the particularities of each on a common scenario using two English-French terminology entries: *Coronavirus* ⇝ *Coronavirus* and *pneumonia* ⇝ *pneumonie*, in the following translation:

Coronavirus can cause pneumonia
Les coronavirus peuvent causer des pneumonies

Table 1 illustrates training examples of each terminology injection methods evaluated in this work. First row shows the ***base*** configuration where no terminology is employed.

2.1 Placeholders

Our first method incorporates non-terminal tokens into NMT systems, which require modifying the pre-and post-processing of the data, and training the system with data that contains the same placeholders which occur in the test sets (Crego et al., 2016; Michon et al., 2020). Following our example, source and translation terms appearing in the sentence pair are replaced by placeholders adapted to cover a wider variety of cases, and to control morphology to allow generalization power.

The method presented in Michon et al. (2020) allows handling very challenging cases concerning homographs. This is, words (or phrases) that sharing the same form (*i.e. spread*) can occur with multiple meanings or different grammatical functions (*verb* or *noun*). The method predicts the part-of-speech of the target placeholder. Thus, solving the source homograph.

Given that the vast majority of the terminology released for this shared task consists of nouns (single words or phrases) we decided to use a simplified version of the method that only considers using noun placeholders.

Row ***mrk*** in Table 1 illustrates the use of placeholders for our previous training example. Each word form detected as terminology is replaced by two placeholders: The first indicates the part-of-speech of the terminology (in this work always a noun 'N') followed by a unique identifier; the second indicates the set of features conveying the morphology of the noun (masc/fem and sing/plur).

To predict the target morphology of the each term, the NMT model may find it useful to have access to the source word form. Thus, in a second version of the method we incorporate the terminology word form (*Coronaviruses* and *pneumonia*) in the source stream. We denote this version ***mrk+***. It is worth to notice that this second version only improves on the previous when the incorporated source term has sufficiently occurred in training.

Note also that Michon et al. (2020) do not require linguistic information in inference since ambiguities are not resolved in the source placeholders. In contrast, our implementation uses `SpaCy`[1] to obtain part-of-speeches and morphology features of input streams.

Target-side streams of methods *mrk* and *mrk+*, require post-processing to replace target-side placeholders by the final word forms. In practice, for each source-target term pair we encode all possible inflections of the source and target word labelled with the corresponding inflection type (placeholders). Not only does this analysis enable to lexically match any inflected form of the source term, but it can also produce any inflected form of the translation term, ensuring full flexibility in the inflection choice made by the neural network. Table 2 illustrates target word forms for the terminology *pneumonia* ⇝ *pneumonie*.

2.2 Learning to apply constraints

This approach tackles the same problem by learning a copy behaviour of terminology at training time (Song et al., 2019; Dinu et al., 2019; Bergmanis and Pinnis, 2021). The NMT model is trained to incorporate terminology translations when they are provided as additional input in the source sentence. Terminology translations are inserted as inline annotations, expecting the model to learn that such additional words must be copied in the target hypothesis. The authors insert terminology translations in the source sentence either by *appending* the target term to its source version, or by

[1] `https://spacy.io/`

Placeholders (tgt)	Word form
<N> <fem_sing>	pneumonie
<N> <fem_plur>	pneumonies

Table 2: Target word forms and associated placeholders for the term entry *pneumonia* ⇝ *pneumonie*.

directly *replacing* the original term with the target one. Example in row ***app*** of Table 1 illustrates the *append* alternative presented in Dinu et al. (2019).

The approach uses a generic NMT architecture which learns to use an external terminology provided at run-time, thus, showing no inference overhead. However, similarly to the preceding approach, it lacks generalization power as it simply "copies" the term found in the terminology base injected in the source sentence, irrespective of the target hypothesis context. Dinu et al. (2019) argue that in some cases the approach exhibits the ability to inflect translation terms.

Finally, a second version of the method is also illustrated in Table 1, denoted as ***app+***. The target term is injected using its lemma form. Thus, forcing the NMT model to produce the right inflection of the term observed in the source stream. In the example, *pneumonie* must be inflected in its plural form, *pneumonies*. Tokens <b>, <i> and <e> are used to inform the model of the source and target terminology boundaries. Note that, in contrast to placeholder methods, no additional post-processing is required.

2.3 Grammatical Error Correction

As previously stated, placeholder methods allow larger generalization power thanks to the flexibility of the inflection mechanism employed in the translation workflow. However, morphology choices made by the network do not take into account the actual word forms, which was observed to result in a higher number of inflection errors (Michon et al., 2020). To alleviate this problem we add a correction module that performs over the resulting translation hypotheses.

We use a correction module based on Gector (Omelianchuk et al., 2020) with a pretrained multilingual BERT to correct grammatically incorrect French words. The model predicts grammatical features for each word in the translated sentence, allowing only for 3 types of edits:

- Transformation of gender/number

  le [Fem] → la
  le [Plur] → les

- transformation of tense/person of verbs

  avez [3_Plur] → avons
  avez [Ind_Imp] → aviez

- Elision

  le [ELISION] → l'

Table 9 in Appendix B illustrates the vocabulary of tags considered by the model. Once the model predicts whether a word needs to be corrected (and which correction), the final word form is found using a dictionary and the predicted tag. Table 3 illustrates examples of translation hypotheses produced by the NMT model (Hyp) predicted tags for each word (Pred) and corrected hypotheses (Corr). Tag ✓ is used to indicate that no transformation is required.

base	Coronaviruses can cause pneumonia Les coronavirus peuvent causer des pneumonies
mrk	*<N#1> <masc_plur> can cause <N#2> <fem_sing>* *Les <N#1> <masc_plur> peuvent causer des <N#2> <fem_plur>*
mrk+	*<N#1> Coronaviruses <masc_plur> can cause <N#2> pneumonia <fem_sing>* *Les <N#1> <masc_plur> peuvent causer des <N#2> <fem_plur>*
app	*<b> Coronaviruses <i> coronavirus <e> can cause <b> pneumonia <i> pneumonies <e>* *Les coronavirus peuvent causer des pneumonies*
app+	*<b> Coronaviruses <i> coronavirus <e> can cause <b> pneumonia <i> pneumonie <e>* *Les coronavirus peuvent causer des pneumonies*

Table 1: Examples of training streams for the same sentence pair using terms *Coronaviruses* ⇝ *Coronavirus* and *pneumonia* ⇝ *pneumonie* according to each injection method evaluated in this work.

Hyp	...	**le**	épidémie	rapidement	propagée	aux	villes	...
Pred	...	**ELISION**	✓	✓	✓	✓	✓	...
Corr	...	**l'**	épidémie	rapidement	propagée	aux	villes	...
Hyp	...	avec	**le**	fièvre	à	peu	près	...
Pred	...	✓	**Fem_Sing**	✓	✓	✓	✓	...
Corr	...	avec	**la**	fièvre	à	peu	près	...
Hyp	...	atteintes	à	la	mise	en	**quarantaines**	...
Pred	...	✓	✓	✓	✓	✓	**Fem_Sing**	...
Corr	...	atteintes	à	la	mise	en	**quarantaine**	...
Hyp	...	cas	de	COVID-19	**confirmées**	en	laboratoire	...
Pred	...	✓	✓	✓	**Masc_Plur_Past_Part**	✓	✓	...
Corr	...	cas	de	COVID-19	**confirmés**	en	laboratoire	...

Table 3: Examples of word edits performed by the correction model.

3 Experimental Framework

3.1 Corpora

Table 7 in Appendix A provides some statistics on the parallel corpora employed for training our models. It is important to note that all corpora used are out-of-domain. We first filtered out longer sentences and sentences with a significant difference in the number of words between the source and the corresponding translation. All data is pre-processed using the OpenNMT tokenizer[2].

In order to train our correction (GeC) model with additional data, we also use the monolingual (French) corpora made available for the shared task. See Table 8 in Appendix A for detailed statistics of monolingual data.

3.2 Terminology

Table 4 illustrates some examples of the terminology entries released by the organisers of the shared task.

English	French
contagious	contagieux
active cases	cas actifs
confirmed cases	cas confirmés

Table 4: English-French terminology examples.

We note that most terminology entries are composed of several words. Indeed 54.8% of terms are groups of two words, 22.3% contains more than three words and only 22.9% are single words as measured in the source side.

3.3 NMT Engines

All our NMT engines follow the Transformer architecture (Vaswani et al., 2017) implemented by the `OpenNMT-tf`[3] toolkit (Klein et al., 2017). More precisely, we use: Word embedding size: 1024; Number of layers: 6; Number of heads in multi-head self-attention layer: 16; Inner dimension of feedforward layer: 4096; Dropout rate: 0.1; In addition, we use shared embeddings for both the input and output layers. The encoder and decoder use the same BPE units (Sennrich et al., 2016) learned from source and target corpora. We train our MT models using Noam schedule (Vaswani et al., 2017) with 4000 warm-up iterations. To balance between the domains of the training corpora, we use the following sampling distribution over the training corpora:

$$\lambda_\alpha(d) = \frac{q_d^\alpha}{\sum_{d=1}^{n_d} q_d^\alpha}, \quad (1)$$

where q_d is the size of d^{th} corpora, the scalar $\alpha \in [0, +\infty]$ changes the sampling distribution as low α upsamples small corpora and downsamples large corpora while high α favors large corpora over small corpora. In the training of our MT systems, we use $\alpha = 0.5$. Learning is performed over 8 GPUs during $300K$ steps with a batch size of $32K$ tokens per step. During training, we filtered out sentences larger than 250 tokens. We applied label smoothing to the cross-entropy loss with a rate of 0.1. Resulting models are built after averaging the last ten checkpoints of the training process. In inference, we apply a length penalty rate of 0.6.

[2] https://github.com/OpenNMT/Tokenizer

[3] https://github.com/OpenNMT/OpenNMT-tf

3.4 Training NMT

Terminology injection approaches implemented for this evaluation rely on NMT models with the ability to translate input streams with target terms (*app* and *app+*) and using placeholders (*mrk* and *mrk+*). Thus, a key step for our models is the availability of training data with such annotations.

To identify terminology pairs in our training database we :

- Analyse English and French using `SpaCy` to produce part-of-speeches, morphology features, noun phrases and lemmas. Only NPs (single words or phrases) are considered.
- Word align English and French parallel corpora using the `fast_align`[4] toolkit (Dyer et al., 2013).

Terminology pairs are only considered when English and French sides consist of noun phrases and when words in a term are only aligned to words in its counterpart.

Words of the terminology entries identified are replaced by the corresponding tokens (depending on the approach). See Table 1 for examples of sentence pairs with terminology entries. We make sure that a given sentence pair does not exceed 5 terminology entries.

3.5 Training GeC

We used all available French corpora to train our GeC network. To include errors in the French streams we replace some words by any inflection of its base form (lemma). The resulting corpora is then tokenized using wordpiece and passed to the BERT language model for embedding. Error detection and tagging are then performed by the network from subword embeddings. Grammatical features, part-of-speeches and lemmas are performed by the `SpaCy` toolkit. Table 3 illustrates examples of word error correction by our model.

3.6 Terms with Multiple Translations

Note that terms released by the shared task organisers may have multiple translation options (*i.e. quarantine ⇝ quarantaine/mise en quarantaine*). Thus, the right translation must be predicted and injected in the translation hypothesis.

The translation workflow implemented for this evaluation considers the injection of each translation option into the input sentence. This is, when a matched term is built with n different translation options, the original input sentence is copied n times and each translation is injected into one copy. Once all copies have been translated, the one showing the lowest perplexity is selected as measured by the pretrained BERT French language model detailed in section 2.3.

4 Results

Table 5 indicates BLEU[5] (Post, 2018) accuracy results of our NMT systems implementing different terminology injection methods before (second column) and after (third column) grammatical error correction.

System	NMT	`+corr`
`base`	**44.9**	44.8
`mrk`	42.3	**42.7**
`mrk+`	44.9	**45.1**
`app`	45.9	**46.0**
`app+`	**45.9**	**45.9**

Table 5: BLEU score of our NMT systems before (NMT) and after the correction model (`+corr`) measured over the development set.

As it can be seen, the methods that learn to apply constraints (*app* and *app+*) obtain the best performance. Overall, the GeC model succeeds in fixing grammatically incorrect French words. However, a benefit barely reflected by BLEU.

We now evaluate the performance of matching terminology entries over the development input sentences. Note that the same matching method is always applied, detailed in Section 2.1, where input sentences are matched against all possible inflections of source terms. Table 6 illustrates the accuracy of recognized terms. 73 percent of the unrecognized terms are verbs which we choose to not process. We recognized also 234 terms that are not highlighted in the development set (FP), most of them do not interfere with translations.

Accuracy	FN	FP
0.97	0.03	0.21

Table 6: Matching rates of terminology entries measured over the development set. FN and FP scores stand respectively for false negatives (terms not identified) and false positives (wrong terminology identifications).

[4] https://github.com/clab/fast_align

[5] https://github.com/mjpost/sacrebleu

5 Conclusions

We presented SYSTRAN English-to-French submissions to WMT 2021 terminology shared task. All our systems follow the Transformer network architecture enhanced with the ability to dynamically include terminology constraints. Several terminology injection methods were evaluated, showing their ability to effectively injecting terms while producing highly accurate translations.

Acknowledgements

The work presented in this paper was partially supported by the European Commission under contract H2020-787061 ANITA.

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif).

References

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. 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, Aurélie Névéol, 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, Berlin, Germany. Association for Computational Linguistics.

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *CoRR*, abs/1610.05540.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. 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.

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.

Eva Hasler, Adrià de Gispert, 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, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. 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.

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.

Elise Michon, Josep Crego, and Jean Senellart. 2020. Integrating domain terminology into neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3925–3937, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA â†' Online. Association for Computational Linguistics.

Álvaro Peris and Francisco Casacuberta. 2019. A neural, interactive-predictive system for multimodal sequence to sequence tasks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 81–86, Florence, Italy. Association for Computational Linguistics.

Minh Quang Pham, Jitao Xu, Josep Crego, François Yvon, and Jean Senellart. 2020. Priming neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 516–527, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk, Jean-Baptiste Fouet, and Jean Senellart. 2008. First steps towards a general purpose French/English statistical machine translation system. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 119–122, Columbus, Ohio. Association for Computational Linguistics.

Jean Senellart, Christian Boitet, and Laurent Romary. 2003. SYSTRAN new generation: the XML translation workflow. In *Proceedings of Machine Translation Summit IX: Papers*, New Orleans, USA.

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.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. 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'12)*, pages 2214–2218, 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 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.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.

A Corpora Statistics

We experiment with English-French corpora made available via the shared task organisers[6] (Tiedemann, 2012), corresponding to texts from: News commentaries (*news*), European Parliament proceedings (*epps*), United Nations official records and documents (*unpc*), web crawling (*ccrawl*, *pcrawl* and *giga*), In addition we also used the next monolingual French data sets: News Commentary 2019 (*news.19*) and News Commentary 2020 (*news.20*).

Table 7 shows statistics of the parallel corpora used for learning NMT models. Statistics computed after a lightly tokenization (to split-off punctuation). Data sets were previously filtered to discard very long sentences (> 80 words) and with very different number of tokens on either side (fertility > 6 words).

Corpus	Sents (M)	Words (M)		Vocab (K)	
		En	Fr	En	Fr
news	0.2	4.4	5.5	70	75
epps	1.5	41.3	47.7	111	127
ccrawl	1.4	28.6	33.2	486	496
giga	9.4	212.2	259.9	1467	1376
unpc	11.8	256.7	330.3	739	622
pcrawl	92.2	1898.6	2237.2	8110	7757

Table 7: Statistics of parallel corpora used for training NMT. Number of sentences and words are given in millions, and vocabularies in thousands.

[6]Freely available from http://opus.nlpl.eu

Table 8 shows statistics of the monolingual (French) corpora used for learning our GeC model. Statistics computed after a lightly tokenization (to split-off punctuation).

Corpus	Sents (M)	Words (M)	Vocab (K)
news.19	10.2	247.9	955
news.20	9.3	232.5	912

Table 8: Statistics of monolingual corpora used for training GeC. Number of sentences and words are given in millions, and vocabularies in thousands.

B Vocabulary of GeC

Table 9 illustrates the vocabulary of tags considered by our GeC model.

Vocabulary	Example
✓	Ø
<Gender=Masc_Number=Sing>	chiennes → chien
<Gender=Fem_Number=Sing>	chiens → chienne
ELISION	le → l’
<VerbForm=Inf>	avons → avoir
<Mood=Ind_Number=Sing_Person=3_Tense=Pres_VerbForm=Fin>	avoir → a
<Gender=Masc_Number=Plur>	chienne → chiens
<Gender=Masc_Number=Sing_Tense=Past_VerbForm=Part>	avoir → eu
<Number=Plur>	homme → hommes
<Gender=Fem_Number=Plur>	chien → chiennes
<Number=Sing>	hommes → homme
<Mood=Ind_Number=Plur_Person=3_Tense=Pres_VerbForm=Fin>	avoir → ont
<Gender=Masc_Number=Sing_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Tense=Pres_VerbForm=Part>	avoir → ayant
<Mood=Ind_Number=Sing_Person=3_Tense=Imp_VerbForm=Fin>	avoir → avait
<Gender=Masc>	chienne → chien
<Gender=Fem_Number=Sing_Tense=Past_VerbForm=Part>	avoir → eue
<Mood=Ind_Number=Sing_Person=3_Tense=Fut_VerbForm=Fin>	avoir → aura
<Gender=Fem_Number=Sing_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eue
<Gender=Masc_Number=Plur_Tense=Past_VerbForm=Part>	avoir → eus
<Mood=Ind_Number=Sing_Person=1_Tense=Pres_VerbForm=Fin>	avoir → ai
<Gender=Masc_Number=Plur_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eus
<Gender=Masc_Tense=Past_VerbForm=Part>	avoir → eu
<Mood=Ind_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → avons
<Gender=Fem_Number=Plur_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eues
<Mood=Cnd_Number=Sing_Person=3_Tense=Pres_VerbForm=Fin>	avoir → aurais
<Gender=Fem_Number=Plur_Tense=Past_VerbForm=Part>	avoir → eues
<Mood=Ind_Number=Plur_Person=3_Tense=Fut_VerbForm=Fin>	avoir → auront
<Mood=Ind_Number=Plur_Person=3_Tense=Imp_VerbForm=Fin>	avoir → avaient
<Gender=Masc_NumType=Ord_Number=Sing>	cents → cent
<Mood=Ind_Number=Sing_Person=3_Tense=Past_VerbForm=Fin>	avoir → eut
<Gender=Fem_NumType=Ord_Number=Sing>	cents → cent
<Tense=Past_VerbForm=Part>	avoir → eu
<Mood=Ind_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → avez
<Mood=Sub_Number=Sing_Person=3_Tense=Pres_VerbForm=Fin>	avoir → ait
<NumType=Ord_Number=Sing>	cents → cent
<Gender=Masc_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Gender=Fem>	chien → chienne
<Mood=Cnd_Number=Plur_Person=3_Tense=Pres_VerbForm=Fin>	avoir → auraient
<Gender=Masc_NumType=Card_Number=Plur>	quatrième → quatrièmes
<Gender=Masc_NumType=Ord_Number=Plur>	cent → cents
<Mood=Imp_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → ayez
<Mood=Ind_Number=Sing_Person=1_Tense=Imp_VerbForm=Fin>	avoir → avais
<Gender=Fem_NumType=Ord_Number=Plur>	cent → cents
<Mood=Sub_Number=Plur_Person=3_Tense=Pres_VerbForm=Fin>	avoir → aient
<Mood=Ind_Number=Plur_Person=1_Tense=Fut_VerbForm=Fin>	avoir → aurons
<Mood=Ind_Number=Plur_Person=1_Tense=Imp_VerbForm=Fin>	avoir → avions
<Gender=Masc_NumType=Card_Number=Sing>	premières → premier
<Mood=Cnd_Number=Sing_Person=1_Tense=Pres_VerbForm=Fin>	avoir → aurais
<Mood=Ind_Number=Plur_Person=3_Tense=Past_VerbForm=Fin>	avoir → eurent
<Mood=Cnd_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → aurais
<Mood=Ind_Number=Plur_Person=2_Tense=Fut_VerbForm=Fin>	avoir → aurez
<Mood=Ind_Number=Sing_Person=1_Tense=Fut_VerbForm=Fin>	avoir → aurai
<Number=Plur_Tense=Past_VerbForm=Part_Voice=Pass>	avoir→ eus
<Number=Sing_Tense=Past_VerbForm=Part>	avoir → eu
<Mood=Sub_Number=Sing_Person=1_Tense=Pres_VerbForm=Fin>	avoir → aie
<Mood=Ind_Person=3_Tense=Pres_VerbForm=Fin>	neiger → neige
<Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Mood=Sub_Number=Sing_Person=3_Tense=Past_VerbForm=Fin>	avoir → eu
<Mood=Cnd_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → auriez
<Number=Sing_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Mood=Imp_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → ayons
<Number=Plur_Tense=Past_VerbForm=Part>	avoir → eus
<Mood=Ind_Number=Plur_Person=2_Tense=Imp_VerbForm=Fin>	avoir → aviez
<Mood=Imp_Tense=Pres_VerbForm=Fin>	avoir → aie
<Mood=Sub_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → avons
<Mood=Ind_Number=Sing_Person=2_Tense=Imp_VerbForm=Fin>	avoir → avais
<Mood=Sub_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → avez

Table 9: Vocabulary of tags of our GeC model.

TermMind: Alibaba's WMT21 Machine Translation using Terminologies Task Submission

Ke Wang, Shuqin Gu, Boxing Chen, Yu Zhao, Weihua Luo, Yuqi Zhang
Machine Intelligence Technology Lab
Alibaba Group
Beijing, China
{wk258730,shuqin.gsq,boxing.cbx}@alibaba-inc.com,
kongyu@taobao.com,{weihua.luowh,chenwei.zyq}@alibaba-inc.com

Abstract

This paper describes our work in the WMT 2021 Machine Translation using Terminologies Shared Task. We participate in the shared translation terminologies task in English to Chinese language pair. To satisfy terminology constraints on translation, we use a terminology data augmentation strategy based on Transformer model. We used tags to mark and add the term translations into the matched sentences. We created synthetic terms using phrase tables extracted from bilingual corpus to increase the proportion of term translations in training data. Detailed pre-processing and filtering on data, in-domain finetuning and ensemble method are used in our system. Our submission obtains competitive results in the terminology-targeted evaluation.

1 Introduction

Terminology is important for domain-specific machine translation. Each domain has its own terminology, which represents the important and core concepts in the domain. In the workflow of human translation, terminology is an effective method to integrate the knowledge of human translator into machine translations (Wuebker et al., 2016; Cheng et al., 2016; Álvaro Peris et al., 2017).

One line of approach is "hard constraint". The terminology is ensured to appear in the translation by adding constraints in beam search decoding (Hokamp and Liu, 2017; Post and Vilar, 2018). However, the enforcement of terminology constraints tends to reduce the fluency of translation (Hasler et al., 2018), especially when there are multiple constraints or the constraint is noisy (Susanto et al., 2020). Another line of approach is "soft constraint". Training data is augmented with placeholders or additional terminology translations (Arthur et al., 2016; Song et al., 2019; Dinu et al., 2019; Chen et al., 2020; Ailem et al., 2021a).

The above methods assume that the terminology translations are good ones. However, in industry or real world the terminology translations may be noisy (Li et al., 2020). And in the human translation workflows the terminology constraints usually need to be applied hierarchically according to priority. In these scenarios one source term will have more than one translation. Therefore, we are happy to participate in this task and develop the method to deal with 1-to-many term translations in neural machine translation systems.

The structure of the paper is as follows. Section 2 describes the dataset, data pre-processing and selection. We introduce details of our system in Section 3. The experiment settings, terminologies used in training and main results are introduced in Section 4. Finally, we conclude our work in Section 5.

2 Data

2.1 Data Source

For this task, we utilize parallel data from English to Chinese language provided in WMT2021: ParaCrawl v7.1, News Commentary v16, Wiki Titles v3, UN Parallel Corpus V1.0, CCMT Corpus and WikiMatrix. In addition, we also require Chinese monolingual data from News crawl and News Commentary corpora for back translation.

2.2 Data Pre-processing

For all datasets, we tokenize English text with Moses[1] and the Chinese text with Jieba[2] tokenizer. We create a joint source and target BPE vocab (Sennrich et al., 2016) with 40k merge operations using filtered bilingual dataset as described in Section 2.3, resulting in a vocabulary with size of 63K words.

2.3 Data Selection

According to the previous works (Li et al., 2019; Sun et al., 2019), we selected data for training with

[1]https://github.com/moses-smt/mosesdecoder
[2]https://github.com/fxsjy/jieba

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 851–856

Source	Those most at risk of COVID-19 **infection** and serious complications are the elderly and those with weakened immune systems or underlying health conditions like cardiovascular disease, **diabetes**, hypertension, **chronic respiratory disease**, and cancer.
Constraint	diabetes 糖尿病 infection 传染病 | 感染 chronic respiratory disease 慢性呼吸道疾病 | 慢性呼吸系统疾病
Match	Those most at risk of COVID-19 <term tgt=" 传染病 | 感染"> **infection** </term> and serious complications are the elderly and those with weakened immune systems or underlying health conditions like cardiovascular disease , <term tgt=" 糖尿病"> **diabetes** </term> , hypertension , <term tgt=" 慢性呼吸道疾病 | 慢性呼吸系统疾病"> **chronic respiratory disease** </term>, and cancer .
Tag & Mask	Those most at risk of COVID-19 <S> **[MASK]** <C> **传染病 [SEP] 感染** </C> and serious complications are the elderly and those with weakened immune systems or underlying health conditions like cardiovascular disease , <S> **[MASK]** <C> **糖尿病** </C> , hypertension , <S> **[MASK] [MASK] [MASK]** <C> **慢性呼吸道疾病 [SEP] 慢性呼吸系统疾病** </C> , and cancer .
Target	COVID - 19 **感染** 和严重并发症风险最高的是老年人、免疫力低下者或患有心血管疾病、**糖尿病** 、高血压、**慢性呼吸道疾病** 和癌症等基础性疾病的人群。

Table 1: Illustration of the terminology data augmentation.

the following schemes:

- Remove the texts of over 120 tokens.
- Remove bitexts with length ratios greater than 3.
- Remove texts with special HTML tags.
- Remove duplicate bitexts.
- Remove texts with fastText-langid (Joulin et al., 2016b,a), which is an open-source tool for text-based language identification.
- Remove Chinese sentences when the proportion of Chinese tokens is less than 0.8.

3 System Overview

In this section, we will describe the details about the model and techniques of our work. First, we will introduce the terminology data augmentation strategy to improve terminology translation accuracy. Then, different transformer model architectures we adopted in the paper will be depicted. Finally, we will introduce several strategies to train our models for performance improvement.

3.1 Terminology Learning

We use a terminology data augmentation strategy to encourage neural machine translation (NMT) to satisfy terminology constraints. The key point of term translation idea is that when multiple possible terms are encountered, the NMT model is preferred copying the correct terms, and the terms are correctly placed in the output sentence. Encouraged by the work (Chen et al., 2020; Ailem et al., 2021b), we use tags to specify the term constraints in the source sentence. We have given an example in the Table 1. A **Source** sentence could have more than one terms. Each term could have multiple **Constraint**. The source term is indicated as tag <S>, and the pair <C> </C> is used to label target term. Tag [SEP] is used to separate multiple possible target terminologies, when there are 1-m term constraints. Following the work (Ailem et al., 2021b) we mask the source tokens of a term to strengthen the learning of target term tokens. In table 1, term source tokens are marked in red, and the term target tokens are in blue. **Tag & Mask** shows an example. <S> indicates term constraint "infection", but the token "infection" is masked with [MASK]. "infection" 's translations " 传染病" and " 感染" are enclosed by <C> and </C>, separated by [SEP].

The official term table is small. We extract a phrase table from the bilingual training data and filter it as synthetic terms. More details are described in Section 4.2.

3.2 Model Architecture

In our systems, we adopt three different model architectures with Transformer (Vaswani et al., 2017):

- **BIG** Transformer is the Transformer-Base model (Vaswani et al., 2017) with 4096 feed-forward network (FFN) width and 16 attention heads.

- **DEEP** Transformer (Sun et al., 2019) is Transformer-Base model with 20 encoder layers.

- **LARGE** Transformer (Ng et al., 2019) is Transformer-Base model with 8192 FNN inner width.

We use 6 decoder layers for all models. Our models are implemented with open-source toolkit Fairseq (Ott et al., 2019).

3.3 Optimization Strategies

To further improve the translation performance, several common strategies are used to train our models such as Back Translation, Finetuning and Ensemble. The strategies are performed basically sequentially. We use the terminology data augmentation on back translation and fine-tuning datasets to train models.

3.3.1 Back Translation

Back translation is a data augmentation technique to incorporate monolingual data into NMT model. Similar to previous work (Edunov et al., 2018), we use back translation to improve the model performance. We first train a Chinese-to-English Transformer-Deep NMT model based on bilingual training dataset. The NMT model is applied to translate Chinese monolingual corpus to English. The pseudo parallel corpus is used to train models together with the bilingual training dataset.

3.3.2 Finetuning

Previous study (Sun et al., 2019) demonstrate that fine-tuning a model on in-domain data effectively improve the model performance. For the term translation task, two fine-tuning datasets are used in our works. We use two kinds of finetuning datasets to train the model sequentially.

Base FT We use all the previous English $\rightarrow$ Chinese development and test dataset as fine tuning corpus, including WMT2017 development data, WMT2017 test data, WMT2018 test data, WMT2019 test data and WMT2020 test data.

In-domain FT To use in-domain dataset to fine tune the model, we perform data selection on out-of-domain corpus based on in-domain n-gram match. The key idea is to select sentence pairs from the large out-of-domain corpus that are similar to the in-domain data. We use the bilingual training data as the out-of-domain corpus and WMT2021 term development dataset as the in-domain corpus. We extract 1-3grams from the in-domain and out-of-domain dataset. After exclude the ngrams from the out-of-domain data, the left in-domain ngrams are applied to match relevant sentence from the bilingual training.

In our work, we use source and target to select in-domain dataset respectively and finally the two sets are combined to train the model.

3.3.3 Ensemble

Model ensemble is an effective strategy widely used in real-world tasks. At each step of translation prediction, it combines the predicted probabilities of different models. We use the log-avg strategy to ensemble the different NMT models. The model diversity is an important factor for ensemble. We have trained three Transformer models with different architectures including the variants of Transformer-BIG, Transformer-DEEP and Transformer-LARGE.

4 Experiments

4.1 Setups

Our models are implemented in Fairseq Library[3]. All the single models are trained based on 4 NVIDIA P100-PCIe GPUs, each with 16 GB memory. The models are optimized with Adam algorithm (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We set max learning rate to 0.001 when training a single model from scratch and 0.0007 when fine-tuning the model. The batch size is set to 2048 tokens per GPU. The 'update-freq' parameter in Fairseq is set to 16 when training a single model from scratch and 4 when fine-tuning the model. The dropout (Gal and Ghahramani, 2016) probabilities are set to 0.1 in all experiments. We select the checkpoint with the best BLEU score on development set as the final checkpoint in each training. Evaluation of results focus on translation accuracy and term translation consistency. We evaluate translation accuracy with SacreBLEU (Post, 2018), which is a

[3]https://github.com/pytorch/fairseq

System	Model	BLEU	Exact-Match Accuracy	Window Overlap Accuracy (2/3)	1-TERm Score
Baseline NMT	LARGE	37.8	65.89	16.52/16.30	36.48
Data Selection	BIG	36.09	71.28	15.82/16.57	30.08
	DEEP	35.85	74.76	17.01/17.62	29.74
	LARGE	36.17	69.23	14.94/15.33	30.91
	+Ensemble	**38.22**	**74.52**	**17.47/17.56**	**33.00**
+Back Translation	BIG	37.72	73.92	17.28/17.71	33.33
	DEEP	37.74	73.92	17.55/18.05	33.85
	LARGE	37.50	72.36	15.97/16.53	32.68
	+Ensemble	**39.39**	**75.60**	**17.87/18.62**	**33.90**
+Base FT	BIG	38.12	71.86	17.57/18.14	34.68
	DEEP	38.17	72.72	17.32/18.18	33.74
	LARGE	40.97	72.95	17.26/18.40	38.03
	+Ensemble	**41.43**	**75.72**	**18.91/19.89**	**38.17**
+In-domain FT	BIG	39.12	71.63	17.09/17.71	36.25
	DEEP	38.33	73.08	17.48/18.25	34.60
	LARGE	41.11	72.72	17.04/18.24	38.48
	+Ensemble	**41.71**	**76.68**	**18.88/19.88**	**39.05**

Table 2: Evaluation results on the WMT2021 English → Chinese development set.

case-sensitive detokenized BLEU. Terminology-targeted metrics (Anastasopoulos et al., 2021) is used to term translation consistency, including exact-match accuracy, window overlap metric and terminology-biased Translation Edit Rate (TERm)[4]. The exact-match accuracy is defined as the ratio between the number of matched source terms and the total number of source terms. The window overlap metric is to evaluate the position accuracy of each target term in translation. The TERm, a metric based on TER (Snover et al., 2006), focuses on penalizing errors related to terminology tokens.

4.2 Terminologies

In order to increase the proportion of term translations in training data, we extract phrase tables from bilingual training corpus to create synthetic term translations. First, we use FastAlign (Dyer et al., 2013) to generate word alignments. Second, based on the word alignments we extract a phrase table by using moses (Koehn et al., 2007) with default settings. We use count-based pruning (Zens et al., 2012) and fastText-langid (Joulin et al., 2016b,a) to filter the phrase table. The count threshold is set to 200. Finally, the term table for the terminology data augmentation is obtained by combining the English → Chinese term table from WMT2021 and the filtered phrase table. The target terms corresponding to the same source term are separated by '|'. The term table contains 1-to-1 and 1-to-many term pairs. The term information with tags will be added into source sentences when they match, as shown in Table 1. 15.4% of the training sentences with the term information. We have used only the official terms from WMT 2021 for the test and dev datasets.

4.3 Results

Table 2 shows the English → Chinese translation results on WMT2021 terminologies development dataset, including BLEU, exact-match accuracy, window overlap accuracy (2/3) and 1-TERm Score. We train multiple single models in each settings and report the best BLEU scores in Table 2. The baseline is the LARGE transformer model using the bilingual training data. Our models using terminology data augmentation are called Term model. Ensemble models of each step consist of 3 single models: BIG, DEEP and LARGE models. As shown in Table 2, the LARGE Term model using the bilingual dataset boosts the exact-match

[4]https://github.com/mahfuzibnalam/terminology_evaluation

accuracy from 65.89 to 69.23. Under each setting, the performance of the ensemble Term models is higher than that of the best single Term model by a BLEU score of 0.46 to 2.05. After adding back translation, we improved the BLEU score to 39.39 and the exact-match accuracy to 75.6 on ensemble models. The base FT can achieve 2 BLEU and 4.3 1-TERm score improvements on ensemble models. After applying In-domain FT, We achieve 0.96 exact-match accuracy and 0.88 1-TERm score improvements on ensemble models.

Considering the effectiveness of fine-tuning, we use WMT2021 development data to fine tune the model after completing 100 steps. In our final submission, we selected sentences with the higher probability from the translations of the ensemble Term model and the ensemble NMT model.

5 Conclusion

This paper presents the submissions by Alibaba for WMT 2021 English to Chinese translation terminologies task. We have applied a terminology data augmentation method to integrate term translations into NMT systems. We also used a series of data filtering strategies, fine-tuning and ensemble methods to improve the system performance. Experimental results show the method can improve terminologies translation performance.

6 Acknowledgments

This work is supported by National Key R&D Program of China (2018YFB1403202).

References

Melissa Ailem, Jinghsu Liu, and Raheel Qader. 2021a. Encouraging neural machine translation to satisfy terminology constraints. *CoRR*, abs/2106.03730.

Melissa Ailem, Jinghsu Liu, and Raheel Qader. 2021b. Encouraging neural machine translation to satisfy terminology constraints. *arXiv preprint arXiv:2106.03730*.

Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, Vassilina Nikoulina, et al. 2021. On the evaluation of machine translation for terminology consistency. *arXiv preprint arXiv:2106.11891*.

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. Association for Computational Linguistics.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization. Main track.

Shanbo Cheng, Shujian Huang, Huadong Chen, Xinyu Dai, and Jiajun Chen. 2016. Primt: A pick-revise framework for interactive machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1240–1249.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. *CoRR*, abs/1906.01105.

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.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29:1019–1027.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *NAACL-HLT (2)*, pages 506–512. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *CoRR*, abs/1704.07138.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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 companion volume proceedings of the demo and poster sessions*, pages 177–180.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Huayang Li, Guoping Huang, Deng Cai, and Lemao Liu. 2020. Neural machine translation with noisy lexical constraints. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1864–1874.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

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, New Orleans, Louisiana. 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 the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with levenshtein transformer. *CoRR*, abs/2004.12681.

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.

Joern Wuebker, Spence Green, John DeNero, Saša Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Berlin, Germany. Association for Computational Linguistics.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

FJWU participation for the WMT21 Biomedical Translation Task

Sumbal Naz[1], **Sadaf Abdul Rauf**[1,2] **and Sami ul Haque**[3]
[1] Fatima Jinnah Women University, Pakistan
[2] Univ. Paris-Saclay, LISN-CNRS, France
[3] National University of Science and Technology, Pakistan
{sadaf.abdulrauf,sumbalnaz01}@gmail.com

Abstract

In this paper we present the FJWU's system submitted to the biomedical shared task at WMT21. We prepared state-of-the-art multilingual neural machine translation systems for three languages (i.e. German, Spanish and French) with English as target language. Our NMT systems based on Transformer architecture, were trained on combination of in-domain and out-domain parallel corpora developed using Information Retrieval (IR) and domain adaptation techniques.

1 Introduction

Due to vast availability of multilingual information, Neural Machine Translation (NMT) systems have achieved remarkable growth over Statistical Machine Translation (SMT) systems. Although the amount of training resources has significantly increased in the past few years but availability of large in-domain parallel data is still a challenging task. Performance of NMT system may quickly degrade as soon as the application domain deviates from training domain. Domain adaptation (Koehn and Schroeder, 2007) is a promising active research topic to enhance the translation quality when faced with data scarcity issues. In domain adaptation, initially large amount of parallel out-of-domain corpora is utilized for training NMT models and then fine-tuning is performed on small in-domain data for adapting to novel domains (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015). Fine-tuning does not require building system from scratch, instead it is fast and efficient method of integrating in-domain data. An NMT model already trained on general domain data is further fine-tuned on in-domain data with less time and effort (Chu et al., 2017; Hira et al., 2019). Training MT systems on back-translated data is a proven domain adaptation method (Abdul Rauf et al., 2020; Sennrich et al., 2015), where synthetic parallel data is combined with original data to generate large in-domain training corpus. In addition, information retrieval (IR) technique to extract relevant sentences from out-of-domain corpus has shown promising results to overcome data scarcity (Naz et al., 2020).

NMT system incorporating multiple languages into single model is known as multilingual NMT (MNMT) (Dabre et al., 2020). Multilingual NMT systems are gaining popularity due to effective use of available resources and boosting translation quality with Translation Knowledge Transfer (Pan and Yang, 2009).

In this paper, we present study on adapting MNMT systems (Multiway many-to-one) for translating English (EN) language from French (FR), German (DE) and Spanish (ES) using fairseq (Ott et al., 2019) implementation of Transformer model. Our main focus is to investigate the effect on EN translation in Biomedical domain using multilingual NMT systems. We have also explored the domain adaptation for fine-tuning of bilingual NMT models into multilingual NMT models using out-domain and in-domain corpora. Furthermore, we show the effectiveness of utilizing in-domain data generated through IR techniques (Naz et al., 2020) by training a NMT system on combined parallel in-domain data. We also compare in-domain multilingual and bilingual models.

The remainder of this paper is organized as follows. Section 2 introduces the literature review followed by corpus processing in Section 3. Section 4 presents experiments and results. In Section 5, we conclude the findings of our work.

2 Literature Review

MNMT models tend to acquire knowledge from more than one language which helps in generalization and in building systems for low resource languages. MNMT models may help in miti-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 857–862
November 10–11, 2021.

Corpus	DE/EN	ES/EN	FR/EN
In-domain training data			
UFAL	2.6 M	631 K	2.6 M
Scielo Health	-	124 K	9 K
Scielo Biological	-	581 K	-
EDP	-	-	3 K
Medline Titles	-	285 K	612 K
Medline Abstracts	18 K	66 K	46 K
EMEA	1.10 M	1.09 M	1.09 M
In-domain IR training data			
News Commentary-IR2	-	-	65 K
WikiPedia-IR2	-	-	84 K
Out-domain training data			
UFAL	30.9 M	74.8 M	73.3 M
UFAL Dictionary	733 K	544 K	744 K
Scielo	-	433 K	-
UN	-	21.9 M	25.8 K
Development data			
Medline18	321	239	311
Medline19	439	437	400
Test data			
Medline20	409	466	479

Table 1: Sentence Pairs Used for Training, Development and Testing of MNMT models (K stands for "Thousand" and M stands for "Million")

gating the problem for resource poor languages (Dabre et al., 2020), where limited training data is available. Tubay and Costa-jussÃ (2018) submitted their NMT systems for English translation with multi-source similar languages including Portuguese, French and Spanish showing improvement of 6 BLEU points over single source NMT system. Soares and Krallinger (2019) also built NMT systems using two of the Romance languages, Spanish and Portuguese for translating into English language. For domain adaptation in NMT, fine-tuning models on in-domain parallel text is a common and effective approach (Peng et al., 2020). We assume that, the same can be used for training multilingual (many-to-one) NMT models. Chu and Dabre (2019) focused on fine-tuning MNMT models for domain adaptation, they initially trained different MNMT models using single domain and then further fine-tune on multi-domain corpora with mixed (combination of out-domain and in-domain) corpora.

3 Corpus Pre-processing

This section describes parallel corpora used in training and evaluation of our models. Statistics of train, development and test data are presented in Table 1. Main sources of data were provided by WMT21 Biomedical Translation Task. Data sources include:

- Medline abstracts and titles in-domain corpora consists of scientific publications (Bawden et al., 2019). We used datasets available for DE/EN, ES/EN and FR/EN provided by WMT. These datasets are aligned through Bilingual Sentence Aligner[1] (Moore, 2002).
- EDP are the in-domain texts of scientific publications available for FR/EN language pair only (Neves et al., 2018).
- EMEA provides in-domain biomedical parallel corpus of documents related to medicinal products (Tiedemann, 2012). We used corpora provided for DE/EN, ES/EN and FR/EN language pairs.
- Scielo in-domain corpus provided by WMT comprises of abstracts and titles in biological and health sciences domain (Neves et al., 2016). We used datasets provided for FR/EN and ES/EN language pairs.

  We also used Scielo full text corpus of scientific articles as general domain data set provided for ES/EN language pair (Soares et al., 2018).
- UFAL Medical Corpus provides various in-domain medical texts and out-domain corpus sources including dictionaries (Jimeno Yepes et al., 2017). We included corpora provided for DE/EN, ES/EN and FR/EN language pairs.
- United Nations (UN) parallel corpus comprises of official records in general domain (Ziemski et al., 2016) . We used sources provided for ES/EN and FR/EN language pairs.

News Commentary[2] and WikiPedia[3] in-domain IR corpora are used. These corpora are extracted using data selection based on IR approach (Abdul-Rauf et al., 2016) by using Medline titles as queries

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52608

[2] http://opus.nlpl.eu/News-Commentary-v14.php

[3] http://opus.nlpl.eu/Wikipedia-v1.0.php

		ID	Train Set	DE → EN	ES → EN	FR → EN
Multilingual	System I	M1	In-domain	26.96[3]	39.12[3]	30.23
		M2	M1⇒Medline	27.22[2]	39.40[1]	33.79[1]
		M3	M2⇒Indomain+IR-2	27.38[1]	39.36[2]	32.07[2]
Multilingual	System II	M4	Out-domain	20.97	29.75	24.86
		M5	M4⇒In-domain	26.40	38.87	30.94[3]
		M6	M5⇒Medline	26.40	38.74	**34.73**
Bilingual	System III	M7	In-domain	**27.40**	**41.60**	30.20

Table 2: Bilingual and Multilingual DE/ES/FR → EN Transformer models and their BLEU scores for Medline20 test-sets for three language directions (DE→EN, ES→EN and FR→EN. (M here stands for 'Model'). Superscripts denote the runs submitted.

for retrieving related biomedical domain sentences. From experiments conducted by (Naz et al., 2020), corpora with top-2 best sentences gave good results in training NMT models for biomedical domain.

Medline18 and 19 testsets are used as development set. We used Medline20 testset provided by WMT20 (Bawden et al., 2020) as initial test sets to determine quality of our translation models. Preprocessing of data include tokenization and learning joint Byte Pair Encoding (BPE) (Sennrich et al., 2016) using sentencepiece[4] with a vocabulary size of 32K over in-domain corpus and encoding all available corpora with learned BPE.

4 Experiments and Results

In this section we present details of experimentation along with training configurations.

4.1 Training and Parameters

We employed Fairseq toolkit to train MNMT systems for (German, Spanish, French) → English translation. We used Transformer architecture and followed similar configuration parameters for out systems as reported in original paper (Vaswani et al., 2017). Batch size of 4K words and Adam optimizer was used in all experiments. Training was done till convergence and stopped if no improvement was noted in BLEU scores on development sets for 2-3 consecutive checkpoints. Fine-tuned models were trained for 150K steps unless early stopping is employed based on bleu score convergence.

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

4.2 NMT Models

We have categorized our experiments into 3 classes based on the corpora and training technique used. I) Multilingual models trained using all in-domain corpus and fine-tuned on Medline and IR. II) Multilingual models trained on all out-domain corpus and fine-tuned on all in-domain and Medline corpus. III) Bilingual models trained on all in-domain corpus. Results of all experiments are depicted in Table 2. BLEU score for all models is calculated using Sacrebleu (Post, 2018) on Medline20 test-set for German-English, Spanish-English and French-English.

For System I:

- $M1$: this is trained on all in-domain parallel corpus with a total size of 3.71M (DE-EN), 2.77M (ES-EN), 1.78M (FR-EN) sentences. Best BLEU score of 39.12 on Medline20 test-set was achieved for ES→EN as it has high rate of Medline sentences (66K) as compared to FR→EN (46K) and DE→EN (18K).
- $M2$: this model derived from $M1$ by further tuning it on Medline corpus for domain adaptation which resulted in significant increase in BLEU score of +3.56 for FR→EN as compared to previous model ($M1$). An increase of +0.26 BLEU for DE→EN and +0.28 BLEU for ES→EN is achieved.
- $M3$: $M2$ was further fine-tuned on IR corpus for FR→EN language pair but we observe no significant improvements in term of BLEU score on out test-set for all combinations of languages. IR corpus was extracted

from News commentary and Wikipedia parallel corpora. Apparently these corpora are far in language jargon from the traditional Medline tests, so we see no apparent gain. It is pertinent to note that IR data was only available for FR-EN thus the in-domain training corpus was used for other language pairs.

For System II:

- $M4$: this model is trained on all out-domain parallel corpus with a total size of 3.82M (DE-EN), 10.6M (ES-EN) 8.33M (FR-EN) sentences. Highest BLEU score of 29.75 is achieved with ES→EN test set. As the model is mainly trained on out-domain corpora, the huge difference in score is visible as compared to previous models. When compared with models trained on in-domain we see significant loss in BLEU scores for our current model.

- $M5$: previous model is fine tuned on in-domain corpus that shows substantial improvements over baseline model. A gain of +5.43 points for DE→EN, +9.12 points for ES→EN and +6.08 points for FR→EN was achieved. This clearly indicates that fine-tuning is an effective method for improving quality of multilingual NMT.

- $M6$: $M5$ is further fine tuned on Medline corpus yielding an improvement of +3.79 points for FR→EN giving best score of **34.73** among all models . No significant improvement in DE→EN and ES→EN is observed.

For System III:

- $M7$: Represents the bilingual models trained on all in-domain corpus. Comparing with the multilingual models; ES→EN achieved the best score of `41.60` BLEU points in bilingual mode. Bilingual DE→EN results are comparable to the multilingual systems whereas for FR→EN multilingual systems majorly outperformed the bilingual systems. Interestingly, ES→EN had more medline corpus as compared to other two. The three language pairs that we work on are not similar and thus do not have too much to gain from each other. Introducing other romance languages in the systems might lead to better performance for French and Spanish. The factor of training corpus imbalance is also playing it's part, we intend to employ better sampling strategies for multilingual systems in future.

5 Conclusion

In this paper we have described our system submissions at WMT21 biomedical shared translation task under FJWU's submission. For our submission we trained multilingual NMT systems for German, Spanish and French languages with English as target language. We focused on utilizing in-domain and out-domain parallel corpora and domain adaptation techniques for training multilingual NMT systems. We showed that, domain adaptation using fine-tuning of multilingual NMT model can be a reasonable alternative to achieve good translation quality for novel domains.

Acknowledgments

This study is funded by the National Research Program for Universities (NRPU) by Higher Education Commission of Pakistan (5469/Punjab/NRPU/R&D/HEC/2016).

References

Sadaf Abdul Rauf, José Carlos Rosales Núñez, Minh Quang Pham, and François Yvon. 2020. Limsi @ wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 803–812, Online. Association for Computational Linguistics.

Sadaf Abdul-Rauf, Holger Schwenk, Patrik Lambert, and Mohammad Nawaz. 2016. Empirical use of information retrieval to build synthetic data for smt domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):745–754.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova.

2020. Findings of the wmt 2020 biomedical translation shared task: Basque, italian and russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation*, pages 660–687, Online. Association for Computational Linguistics.

Chenhui Chu and Raj Dabre. 2019. Multilingual multi-domain adaptation approaches for neural machine translation. *arXiv preprint arXiv:1906.07978*.

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, Vancouver, Canada. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Noor-e Hira, Sadaf Abdul Rauf, Kiran Kiani, Ammara Zafar, and Raheel Nawaz. 2019. Exploring transfer learning and domain data selection for the biomedical translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 156–163, Florence, Italy. Association for Computational Linguistics.

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, 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.

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.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Da Nang, Vietnam.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proc. AMTA'02*, Lecture Notes in Computer Science 2499, pages 135–144, Tiburon, CA, USA. Springer Verlag.

Sumbal Naz, Sadaf Abdul Rauf, Noor-e Hira, and Sami Ul Haq. 2020. Fjwu participation for the wmt20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 849–856, Online. 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: Shared Task Papers*, pages 324–339, Belgium, Brussels. 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'16)*, pages 2942–2948, Portorož, Slovenia. European Language Resources Association (ELRA).

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Wei Peng, Jianfeng Liu, Minghan Wang, Liangyou Li, Xupeng Meng, Hao Yang, and Qun Liu. 2020. Huawei's submissions to the wmt20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 857–861, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

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.

Felipe Soares and Martin Krallinger. 2019. Bsc participation in the wmt translation of biomedical abstracts. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 177–180, Florence, Italy. 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)*. European Language Resource Association.

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).

Brian Tubay and Marta R. Costa-jussÃ. 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, Volume 2: Shared Task Papers*, pages 678–681, Belgium, Brussels. 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.

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'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

High frequent in-domain word segmentation and forward translation for the WMT21 Biomedical task

Bardia Rafieian and Marta R. Costa-jussà
TALP Research Center, Universitat Politècnica de Catalunya, Barcelona
{bardia.rafieian,marta.ruiz}@upc.edu

Abstract

This paper reports the optimization of using the out-of-domain data in the Biomedical translation task. We firstly optimized our parallel training dataset using the BabelNet in-domain terminology words. Afterward, to increase the training set, we studied the effects of the out-of-domain data on biomedical translation tasks, and we created a mixture of in-domain and out-of-domain training sets and added more in-domain data using forward translation in the English-Spanish task. Finally, with a simple bpe optimization method, we increased the number of in-domain subwords in our mixed training set and trained the Transformer model on the generated data. Results show improvements using our proposed method.

1 Introduction

Domain adaptation is one of the known challenges in Machine Translation since NMT(neural machine translation) models are susceptible to the training data (Koehn and Knowles, 2017). To say, NMT models perform poorly for domain-specific translation when trained on large out-resource data (Chu and Wang, 2018). As a result, due to the limitations of specific domain data, domain adaptation strategies help NMT models by increasing the parallel corpora. There have been several tasks to address domain adaptation which recently, in (Sato et al., 2020) they proposed a vocabulary adaptation to fine-tune the embedding layers of the NMT model by projecting general word embeddings induced from monolingual data in a target domain onto a source-domain embedding space to improve translation score. On the other hand, augmenting bilingual training data with forwarding and backward translation improves the in-domain translation quality (Nayak et al., 2020). Inspired by mentioned ideas, in this work, we implemented our strategy by two essential steps: 1) collecting and augmenting data by forwarding translation and then tuning it using Babelnet to include biomedical sentences 2) Implementing subwords bpe optimization on the train set to study the adaptation of out-of-domain data in the biomedical task. After that, We selected the transformer model (Vaswani et al., 2017) to train our system in different experimental settings. The remainder of the paper is organized as follows. In Sec. 2 we describe data collection and preparation. Sec. 3 explains our bpe optimization strategy to adapt out-of-domain data in the biomedical task. Sec. 4 shows our experimental setups and evaluation results, and finally, we conclude and discuss future works in the Sec. 5.

2 Data production

One of the critical topics in machine translation (MT) is selecting and fitting well-organized domain-relevant data (Wang et al., 2018). This section describes our data preparation approach to tune, clean, and optimize data for our translator model. The details of the dataset are described in the section 4.

2.1 In-domain dataset tuning

The gathered in-domain data is not well-tuned for the biomedical domain, so that we extracted a list of biomedical terms(word level) using the BabelNet API (Navigli and Ponzetto, 2012) by referring to the ”biomedical” tags in the BabelNet: bio-science, technology, medical practice, medical specialty, neurology, and orthopedics. To address it, we gathered a total of 5,800 biomedical terms for both English and Spanish languages. Secondly, we selected the sentences which specifically contain biomedical words. The outcome holds in-domain parallel data which each sentence at least carries a related biomedical term. Algorithm 1 shows our approach to select in-domain sentences.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 863–867
November 10–11, 2021.

Result: *indomain parallel dataset*
dataset-tuning;
initialization;
input(*EN bio words*, *ES bio words*);
input(*standard en es parallel*)
init(*opt en es parallel*);
for
sentence 1 *and sentence* 2 *in standard en es parallel* :
do
if(*any token sentence* 1 *in* (*EN bio words*))
and (*any token sentence* 2 *in* (*ES bio words*)) :
OptimizedEnEsParallel.append
(*sentence* 1 *and sentence* 2)
end
return(*OptimizedEnEsParallel*)

Algorithm 1: Optimizing the parallel corpus using BabelNet; selecting sentences that contain at least one token in-domain word

2.2 In-domain forward translation

Considering a translation task of L1 → L2, where L1 has more significant monolingual data than L2, a forward translation translates the L1 to L2 and uses the translated L2 to recreate a synthetic parallel corpus. It has been widely reported that forward and back translation brings significant results. (Bogoychev and Sennrich, 2019). We benefited from this fact and produced bilingual data from the English source, which did not have any target or good target parallel translation. However, to ensure the availability of in-domain data, we first passed the previous step on the available monolingual side. Then we translated the source side using our MT model and added bilingual data for retraining. Finally, we merged the in-domain and out-of-domain parallel corpus to achieve a bigger train set.

3 Subword BPE optimization

Byte Pair Encoding, or BPE, is a subword segmentation algorithm that encodes rare and unknown words as sequences of subword units by merging the most frequent consecutive byte pair into a new subword (Sennrich et al., 2015). Since we enriched the train set with out-of-domain data, We propose "bpe-terms in-domain optimization" to handle open vocabulary problems and enhancing the morphology when out-of-domain data is available. Consequently, increasing the frequency of in-domain words in the subword bpe training raises the chance of having in-domain words in the vocabulary. As a result, out-of-domain data will not affect the quality of the model on translating the in-domain words, while they let the model learn on an enormous corpus. We performed this strategy by first learning the subwords on 10x duplicated in-domain parallel sentences with a size of eight million mixed with smaller out-of-domain corpora (no duplication) and then applying the trained subword model on the standard-sized corpus. After that, we expect to have the biomedical in-domain words directly translated to the target language without breaking them into subwords.

4 Experiments

Experiments illustrated in this section study the effects of the out-of-domain data on in-domain(biomedical) translation task as well as the possibility of adapting it by performing a tuned subword-bpe segmentation algorithm 3 to improve the translation quality. We split this section into four parts which start with data collection and prepossessing. Then, we describe the training system and, finally, the evaluation scores of the competition.

4.1 Data collection

We rely on the WMT21 official webpage to collect the (en/es) parallel in-domain data. Out of the provided resources, in particular, for the in-domain train set, we selected UFAL, Pubmed, Medline, IBECS (Villegas et al., 2018) and UNcorpus (Ziemski et al., 2016) along with the OPUS collection (Tiedemann, 2012). Next, we cleaned the data by removing empty lines, duplicates, and very short and long sentences. Also, to perform our experiments on out-of-domain data, we collected the parallel sentences provided from the same WMT21 official website.

4.2 Data preprocessing

To prepare our data for training, we followed the standard pipelines by performing normalization, tokenization, and removing words that contain non-alphabetic characters using Moses (Koehn et al., 2007). Then, we removed concise and long sentences by keeping the thresh-hold between 2 and 30 words for each sentence and implemented the strategy described in section 2 to select in-domain sentences. As a report, we collected 6,855,049 in-domain and added 1,965,824 out-of-domain parallel data (English/Spanish). We also translated 1,558,834 in-domain UFAL monolingual English data to Spanish and added it to our bilingual corpus for retraining the en/es model.

4.3 Training on optimized segmented data

Our method focuses on data preparation and investigates how the out-of-domain data affects the BLEU score. We imply that tuning the vocabulary of subwords would improve the accuracy of the in-domain translation(biomedical) even though some of the data is out of the domain.

The two crucial factors applied in our experiments are preprocessing the parallel corpus with BabelNet, and tuning the learning step of subwords to adapt out-of-domain data. following the strategy, four experiments have been done with two different trainsets, in-domain and mixture of in-domain and out-of-domain data:

1. In the first experiment, we used word-level data in both the source and target sides to evaluate the impact of out-of-domain usage in an in-domain task.

2. In the second experiment, we applied subword-bpe level on both source and target side with shared embeddings; however, the data were preprocessed by using Babelnet (described in section 2) to adjust the in-domain sentences in the train set for all the experiments.

3. We used the same strategy as the second experiment but with applying BPE-dropout (Provilkov et al., 2019) on both the source and target side of the data.

4. The last experiment was carried out by using tuned in-domain subword level data on both source and target sides as explained in the section 3.

In all experiments, we trained baselines on word-level and subword-bpe level to measure the proposed methods.

We selected a vocabulary size of 50k tokens and trained the data by the Transformer model with its default parameters using Open-nmt (Klein et al., 2017) neural machine translation framework.

4.4 Evaluation and results

The evaluation has been done on WMT18 and WMT19 test sets based on the BLEU score. We compared the trained models with word-level, standard subword bpe level, bpe drop out and tuned subword bpe level of the parallel corpus in the trainset to follow our experiments. We also studied the results with three types of trainsets:

- in-domain
- fair mixture of in-domain and out-of-domain sentences
- an unfair mixture of in-domain and out-of-domain with more in-domain sentences

We started and continued each training until it accomplished the best BLEU score on the validation set. We realized that using bpe dropout in the trainset gives worse results than the standard bpe level in terms of the BLEU score. Also, as expected, the worst results belong to word level and hybrid wordlevel+subword level trainset. On the other hand, using out-of-domain data in an in-domain task caused a dramatic drop in the BELU score. In this regard, there was a slight improvement in BLEU score by increasing the frequency of biomedical words in the mixture of in-domain and out-of-domain trainset in both fair and unfair distribution of each domain sentence. For WMT21 competition, we selected the models which achieved the highest scores in the wmt18 and wmt19 en2es and es2en test sets.

Table 1 describes our (en2es) results on a mixture of 2.7 million in-domain + 1.7 million out-of-domain parallel sentences (described the data in the section 2). As well, Table 2 shows the results on 2.7 million in-domain parallel sentences and also a mixture of 8 million in-domain + 1.7 million out-of-domain parallel data (all of that data). Similarly, we show the (es2en) results in the tables 3 and 4

5 Conclusion and future works

This work presented a method to adapt out-of-domain data in an in-domain(biomedical) task to improve the BLEU score. We tuned the parallel data by BabelNet, then found and increased the frequency of biomedical words in subword-learning to raise the weight of in-domain words in the vocabulary. Our results obtained in a different mixture of datasets show that our method improves the BLEU score compared with the standard subword-bpe approach. In the future, we plan to extend our approach to more low-resource languages and domains. Moreover, we plan to increase out-of-domain data and configure the frequency of in-domain words based on the domain type.

Acknowledgements

This work was supported by the project ADAVOICE, PID2019-107579RB-I00 / AEI / 10.13039/501100011033.

Dataset: 2.7m indomain+ 1.7m out-of-domain		
EXP type	**wmt18**	**wmt19**
Word level indomain+out-of-domain	35.0	36.6
Word level Indomain+ subword level out-of-domain	34.5	36.1
Subword level indomain+ subword level out-of-domain (baseline)	35.6	42.4
10x freq subword indomain+subword out-of-domain (our approach)	**39.8**	**42.7**
bpe dropout indomain + bpe dropout out-of-domain	38.5	41.9

Table 1: en2es BLEU score results on hybrid dataset using different word segmentation approaches, word level, hybrid, standard bpe, bpe dropout and tuned subword bpe

	Dataset: 2.7m indomain		**Dataset: 8m in + 1.7m out**	
EXP type	**wmt18**	**wmt19**	**wmt18**	**wmt19**
subword bpe in domain (baseline)	39.8	42.1	**40.1**	42.8
10x freq subwords indomain (our approach)	**39.9**	**42.2**	39.2	**43.0**
bpe dropout	39.7	39.2	37.1	41.7

Table 2: en2es BLEU score results on solid indomain and eight million hybrid datasets using different word segmentation approaches, word level, hybrid, standard bpe, bpe dropout and tuned subword bpe

Dataset: 2.7m indomain+ 1.7m out-of-domain		
EXP type	**wmt18**	**wmt19**
Word level indomain+out-of-domain	NA	NA
Word level Indomain+ subword level out-of-domain	NA	NA
Subword level indomain+ subword level out-of-domain (baseline)	38.1	43.23
10x freq subword indomain+subword out-of-domain (our approach)	**39.6**	**43.3**

Table 3: es2en BLEU score results on hybrid dataset using different word segmentation approaches, word level, hybrid, standard bpe, bpe dropout and tuned subword bpe

	Dataset: 2.7m indomain		**Dataset: 8m in + 1.7m out**	
EXP type	**wmt18**	**wmt19**	**wmt18**	**wmt19**
subword bpe in domain (baseline)	**42.1**	**44.0**	**43.0**	44.1
10x freq subwords indomain (our approach)	41.9	43.6	42.3	44.1

Table 4: es2en BLEU score results on hybrid indomain+out-of-domain dataset and unfair distribution.

References

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *CoRR*, abs/1911.03362.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *CoRR*, abs/1806.00258.

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, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. 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.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Prashant Nayak, Rejwanul Haque, and Andy Way. 2020. The ADAPT's submissions to the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 841–848, Online. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *CoRR*, abs/1910.13267.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Vocabulary adaptation for distant domain adaptation in neural machine translation. *CoRR*, abs/2004.14821.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

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. *CoRR*, abs/1706.03762.

Marta Villegas, Ander Intxaurrondo, A. Gonzalez-Agirre, M. Marimon, and Martin Krallinger. 2018. The mespen resource for english-spanish medical machine translation and terminologies : Census of parallel corpora , glossaries and term translations.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. *CoRR*, abs/1809.00068.

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'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

Huawei AARC's Submissions to the WMT21 Biomedical Translation Task: Domain Adaption from a Practical Perspective

Weixuan Wang[1] *, **Wei Peng**[1] *, **Xupeng Meng**[1], **Qun Liu**[2]
[1]Artificial Intelligence Application Research Center, Huawei Technologies
{weixuanwang2,peng.wei1,mengxupeng}@huawei.com
[2]Noah's Ark Lab, Huawei Technologies
{qun.liu}@huawei.com

Abstract

This paper describes Huawei Artificial Intelligence Application Research Center's neural machine translation systems and submissions to the WMT21 biomedical translation shared task. Four of the submissions achieve state-of-the-art BLEU scores based on the official-released automatic evaluation results (EN→FR, EN↔IT and ZH→EN). We perform experiments to unveil the practical insights of the involved domain adaptation techniques, including finetuning order, terminology dictionaries, and ensemble decoding. Issues associated with overfitting and under-translation are also discussed.

1 Introduction

General-purpose machine translation systems have limited capability in addressing domain-specific tasks (Koehn and Knowles, 2017), for example, the WMT biomedical translation shared task, due to the low availability for high-quality in-domain data. In our WMT20 submission, various domain adaption technologies (Bawden et al., 2019, 2020) have been applied including practical approaches finetuning on general-purpose models, back-translation (Sennrich et al., 2016) and leveraging in-domain dictionaries (Peng et al., 2020b). Despite achieving state-of-the-art (SOTA) BLEU scores for most of the submissions, few efforts were put in place to disclose the practical insights associated with these techniques.

This year, the Artificial Intelligence Application Research Center (AARC) participate in the WMT21 biomedical translation task for eight language directions between English and other four languages (French, German, Italian, and Chinese). The baseline model is an in-house general-purpose NMT model built upon the transformer-big architecture (Vaswani et al., 2017). Apart from presenting an overview of the proposed biomedical Neural Machine Translation (NMT) system, we investigate the practical insights of the involved domain adaptation techniques, including finetuning order, terminology dictionaries, and ensemble decoding. Issues associated with overfitting to in-domain data and under-translation are also discussed.

2 The Data

In this section we detail the bilingual and monolingual data used in this shared task (Table 1).

2.1 Bilingual Data

In-domain bilingual data In all directions, we use the in-domain data (IND) provided by the shared task organizers to finetune the base model. [1] The IND data consists of WMT-released bitexts from Pubmed, UFAL [2] and Medline. [3]

We notice that the official release of IND data suffers from issues of misalignment between source and target sentences, and missing target sentences. The translation of a source sentence may be misplaced in a different line or even appeared in multiple lines on the target side. Moreover, a source sentence may have not been translated into in a target sentence. A data processing pipeline is developed to address the issues mentioned above (depicted in 3.4). The test data is the official release of the WMT19 shared task.

Augmented Bilingual Data We collect in-domain data from TAUS [4] for the English-French, English-Italian and English-Chinese language pairs (depicted in Table 1 as IND-Aug.) to address the in-domain data scarcity issue. For English-Chinese data, after collecting a portion of abstracts of China

*Co-first authors.

[1]http://www.statmt.org/wmt21/biomedical-translation-task.html
[2]https://ufal.mff.cuni.cz/ufal_medical_corpus
[3]https://github.com/biomedical-translation-corpora/corpora
[4]https://md.taus.net/corona

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 868–873
November 10–11, 2021.

Directions	Train					Dev.	Test	Vocab.
	OOD	IND	IND-Dict.	IND-Aug.	IND-BT.			
EN→FR	3M	2.8M	62.5K	-	-	1.6K	1147	40K
FR→EN	3M	2.8M	62.5K	889K	53M	1.6K	952	40K
EN→DE	6M	2.4M	62.5K	-	5.5M	1.1K	963	42K
DE→EN	6M	2.4M	62.5K	-	53M	1.1K	794	42K
EN→IT	6M	139K	60.6k	235K	695k	0.8K	708	40K
IT→EN	6M	139K	60.6k	235K	55M	0.8K	760	40K
EN→ZH	3M	-	60.1K	847K	-	5K	774	50K
ZH→EN	3M	-	60.1K	847K	-	5K	418	50K

Table 1: Data used for training and evaluating the system. Note that "OOD" is short for the general domain data. "IND" is the in-domain data provided by the WMT organizers. "IND-Dict." refers to the in-domain dictionary. "IND-Aug." is the augmented IND data collected manually (not from MEDLINE, as depicted in 2.1). "IND-BT." is the IND monolingual data used for the back-translation. M is the acronym for "million," and K stands for "thousand".

Master's and Doctoral Dissertations, we align the data on the sentence level by using a model proposed by Açarçiçek et al. (2020). This is done by finetuning a RoBERTa (Liu et al., 2019) filter model on the TAUS dataset and selecting the source-target sentence pairs above a normalized log-probability threshold of 90%.

General-domain bilingual data We observe that finetuning the base model with IND data alone may incur sub-optimal BLEU scores. A conjecture is that the test data has a different distribution to that of the IND data. We present a case to show that finetuning the base model on a mixture of general domain data (OOD) and IND data can produce minor improvements (shown in 4.2).

2.2 Monolingual Data

A batch of monolingual Medline data in English (IND-BT.) dated before July 2018 has been collected and back-translated for data augmentation. The official released IND data from WMT is also back-translated. The models used for back-translation are from our last year's competition (Peng et al., 2020b).

3 The Approaches

The proposed systems are finetuned using the following methods. All models are trained on one Tesla V100 GPU, taking approximately 8-20 hours depending on the volumes of data involved.

3.1 Leveraging In-domain Dictionary

Leveraging domain-specific dictionaries is a viable solution for domain adaptation in NMT (Peng et al., 2020a,b) to enhance IND data coverage. We collect lexicons from SNOMED-CT [5], DOPPS[6], WFOT [7] and generate a terminology dictionary which is subsequently attached to the end of training data. Terminology is further entended to cover COVID-19 related terms obtained from Neulab.[8]

3.2 Ensemble

Ensembling methods is a machine learning technique that aggregates several base models to generate one optimal predictive model (Garmash and Monz, 2016). We choose the top two models to ensemble in an attempt to produce a more general NMT model.

3.3 Architecture

To train the in-domain NMT model, we choose the in-house NMT system trained on general domain data as a baseline built upon the transformer-big architecture. LazyAdam optimizer is used during the training phase with a learning rate of $1e^{-5}$ and a warm-up period of 16,000 steps. The dropout ratio is set to 0.1, and the batch size for training and validation is 6,144 and 32 tokens, respectively. The width of the beam search is 4. Early stopping is applied to the training.

[5]https://www.nlm.nih.gov/healthit/snomedct/index.html
[6]https://static.lexicool.com/dictionary/XJ9XO98314.pdf
[7]https://static.lexicool.com/dictionary/HY1TK12777.pdf
[8]https://github.com/neulab/covid19-datashare/tree/master/parallel/terminologies

System I	EN→FR	FR→EN	EN→DE	DE→EN	EN→IT	IT→EN	EN→ZH	ZH→EN
baseline	42.94	42.10	31.05	38.24	40.54	49.19	34.41	33.41
+ IND	45.03	44.81	31.90	33.81	36.35	42.28	-	-
+ IND, IND-Dict.	45.93	45.05	32.68	38.98	36.69	45.13	-	-
+ IND, IND-Dict., OOD	45.65	-	32.45	39.26	41.77	48.88	-	-
+ IND, IND-Dict., OOD, IND-BT	-	44.56	33.79	40.25	42.69	50.80	-	-
+ IND, IND-Dict., OOD, IND-Aug.	-	-	-	-	40.83	-	36.08	35.35
+ IND, IND-Dict., OOD, IND-Aug., IND-BT	-	45.15	-	-	41.39	50.91	-	-
WMT21 Submission (Huawei_AGI)	**45.31**	**48.71**	**31.98**	**41.32**	**44.25**	**45.70**	**44.40**	**39.43**
WMT21 Best Official	**45.31**	**49.28**	**32.59**	**45.01**	**44.25**	**45.70**	**46.50**	**39.43**

Table 2: BLEU scores on all related submissions. The baseline models are finetuned in various configurations, including mixed finetuning on general-domain data (aka "OOD"), IND bitexts ("IND"), "IND-Dict." and the augmented IND data ("IND-Aug.").

3.4 Data Processing

Several pre-processing techniques are introduced to ensure the quality of the data.

- First, we perform punctuation normalization to standardize their formats using Moses library (Koehn et al., 2007).
- Then we carry out a primary data cleaning process to remove nonstandard sentences, including those with special characters, weblinks, extra spaces, and other bad cases.
- According to the length of the sentence after segmentation and the proportion of rare words, we remove bitexts with more rare words in the sentences. We further clean the data by skipping those sentence pairs with more than 100 subwords or less than one subword. The bitexts with a source and target sentence length ratio of more than 2.5 are excluded. A language detection tool [9] is used to filter out bitexts with abnormal language patterns, i.e., sentences with undesirable *langid*.
- An alignment model trained by fast-align (Dyer et al., 2013) [10] is used to score the corpus to remove misaligned parallel sentences.

After decoding, post-processing is performed to detokenize subwords and remove undesirable spaces between special characters and numbers, i.e., converting "rs = 0.9148" into "rs=0.9148".

4 Experimental Results and Analysis

The base systems are trained with OOD data and finetuned using IND data enhanced with monolingual data to produce the submitted results. We extract the OK-aligned data from the last two years (WMT19 and WMT20) and produce the test data to evaluate the NMT models. The BLEU scores are calculated using the MTEVAL script from Moses (Koehn et al., 2007). Results are shown in Table 2. The final two rows demonstrate the results of our submissions this year and the best official records released by the organizers.

4.1 Finetuning Order Does Matter

We identify the order of training is crucial in the experiment. We perform the experiment under the following three training strategies:

1. Strategy 1 (S1): the baseline is finetuned on the back-translation (BT) pseudo parallel corpus, followed by another finetuning using IND data.
2. Strategy 2 (S2): the baseline is finetuned using the IND data, followed by another finetuning using the BT data.
3. Strategy 3 (S3): the baseline is finetuned using a mixture of BT and IND data.

Table 5 presents the results of this comparative study for French→English translation direction. It can be observed that finetuning order generates significantly different BLEU scores, with Strategy 1 achieving a BLEU score +8.89 higher than that from Strategy 2. We follow the training strategy 1 in WMT21 shared task to this end.

4.2 OOD Data Mixed Finetuning

We observe that finetuning the base model with IND data alone (particularly with a limited amount of IND data) may result in sub-optimal BLEU scores. This may indicate overfitting to the training data, which has a different distribution to the test data. We perform a series of experiments to

[9] https://github.com/aboSamoor/polyglot
[10] https://github.com/clab/fast_align

Data	EN→FR	FR→EN	EN→DE	DE→EN	EN→IT	IT→EN
baseline	42.94	42.1	31.05	38.24	40.54	49.19
+IND	45.03	44.81	31.9	38.81	36.35	42.28
+IND + IND-Dict.	**45.93 (+0.90)**	**45.05 (+1.24)**	**32.68 (+0.78)**	**38.98 (+0.17)**	**36.69 (+0.34)**	**45.13 (+2.85)**

Table 3: Effects of applying terminology dictionaries to train English⇔French, English⇔German, English⇔Italian models on WMT20.

models	EN→FR	FR→EN	EN→DE	DE→EN	EN→IT	IT→EN	EN→ZH	ZH→EN
baseline	42.94	42.10	31.05	38.24	40.54	49.19	34.41	33.41
model-1	45.93	45.23	**33.37**	**40.15**	42.52	50.91	**36.05**	**35.31**
model-2	45.57	45.15	33.10	39.97	42.39	50.80	34.94	35.13
ensemble	**46.15**	**46.21**	33.27	40.12	**42.59**	**51.28**	35.78	35.11

Table 4: Results on the ensemble of three models on WMT20

Data	FR→EN	
	WMT19	WMT20
baseline	37.98	42.1
BT	30.06	34.19
IND	38.26	44.81
BT-IND (S1)	**39.26**	**45.10**
IND-BT (S2)	33.10	36.21
BT+IND (S3)	39.09	42.17

Table 5: The comparative study of finetuning order in French→English translation direction.

Data	EN→IT	IT→EN
baseline	40.54	49.19
IND	36.35	42.28
OOD-1M + IND + IND-Dict.	**41.77**	48.88
OOD-3M + IND + IND-Dict.	41.63	**49.10**
OOD-6M + IND + IND-Dict.	38.32	-

Table 6: Mixed finetuning OOD data creates improvements to address overfitting to IND when training English⇔Italian translation models on WMT20.

disclose this issue. As shown in Tables 6 and 7, finetuning with a mixture of OOD and IND data generates minor improvements. Interestingly, the experiment results are sensitive to the amount of OOD data involved. Future work is planned to look into this issue in detail.

4.3 The Effect of Terminology Dictionaries

In this section, we perform an ablation study to show the effectiveness of terminology dictionaries. The IND dictionaries are appended to bitexts as a part of the corpus to train NMT models. Table 3 presents consistent improvements for all six models in the experiment.

Data	EN→FR	
	WMT19	WMT20
baseline	39.06	42.94
IND	43.56	45.03
OOD-3M + IND + IND-Dict.	**43.65**	**45.65**
OOD-9M + IND + IND-Dict.	39.70	43.50

Table 7: The effects of mixed finetuning OOD data in improving the potential overfitting issue with IND data when training English→French translation models.

4.4 Ensemble Decoding

Ensemble decoding is applied to improve the generality of the NMT model by averaging the logarithmic probabilities of a decoded token. It can be observed from Table 4 that ensemble decoding is marginally effective compared to well-learned NMT models. This finding is consistent with that obtained from Wang et al. (2020).

4.5 Under-translation with Overfitting

Under-translation occurs when the NMT model fails to decode a portion of the input sentence. One of Chinese→English models under-translates a particular sentence of the WMT21 test data. For example, as shown in Table 8, "无危险器官受累患者的预后显著优于有危险器官受累的患者" of the input has been left untranslated. After increasing the width of the beam search, under-translation can be avoided. In our opinion, under-translation may be caused by noisy IND data, in which the learned self-attentions are not differentiable during decoding. By ensembling the affected model with the baseline, we successfully rectify the problem.

sentence	example
input	The disease duration ranged from 2 weeks to 60 months (median, 4 months), and the affected segment was C All the patients were followed up 3 to 42 months (median, 12 months).
prediction	病程2周
input	The median age of the 30 patients was 56.5 (28-80) years old, among them, 25 patients were primary plasma cell leukemia, and 5 patients were secondary plasma cell leukemia.
prediction	30例患者的中位年龄为56.5（28
input	无危险器官受累患者的预后显著优于有危险器官受累的患者，患者10年OS率分别为100%和60.6%（P=0.0007）。
prediction	The 10-year os rate was 100% and 60.6% respectively (p=0.0007).

Table 8: Under-translated examples of English⇔Chinese. The portion of the sentence marked in red is under-translated.

5 Conclusion

This paper depicts Huawei's neural machine translation systems and submissions to the WMT21 biomedical shared task. We have achieved state-of-the-art BLEU scores for four of eight language pairs (EN→FR, EN↔IT and ZH→EN) based on the official-released results. We also explore practical issues for the involved domain adaptation techniques, including the effects of finetuning order, terminology dictionaries, and ensemble decoding on enhancing the performances of cross-domain NMT. We have discussed issues associated with overfitting and under-translation.

Acknowledgements

We express our gratitude to colleagues from HUAWEI AARC and Noah's Ark Lab for their continuous support. We also appreciate the WMT 21 organizers for hosting this shared task and the anonymous reviewers' insightful comments.

References

Haluk Açarçiçek, Talha Çolakoglu, Pinar Ece Aktan Hatipoglu, Chong Hsuan Huang, and Wei Peng. 2020. Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 940–946. Association for Computational Linguistics.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno-Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurélie Névéol, Mariana L. Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 29–53. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno-Yepes, Nancy Mah, David Martínez, Aurélie Névéol, Mariana L. Neves, Maite Oronoz, Olatz Perez-de-Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 biomedical translation shared task: Basque, italian and russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 660–687. 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.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1409–1418. ACL.

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, Prague, Czech Republic. 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, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Wei Peng, Chongxuan Huang, Tianhao Li, Yun Chen, and Qun Liu. 2020a. Dictionary-based data augmentation for cross-domain neural machine translation. *CoRR*, abs/2004.02577.

Wei Peng, Jianfeng Liu, Minghan Wang, Liangyou Li, Xupeng Meng, Hao Yang, and Qun Liu. 2020b. Huawei's submissions to the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 857–861. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer 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.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020. Transductive ensemble learning for neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6291–6298. AAAI Press.

Tencent AI Lab Machine Translation Systems for the WMT21 Biomedical Translation Task

Xing Wang **Zhaopeng Tu** **Shuming Shi**
Tencent AI Lab, Shenzhen, China
{brightxwang,zptu,shumingshi}@tencent.com

Abstract

This paper describes the Tencent AI Lab submission of the WMT2021 shared task on biomedical translation in eight language directions: English-German, English-French, English-Spanish and English-Russian. We utilized different Transformer architectures, pre-training and back-translation strategies to improve the translation quality. Concretely, we explore mBART (Liu et al., 2020) to demonstrate the effectiveness of the pre-training strategy. Our submissions (**T**encent AI Lab **M**achine **T**ranslation, TMT) in German/French/Spanish⇒English are ranked 1st respectively according to the official evaluation results in terms of BLEU scores.

1 Introduction

This paper describes the Tencent AI Lab submission of the WMT2021 shared task on biomedical translation. Last year, we participated in three translation tasks: News (Wu et al., 2020), Chat (Wang et al., 2020a), and Biomedical (Wang et al., 2020b). In biomedical translation, we adopt DEEP TRANSFORMER (Dou et al., 2018, 2019), HYBRID TRANSFORMER (Hao et al., 2019) and DATA REJUVENATION[1] (Jiao et al., 2020). This year, we participated in eight language directions: English-German (En-De), English-French (En-Fr), English-Spanish (En-Es) and English-Russian (En-Ru) in the biomedical translation.

In this paper, we also apply the pre-train and fine-tune paradigm for the biomedical translation task. The pre-train model is first trained on the the large-scale monolingual data in a self-supervised manner, then is fine-tuned on downstream bilingual data. Specifically, we adopt the encoder-decoder pre-trained model mBART (Liu et al., 2020) to implement the pre-training strategy.

[1] https://github.com/wxjiao/Data-Rejuvenation

The rest of this paper is organized as below. Section 2 presents our system: Transformer and pre-trained model mBART. Section 3 describes the training and validation data used in our system. Section 4 reports experimental results in the participated eight language directions. Finally, we conclude our work in Section 5.

2 System

Our systems are implemented with Transformer (Vaswani et al., 2017) and the pre-trained model mBART. The training details of theses models are described in Section 4.

2.1 Transformer

We adopt the BIG and LARGE Transformer models used in the previous year (Wang et al., 2020b) as the basic Transformer models. BIG and LARGE Transformer models contain 6-layer and 20-layer encoders with TRANSFORMER-BIG setting (Vaswani et al., 2017), respectively.

2.2 Pre-train Model

For the sequence-to-sequence pre-training, we adopt mBART25 (Liu et al., 2020) as the pre-train model for our experiments, which consists of 12 encoder and decoder layers with the default size of hidden state is 1024. The model is pre-trained with the denoising objective on the large-scale monolingual data and is fine-tuned on the downstream tasks. mBART has achieved significant improvements on many low resource language paris.

3 Data

In this section, we present the training and validation data used in our system.

Besides the in-domain data provided by organisers, we collect the out-of-domain bilingual data from WMT news translation shared task.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 874–878

	En-De	En-Fr	En-Es	En-Ru
Out-of-domain	37.8M	28.0M	30.3M	92.0M
In-domain	2.5M	3.5M	1.6M	43.0K
Validation set	9.8K	1.5K	1.5K	4.0K

Table 1: The detailed statistics of training and validation data used in our system.

- En-De: Europarl-v10[2], Common Crawl corpus[3], ParaCrawl[4], News Commentary-v15[5] and Wiki Titles-v2[6].
- En-Fr: Europarl-v7[7], Common Crawl corpus, News Commentary[8], English-French Giga Corpus[9].
- En-Es: Europarl-v7[10], Common Crawl corpus, News Commentary[11], ParaCrawl[12].
- En-Ru: Common Crawl corpus, News Commentary[13], ParaCrawl[14], Yandex Corpus[15], Wiki Titles-v2, Back-translated news[16].

For the validation data, we use the Khresmoi development data[17] (En-De, En-Fr, En-Es) as the validation sets. We also use the HimL test sets 2015 and 2017[18] to enlarge the En-De validation set. For En-Ru, we randomly sample 4000 examples from the training data as the validation set.

The statistics of the in-domain and out-of-domain training data and the validation data are listed in Table 1.

To enlarge the in-domain bilingual corpus, we follow Wang et al. (2020b) to adopt back-translation method to generate synthetic bilingual sentence pairs. For English-X pair, we train a English-X LARGE model on the combination of in-domain and out-of-domain data, and use the model to generate synthetic bilingual data. We also collect the En-Ru bilingual biomedical data (about 1.0 M sentence pairs) from Internet as the in-domain data.

In this work, all corpora are tokenized by sentence-piece (Kudo and Richardson, 2018) model[19] without any pre-processing procedures.

4 Experiments

For the corpus filtering, we follow Wang et al. (2020b) to filter duplicate sentence pairs (Khayrallah and Koehn, 2018), sentence pairs with wrong language (Khayrallah and Koehn, 2018) or length problem (Ott et al., 2018).

For the synthetic bilingual data generation, we adopt iterative knowledge distillation (Li et al., 2019) to improve the translation quality. Our iterative knowledge distillation is performed with 3 BIG Transformer teachers and 3 iterations. We also try to use the Right-to-Left (R2L) training (Wu et al., 2020) but fail in achieving significant improvements on the test sets.

We follow Wang et al. (2020b) to train the BIG and LARGE Transformer models. Specifically, we first use the combination of the out-of-domain data and the in-domain data to train the teacher model. Then we use the teacher model to generate the synthetic bilingual data. Finally, we train the student model on the combination of the synthetic and real bilingual data (Jiao et al., 2021). The learning rate is set to 0.0007. All models are trained for 600K steps on 8 Tesla V100 GPUs where each is allocated with a batch size of 8192 tokens.

[2] http://www.statmt.org/europarl/v10/
[3] www.statmt.org/wmt13/training-parallel-commoncrawl.tgz
[4] https://s3.amazonaws.com/web-language-models/paracrawl/release8/en-de.txt.gz
[5] http://data.statmt.org/news-commentary/v15/
[6] http://data.statmt.org/wikititles/v2/
[7] http://www.statmt.org/wmt13/training-parallel-europarl-v7.tgz
[8] http://www.statmt.org/wmt15/training-parallel-nc-v10.tgz
[9] http://www.statmt.org/wmt10/training-giga-fren.tar
[10] training-parallel-europarl-v7.tgz
[11] http://www.statmt.org/wmt13/training-parallel-nc-v8.tgz
[12] https://s3.amazonaws.com/web-language-models/paracrawl/release8/en-es.txt.gz
[13] http://data.statmt.org/news-commentary/v16
[14] http://paracrawl.eu/download.html
[15] https://translate.yandex.ru/corpus?lang=en
[16] http://data.statmt.org/wmt20/translation-task/back-translation/
[17] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2122
[18] https://www.himl.eu/test-sets
[19] https://github.com/google/sentencepiece

System	De		Fr		Es		Ru
	2019	2020	2019	2020	2019	2020	2020
Best Official 19 (Bawden et al., 2019)	38.84	–	38.24	–	48.33	–	–
Best Official 20 (Bawden et al., 2020)	–	41.65	–	44.45	–	50.75	43.31
Transformer-Big	38.66	39.15	37.32	41.92	50.63	48.22	30.89
Transformer-Large	39.41	39.64	38.12	42.77	52.58	49.26	31.92

Table 2: BLEU scores on the German/French/Spanish/Russian⇒English biomedical test sets. Only the correctly aligned sentences are used in the test set.

System	De		Fr		Es		Ru
	2019	2020	2019	2020	2019	2020	2020
Best Official 19 (Bawden et al., 2019)	35.39	–	42.41	–	48.96	–	–
Best Official 20 (Bawden et al., 2020)	–	36.89	–	43.51	–	46.72	39.36
mBART	29.96	28.47	40.13	44.04	44.79	42.92	32.23
Transformer-Big	30.43	29.56	40.33	43.58	44.23	42.87	31.96
Transformer-Large	31.60	30.89	41.04	44.01	44.68	43.05	31.79

Table 3: BLEU scores on the English⇒German/French/Spanish/Russian biomedical test sets. Only the correctly aligned sentences are used in the test set.

System	2019
Baseline	37.72
+ In-domain Data	38.14
+ Data Rejuvenation	38.47
+ Back-translation	38.66
+ Ensemble	39.14

Table 4: BLEU scores of the Transformer-Big model on the German⇒English WMT2019 biomedical test set. Only the correctly aligned sentences are used in the test set.

For the pre-train model, we adopt the publicly available mBART25[20] model and fine-tune the mBART25 on the in-domain data. In the fine-tuning phase, we minimize the label smoothed cross entropy with the smoothing factor of 0.2. We use the Adam (Kingma and Ba, 2015) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e{-}6$. The learning rate is scheduled to increase from 0 to the maximum value in the warm-up phase and decreases linearly to 0 in the remaining steps. The dropout rate is 0.3 for each residual connection and 0.1 for attention matrices.

We carry out ablation study on De→En translation task. The results are shown in Table 4. The in-domain data improves the baseline Transformer-Big model with 0.42 BLEU point. We then apply the Data Rejuvenation, Back-translation and model ensemble strategies and achieve the further improvement.

We adopt the In-domain Data, Data Rejuvenation, Back-translation as the default setting and apply the setting to Transformer-Big and Transformer-Large models on the eight language directions. We train 5 BIG and 5 LARGE Transformer models with different random seeds initialization. With the trained models, we employ the model ensemble strategy with the greedy based ensemble (Li et al., 2019; Wu et al., 2020) to get the final translation outputs. For model inference, the length penalty is set to 0.6 and the beam size is set to 4.

Translation results are reported in term of BLEU score in Table 2 and Table 3. From the tables, we find that 1) utilizing different Transformer architectures, pretraining and back-translation strategies achieve strong performance on the De→En, En↔Fr and Es→En translation tasks. 2) the lack of the large-scale in-domain data makes our En-Ru NMT system significantly lower than the state-of-the-art systems, demonstrating that the in-domain data plays a critical role in the development of NMT system.

[20] https://github.com/pytorch/fairseq/tree/master/examples/mbart

System	En-De		En-Fr		En-Es		En-Ru	
Direction	←	→	←	→	←	→	←	→
Best Official	38.16	27.76	48.05	44.65	52.99	48.52	36.23	30.78
TMT	38.16	23.32	48.05	43.90	52.99	41.57	29.98	25.43

Table 5: Official BLEU scores of our submissions for WMT21 biomedical task.

Post-process We find that several long sentences exist in the 2021 test sets, which pose a great challenge for our NMT system. Take the following two sentences for example:

Sentence 6 in doc73 in medline_fr2en_fr.txt: "Nous avons constaté que: (i) malgré le fardeau de plus en plus lourd des maladies non transmissibles, nombre de pays à faible et moyen revenu ne possédaient pas les fonds suffisants pour assurer des services de prévention; (ii) les professionnels de santé au sein des communautés manquaient fréquemment de ressources, de soutien et de formation; (iii) les frais non remboursables dépassaient 40% des dépenses de santé dans la moitié des pays étudiés, ce qui entraîne des inégalités; et enfin, (iv) les régimes d'assurance maladie étaient entravés par la fragmentation des systèmes publics et privés, le sous-financement, la corruption et la piètre mobilisation des travailleurs informels."

Sentence 3 in doc27 in medline_es2en_es.txt: "Este artículo tiene como objeto el análisis de los ensayos clínicos que permitieron dicha autorización, así como la revisión de nuevas terapias para el tratamiento del carcinoma urotelial localmente avanzado o metastásico. MÉTODO: Búsqueda bibliográfica realizada en Pub-Med y ClinicalTrials.gov mediante la combinación de las palabras clave, en español e inglés: "carcinoma urotelial", "cáncer de vejiga", "localmente avanzado","metastásico", "inmunoterapia", "CTLA-4", "PD1", "PDL-1", "atezolizumab", "nivolumab", "ipilimubab", "pembrolizumab","avelumab", "durvalumab", "tremelimumab", "terapia antiangiogénica", "terapia molecular dirigida" e "inhibidores VEGF"."

To address the problem, we manually split the long sentences into multiple sentences, and use the splitted ones as the system input to perform the translation.

We also find our system may generate wrong translations for the very short input sentences, e.g., "RéSUMé: " (Sentence 1 in doc92 in medline_fr2en_fr.txt), "(" (Sentence 4 in doc11 in medline_es2en_es.txt). To overcome the problem, we extract the target translation from the SMT phrase table and use it as the final translation output, as the NMT and SMT models are identical in modeling the bilingual knowledge (He et al., 2020).

5 Official Results

The official automatic evaluation results of our submissions for WMT 2021 biomedical translation task are shown in Table 5. Our final systems in German/French/Spanish⇒English are ranked 1st respectively, in terms of BLEU score.

6 Conclusion

In this paper, we present Tencent AI Lab machine translation systems for the WMT21 biomedical translation shared task. we participated in eight language directions: English-German (En-De), English-French (En-Fr), English-Spanish (En-Es) and English-Russian (En-Ru). Our systems German/French/Spanish⇒English are ranked 1st according to the official evaluation results in terms of BLEU scores.

It is worth mentioning that most advanced technologies reported in this paper are also adapted to our systems for news translation task (Wang et al., 2021), which achieve the 1st rank in Chinese⇒English task.

In the future, we plan to explore Non-Autoregressive machine Translation (NAT) models to improve the system performance (Zhou et al., 2020; Ding et al., 2020; Hao et al., 2021) and will integrate these advanced techniques in our Tencent TranSmart System (Huang et al., 2021)[21].

References

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, et al. 2019. Findings of the wmt 2019 biomedical translation shared task:

[21] https://transmart.qq.com/index

Evaluation for medline abstracts and biomedical terminologies. In *WMT*.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurelie Neveol, Mariana Neves, Maite Oronoz, et al. 2020. Findings of the wmt 2020 biomedical translation shared task: Basque, italian and russian as new additional languages. In *WMT*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2020. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Zi-Yi Dou, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2019. Exploiting deep representations for natural language processing. *Neurocomputing*.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Towards better modeling hierarchical structure for self-attention with ordered neurons. In *EMNLP-IJCNLP*, pages 1336–1341.

Yongchang Hao, Shilin He, Wenxiang Jiao, Zhaopeng Tu, Michael Lyu, and Xing Wang. 2021. Multi-task learning with shared encoder for non-autoregressive machine translation. In *NAACL*.

Shilin He, Xing Wang, Shuming Shi, Michael R Lyu, and Zhaopeng Tu. 2020. Assessing the bilingual knowledge learned by neural machine translation models. *arXiv*.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: a practical interactive machine translation system. *arXiv*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data Rejuvenation: Exploiting Inactive Training Examples for Neural Machine Translation. In *EMNLP*.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *ACL*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *WMT*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of EMNLP*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *WMT*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *TACL*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

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 *NIPS*.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021. Tencent translation system for the WMT21 news translation task. In *WMT*.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020a. Tencent AI lab machine translation systems for WMT20 chat translation task. In *WMT*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2020b. Tencent AI lab machine translation systems for the WMT20 biomedical translation task. In *WMT*.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020. Tencent neural machine translation systems for the WMT20 news translation task. In *WMT*.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2020. Improving autoregressive NMT with non-autoregressive model. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*.

HW-TSC's Submissions to the WMT21 Biomedical Translation Task

Hao Yang[1], **Zhanglin Wu**[1], **Zhengzhe Yu**[1], **Xiaoyu Chen**[1], **Daimeng Wei**[1], **Zongyao Li**[1], **Hengchao Shang**[1], **Jiaxin Guo**[1], **Minghan Wang**[1], **Lizhi Lei**[1], **Chuanfei Xu**[1], **Min Zhang**[1], **Ying Qin**[1],
[1]Huawei Translation Service Center, Beijing, China
{yanghao30,wuzhanglin2,yuzhengzhe,chenxiaoyu35,weidaimeng,
lizongyao,shanghengchao,guojiaxin1,wangminghan,
leilizhi,xuchuanfei,zhangmin186,qinying}@huawei.com

Abstract

This paper describes the submission of Huawei Translation Service Center (HW-TSC) to WMT21 biomedical translation task in two language pairs: Chinese↔English and German↔English (Our registered team name is HuaweiTSC). Technical details are introduced in this paper, including model framework, data pre-processing method and model enhancement strategies. In addition, using the wmt20 OK-aligned biomedical test set, we compare and analyze system performances under different strategies. On WMT21 biomedical translation task, Our systems in English→Chinese and English→German directions get the highest BLEU scores among all submissions according to the official evaluation results.

1 Introduction

We have witnessed great progress made by neural machine translations (Bahdanau et al., 2015; Vaswani et al., 2017) in recent years. However, domain adaptation remains to be a tough issue. As noted by Koehn and Knowles (Koehn and Knowles, 2017), translations by NMT systems in out-of-domain scenarios are relatively poor, and high-quality data in specific domains are difficult to obtain, which pose great challenges to certain translation tasks (e.g. biomedical translation). To address the domain adaptation issue, on one hand, we leverage data diversification (Nguyen et al., 2020), forward translation (Wu et al., 2019) and back translation (Sennrich et al., 2016a; Edunov et al., 2018) to generate synthetic in-domain corpora. On the other hand, fine-tuning (Sun et al., 2019) and ensemble (Freitag et al., 2017; Li et al., 2019) are used to further enhance system performances in the biomedical domain.

We introduce our data strategy in section 2, and model architectures as well as model enhancement techniques in section 3. Section 4 presents experimental results of both language pairs on the wmt20 OK-aligned biomedical test set. Section 5 is a conclusion of our work.

2 Dataset

2.1 Data Source

Our baseline model is trained with out-of-domain WMT21 news data. The sizes of bilingual and monolingual data for Chinese↔English and German↔English language pairs are shown in Table 1.

With regard to in-domain data, we use both the bilingual data and monolingual data provided by the WMT21 Biomedical Translation Shared task. For German↔English task, we select Biomedical Translation and UFAL Medical Corpus as in-domain training data. Besides, 21.43M in-house English monolingual data are used. For Chinese↔English task, the used in-house data includes: 1.35M parallel data, 21.43M English monolingual data, and 36.11M Chinese monolingual data. Table 2 shows the details of data in the biomedical domain for German↔English and Chinese↔English tasks.

2.2 Data Pre-processing

Our data pre-processing methods include:

- Filter out repeated sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Normalize punctuations using Moses (Koehn et al., 2007).
- Filter out sentences with repeated fragments.
- Filter out sentences with mismatched parentheses and quotation marks.
- Filter out sentences of which punctuation percentage exceeds 0.3.
- Filter out sentences with the character-to-word ratio greater than 12 or less than 1.5.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 879–884
November 10–11, 2021.

Language	corpus WMT21 News Shared Task's Corpus	Mono English	Mono German	Mono Chinese
German↔English	96.6M	150M	150M	-
Chinese↔English	16.5M	150M	-	150M

Table 1: Out-domain data size of WMT21 Biomedical Translation Task

Language	corpus Biomedical Translation && UFAL	In-house Corpus	Mono English	Mono German	Mono Chinese
German↔English	3.06M	-	21.43M	-	-
Chinese↔English	-	1.35M	21.43M	-	36.11M

Table 2: In-domain data size of WMT21 Biomedical Translation Task

- Filter out sentences with more than 120 words.
- Apply langid (Joulin et al., 2017, 2016) to filter sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter sentence pairs with poor alignment.

It should be noted that for the German↔English translation task, we employ joint SentencePiece model(SPM) (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation, with the size of the vocabulary set to 32k. As for the Chinese↔English translation task, Jieba tokenizer is used for Chinese word segmentation while Moses tokenizer for English word segmentation. Byte Pair Encoding (BPE) (Sennrich et al., 2016b) is adopted for Chinese and English sub-word segmentation. We train BPE models with 32,000 merge operations for both the source and target sides.

3 System overview

3.1 Model

Our system uses Transformer (Vaswani et al., 2017) model architecture, which adopts full self-attention mechanism to realize algorithm parallelism, accelerate model training speed, and improve translation quality. Two Transformer deep-large model architectures are used in our experiments:

- Deep 25-6 (Wang et al., 2018; Li et al., 2019): Based on the Transformer-base model architecture, the deep 25-6 model features 25-layer encoder, 6-layer decoder, 1024 dimensions of word vector, 4096-hidden-state, 16-head self-attention and layer normalization.
- Deep 35-6 (Wu et al., 2020; Sun et al., 2019): Based on the Transformer-base model architecture, the deep 35-6 model features 35-layer encoder, 6-layer decoder, 788 dimensions of word vector, 3072-hidden-staten, 16-head self-attention and layer normalization.

We use the open-source Fairseq (Ott et al., 2019) for training. The main parameters are as follows: Each model is trained using 8 GPUs. The size of each batch is set as 2048, parameter update frequency as 32, learning rate as 5e-4 (Vaswani et al., 2017) and label smoothing as 0.1 (Szegedy et al., 2016). The number of warmup steps is 4000, and the dropout is 0.1. We also use the Adam optimizer (Kingma and Ba, 2015) with $\beta 1 = 0.9$, $\beta 2 = 0.98$. In the inference phase, The beam-size is 8, The length penalties for Chinese→English and German→English are set to 0.5, and the length penalties for the other two directions are set to 1.5.

3.2 Data augmentation

Given the small size of in-domain bilingual data, how to generate more training data becomes a crucial issue for model performance enhancement in the biomedical field. We adopt three data augmentation methods:

- Data diversification (Nguyen et al., 2020): Data diversification is a simple but effective strategy to enhance the performance of NMT. It uses predictions from multiple forward and backward models and then combines the results with raw data to train the final NMT model. The method does not require additional monolingual data and is suitable for all types of NMT models. It is more efficient than knowledge distillation and dual learning, and exhibits strong correlation with model integration. In our Chinese↔English and German↔English systems, we use only the forward model and the backward model to

create synthetic data and add the data to the original parallel corpora.

- Forward translation (Wu et al., 2019): Forward translation usually refers to using source language monolinguals to generate synthetic data through beam search decoding, and then add synthetic data to the training data so as to increase the training data size. Although merely using forward translation may not work well, forward translation can be used in conjunction with a back translation strategy, which also works better than using back translation alone. We do not use forward translation for the German→English system task due to the lack of high-quality in-domain German monolinguals. We then give up forward translation for the English→German direction because forward translation and back translation cannot be used jointly for better effects. Ultimately, we only adopt forward translation for our Chinese↔English systems.

- Back translation (Edunov et al., 2018): Back translation translates target side monolingual data back to the source language so as to increase the training data size, which has been proved to be an effective method to improve neural machine translation performances. There are many methods for generating synthetic corpus through back translation. In a non-extremely low-resource scenario, sampling or noisy beam search decoding method is more effective than beam search or greedy search, and the synthetic data generated by sampling or noisy beam search decoding method may introduce more diversity to training data. In our experiment, sampling decoding is adopted. We use back translation for all directions expect English→German, due to the lack of high-quality in-domain German monolinguals.

3.3 Training strategy

We first use in-domain training data to conduct incremental training with baseline models trained by WMT21 news data for domain transfer. Then, we use three monolingual enhancement strategies, data diversity, forward translation and back translation, to create synthetic data and add them to the in-domain training data to further expand the scale of the training data, and then perform incremental training again. In addition, we fine-tune our models with test sets from previous years of the same task in hope of further improving in-domain performances. Specifically, we ensemble multiple models to forward translate the source side of test sets to increase the size of the training data, and then add noise (Meng et al., 2020) to the target side of the training data to achieve a better fine-tuning effect. Finally, multiple models are ensembled to achieve better performance.

Algorithm 1: Strategies for selecting ensemble models

Input : The list of all NMT models to be selected $M := [M^1, ..., M^n]$, n is the Number of M, and the test Set T;
Output : The optimal model combination $B := [M^i, ..., M^j]$;

```
Initialize the test set T's maximum BLEU score maxbleu := 0;
Initialize the optimal model combination B := [];
for num ∈ range(1, n) do
    Generate a list of model combination numlist, which is all possible combination of num models in M;
    for current model combination subnumlist ∈ numlist do
        Calculate the current BLEU score curbleu of the current combined model on the test set T.;
        if curbleu > maxbleu: then
            B := subnumlist
            maxbleu := curbleu
        end
    end
end
return B
```

3.4 Ensemble

For each translation task, we randomize two sets of training data and train four models using the two model architectures mentioned above. In the course of our experiments, we find that directly ensemble all models does not necessarily perform better on test set than a single model. To achieve a better ensemble effect, we design an algorithm, as shown in the algorithm 1. The core idea is to traverse all combinations of models and find the best one in the

System	English→Chinese	Chinese→English
baseline	40.0	28.3
+ biomedical corpus	44.5 (+4.5)	31.4 (+3.1)
+ data diversification	44.9 (+0.4)	32.3 (+0.9)
+ forward translation & back translation	45.8 (+0.9)	34.6 (+2.3)
+ fine-tuning	47.6 (+1.8)	35.9 (+1.3)
+ ensemble	**47.7** (+0.1)	**36.4** (+0.5)
WMT20 best official	46.9	35.3

Table 3: Chinese↔English BLEU scores on the WMT20 OK-aligned biomedical test set.

test set. The experiment results show that ensemble with the best combination found by the traverse strategy is much better than simply ensemble all models. In our experiment, the model combination that performs best on the wmt20 OK-aligned biomedical test set is used as the final submission.

4 Experimental result

We train baseline models using WMT21 news data, then incrementally train them using medical bilingual corpora and synthetic data generated by data augmentation techniques, fine-tune models with previous years' test sets, and finally ensemble multiple models to produce submitted results. We benchmark our submissions using the WMT20 OK-align test set. BLEU scores are calculated using the MTEVAL script from Moses (Koehn et al., 2007). The results are shown in Table 3 and Table 4. Our models outperform last year's official best results in three language directions. The tables mainly show the results of deep 35-6 models. Only in the last ensemble phase, multiple model architectures are used. we compare our results with the best official results from last year. We notice that our baseline models trained by WMT news data may also perform quite well in the biomedical field. For example, in German→English, Our baseline model is only 2.2 BLEU below last year's best result.

4.1 Chinese↔English

For Chinese↔English task, we first train the baseline model on WMT21 news data. Then, incremental training is conducted with in-domain bilingual and synthetic data. Finally, models are fine-tuned with the previous test sets, and multiple models are ensembled to produce the final result. The experimental results of Chinese↔English are shown in Table 3. Compared with the baseline model, the final systems achieve improvements of 8.1 BLEU and 7.7 BLEU on Chinese→English and English→Chinese directions, respectively. Incremental training alone leads to increases of 3.1 BLEU and 4.5 BLEU on Chinese→English and English→Chinese respectively. Besides, the combination of data diversity, forward translation, and back translation also lead to significant improvements (3.2 BLEU increase for Chinese→English and 1.3 BLEU for the opposite direction). Fine-tuning on previous test sets further improves the model quality by 1.3 BLEU for Chinese→English and 1.8 BLEU for English→Chinese. Notably, no further improvements is achieved by ensemble all models, while ensemble the model combinations found through the ergodic approach further improves translation quality by 0.5 BLEU and 0.1 BLEU on Chinese→English and English→Chinese, respectively. Ultimately, on Chinese↔English task, our results outperform last year's official best results.

4.2 German↔English

For German↔English task, the model training strategy used is similar to that for Chinese↔English task, except data augmentation techniques. As mentioned above, due to the lack of in-domain German monolingual data, we use data diversity and back translation strategies for German→English direction and only data diversity for English→German direction. The German↔English experiment results are shown in Table 4. Data augmentation results in significant performance improvements, with 1.1 BLEU and 1.7 BLEU on German→English and English→German respectively. Fine-tuning with previous years' test sets has also improved the quality of in-domain translations. On German→English, we fine-tune the model with wmt18 and wmt19 test sets and see an improvement of 1.1 BLEU. On English→German, fine-tuning leads to an increase of 0.4 BLEU.

System	English→German	German→English
baseline	33.8	39.5
+ biomedical corpus	34.9 (+1.1)	39.8 (+0.3)
+ data diversification	35.5 (+0.6)	40.4 (+0.6)
+ back translation	-	40.6 (+0.2)
+ fine-tuning	35.9 (+0.4)	41.7 (+1.1)
+ ensemble	36.5 (+0.6)	**42.4** (+0.7)
WMT20 best official	**36.9**	41.7

Table 4: German↔English BLEU scores on the WMT20 OK-aligned biomedical test set.

Ensemble the model combinations found through the ergodic approach contribute to 0.7 BLEU increase for German→English and 0.6 BLEU for English→German. Ultimately, due to the lack of effective in-domain German monolingual data, we only surpass last year's official best results on German→English direction.

5 Conclusion

This paper presents the submissions of HW-TSC to the WMT21 Biomedical Translation Task. We perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated by our experiment results. Combining with data augmentation strategies, incremental training with in-domain data on the basis of a baseline model from new domain can effectively improve in-domain translation quality. Our systems in English→Chinese and English→German directions get the highest BLEU scores among all submissions according to the official evaluation results.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

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.

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*, pages 489–500.

Markus Freitag, Y. Al-Onaizan, and B. Sankaran. 2017. Ensemble distillation for neural machine translation. *ArXiv*, abs/1702.01802.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jegou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš 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*, pages 427–431.

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.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

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 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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL (1)*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang

Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, et al. 2020. Wechat neural machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 239–247.

Xuan-Phi Nguyen, Shafiq Joty, Wu Kui, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *ACL (1)*.

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.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

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.

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.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The niutrans machine translation system for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 528–534.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020. The volctrans machine translation system for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 305–312.

RTM Super Learner Results at Quality Estimation Task

Ergun Biçici
orcid.org/0000-0002-2293-2031
bicici.github.io

Abstract

We obtain new results using referential translation machines (RTMs) with predictions mixed to obtain a better mixture of experts prediction. Our super learner results improve the results and provide a robust combination model.

1 Introduction

Quality estimation task in WMT21 (Specia et al., 2021) (QET21) address machine translation (MT) performance prediction (MTPP), where translation quality is predicted without using reference translations, at the sentence-level (Tasks 1, 2, and 3) and with classification of sentences into containing a critical error or not (Task 3). Task 1 predicts the sentence-level direct assessment (DA) in 11 language pairs categorized according to the MT resources available:

- high-resource, English–German (en-de), English–Chinese (en-zh), and Russian-English (en-ru),
- medium-resource, Romanian–English (ro-en) and Estonian–English (et-en),
- low-resource, Sinhalese–English (si-en) and Nepalese–English (ne-en), and
- no-resource, English–Czech (en-cs), English–Japanese (en-ja), Pashto–English (ps-en), and Khmer–English (km-en) for zero-shot prediction.

en-ru contains sentences from both Wikipedia and Reddit articles while others use only Wikipedia sentences with 7000 sentences for training, 1000 for development, 1000 for test QET in 2020, and 1000 for testing at QET21. The target to predict in Task 1 is z-standardised DA scores, which changes the range from $[0, 100]$ for DA scores to $[3.178, -7.542]$ in z-standardized DA scores.

	Task	Train	Test	RTM interpretants setting	Training	LM
Task 1 and Task 2	en-de	9000	1000	bilingual	0.3 M	3.5 M
	en-zh	9000	1000	bilingual	0.2 M	3.5 M
	et-en	9000	1000	bilingual	0.2 M	3.5 M
	ne-en	9000	1000	bilingual	0.2 M	3.5 M
	ro-en	9000	1000	bilingual	0.2 M	3.5 M
	ru-en	9000	1000	bilingual	0.2 M	3.5 M
	si-en	9000	1000	bilingual	0.2 M	3.5 M
	en-cs	63000	1000	bilingual	0.2 M	3.5 M
	en-ja	63000	1000	bilingual	0.2 M	3.5 M
	km-en	63000	1000	bilingual	0.2 M	3.5 M
	ps-en	63000	1000	bilingual	0.2 M	3.5 M
Task 3	en-cs	9000	1000	bilingual	0.2 M	3.5 M
	en-de	9000	1000	bilingual	0.2 M	3.5 M
	en-ja	9000	1000	bilingual	0.2 M	3.5 M
	en-zh	9000	1000	bilingual	0.2 M	3.5 M

Table 1: Number of instances in the tasks and the size of the interpretants used.

The target to predict in Task 2 is sentence HTER (human-targeted translation edit rate) scores (Snover et al., 2006). We participated in sentence-level subtasks. 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 referential translation machine (RTM) (Biçici and Way, 2015; Biçici, 2020) models (M for million). In zero-shot prediction, we use all of the training instances made available to the task in all 7 translation directions. We tokenize and truecase all of the corpora using Moses' (Koehn et al., 2007) processing tools.[1] Language models (LMs) are built using `kenlm` (Heafield et al., 2013).

2 RTM for MTPP

We use RTM models for building our prediction models. RTMs predict data translation between the instances in the training set and the test set using interpretants, text data selected close to the task instances in bilingual training settings or

[1] https://github.com/moses-smt/mosesdecoder/tree/master/scripts

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 885–889
November 10–11, 2021.

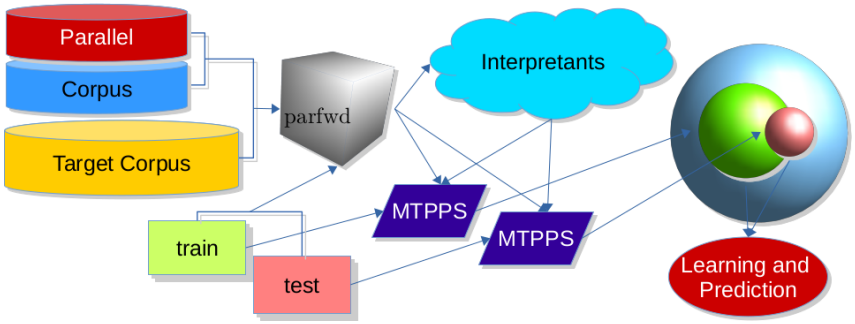


Figure 1: RTM: parfwd selects interpretants close to the training and test data using parallel corpus in bilingual settings and monolingual corpus in the target language or just the monolingual target corpus in monolingual settings; an MTPPS use interpretants and training data to generate training features and another use interpretants and test data to generate test features in the same feature space (largest sphere); learning and prediction use these features as input.

monolingual LM settings. Interpretants are text data that 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[2], the capability of the interpretant datasets selected to provide context for the training and test sets improve with parallel feature weight decay (`parfwd`) instance selection (Biçici, 2019). RTMs use `parfwd` for instance selection and for machine translation performance prediction system (MTPPS) (Biçici et al., 2013; Biçici and Way, 2015) to obtain the features, where additional features from word alignment are added. Figure 1 depicts RTMs and explains the model building process.

We treated all of Tasks 1, 2, and 3 as bilingual tasks where parallel corpora are obtained from WMT translation task.[3] The related monolingual or bilingual datasets are used during feature extraction. The machine learning models we use include ridge regression (RR), support vector regression (SVR) (Boser et al., 1992), gradient tree boosting, extremely randomized trees (Geurts et al., 2006), and multi-layer perceptron (Bishop, 2006) 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`.[4] We use RR to estimate the noise level for SVR, which obtains accuracy with 5% error compared with estimates obtained with known noise level (Cherkassky and Ma, 2004) and set $\epsilon = \sigma/2$. We use Pearson's correlation (r), mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), relative MAE (MAER), and mean RAE relative (MRAER) as evaluation metrics (Biçici and Way, 2015). Our best non-mixed results are in Table 2. Official evaluation metric is r_P.

3 Mixture of Experts Models

We use prediction averaging (Biçici, 2018) 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 with w:

$$\begin{aligned} w_{j,i} &= \frac{w_{j,i}}{1-w_{j,i}} \\ \hat{\boldsymbol{y}}_{\mu_k} &= \frac{1}{k}\sum_{i=1}^{k}\hat{\boldsymbol{y}}_i && \text{MEAN} \\ \hat{\boldsymbol{y}}_{j,w_k^j} &= \frac{1}{\sum_{i=1}^{k} w_{j,i}}\sum_{i=1}^{k} w_{j,i}\,\hat{\boldsymbol{y}}_i \\ \hat{\boldsymbol{y}}_k &= \frac{1}{|J|}\sum_{j\in J}\hat{\boldsymbol{y}}_{j,w_k^j} && \text{MIX} \end{aligned} \quad (1)$$

MEAN is the averaged results and MIX is the weighted average. We assume independent predictions and use $p_i/(1-p_i)$ for weights where p_i represents the accuracy of the independent classifier i in a weighted majority ensemble (Kuncheva and Rodríguez, 2014). We use the MIX prediction only when we obtain better results on the training set. We select the best model using r and mix the

[2] http://statmt.org/wmt21/

[3] http://statmt.org/wmt21/translation-task.html

[4] http://scikit-learn.org/

		r_P	MAE	RMSE
Task 1	en-de	0.212	0.4752	0.6809
	en-zh	0.223	3.8003	3.9333
	et-en	0.143	2.3699	2.5863
	ne-en	0.088	5.06	5.291
	ro-en	0.59	1.3623	1.5143
	ru-en	0.475	0.6301	0.8149
	si-en	0.21	0.8208	1.0258
	en-cs	0	7.6367	7.6871
	en-ja	0	7.5808	7.6215
	km-en	0.0209	7.4564	7.5266
	ps-en	-0.028	7.5792	7.638
Task 2	en-de	0.195	0.1605	0.2389
	en-zh	0.04	0.7707	0.8145
	et-en	0.148	0.1885	0.2271
	ne-en	0.075	0.1629	0.2058
	ro-en	0.716	0.1644	0.1927
	ru-en	0.356	0.1843	0.2383
	si-en	0.218	0.1946	0.2457
	en-cs	0.031	0.745	0.7876
	en-ja	0.031	0.3114	0.3872
	km-en	-0.094	0.3618	0.4379
	ps-en	0	0.5278	0.6322

Table 2: RTM test results in sentence-level MTPP in tasks 1 and 2 using the best non-mix result. r_P is Pearson's correlation.

results using r, RAE, MRAER, and MAER. We filter out those results with higher than 0.95 relative evaluation metric scores.

We also use generalized ensemble method (GEM) as an alternative to MIX to combine using weights and correlation of the errors, $C_{i,j}$, where GEM achieves smaller error than the best combined model (Perrone and Cooper, 1992):

$$\begin{aligned} \hat{\mathbf{y}}_{\text{GEM}} &= \sum_{i=1}^{L} w_i \psi_i(\mathbf{x}) = \mathbf{y} + \sum_{i=1}^{L} w_i \epsilon_i \\ C_{i,j} &= E[\epsilon_i, \epsilon_j] = (\psi_i(\mathbf{x}) - \mathbf{y})^T (\psi_i(\mathbf{x}) - \mathbf{y}) \\ w_i &= \frac{\sum_{j=1}^{L} C_{i,j}}{\sum_{k=1}^{L} \sum_{j=1}^{L} C_{k,j}} \end{aligned}$$

Super learner (Polley and van der Laan, 2010) is a stacking model on a library of L learning models that are V-fold cross-validated on the training set and constructs an $V \times L$ level 1 dataset. Theoretical results show that as the number of different predictors in the ensemble increase, the ensemble result gets closer to the oracle result (Dudoit and van der Laan, 2005). The function that minimize the empirical risk on the validation set will achieve lower error than the function that minimize the overall risk: $\frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\psi^*, y_i) - \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{\psi}, y_i) \geq 0$ (Vapnik, 1998).

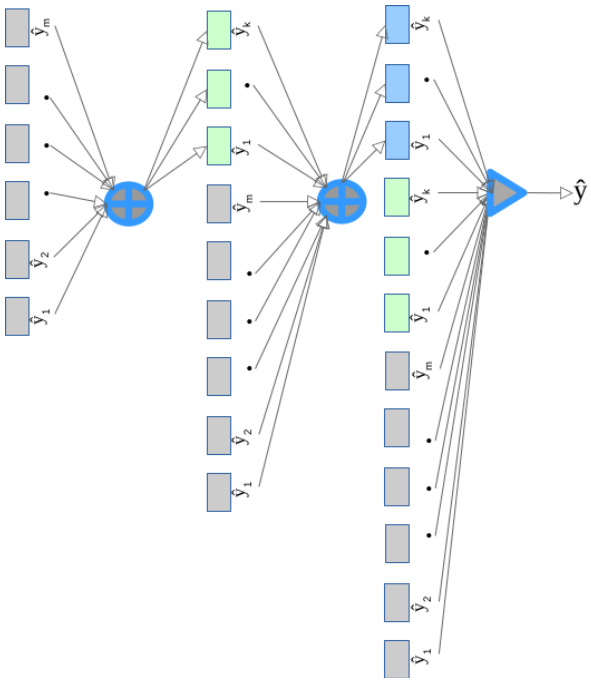


Figure 2: Model combination.

Model combination (Figure 2) selects top k combined predictions and adds them to the set of predictions where the next layer can use another model combination step or just pick the best model according to the results on the training set. We use a two layer combination where the second layer is a combination of all of the predictions obtained. The last layer is an $\arg\max$.

Our test set results using super learner are in Table 3. Before model combination, we further filter prediction results from different machine learning models based on the results on the training set to decrease the number of models combined and improve the results. A criteria that we use is MREAR ≥ 0.95 since MRAER computes the mean relative RAE score, which we want to be less than 1. In general, the combined model is better than the best model in the set. Super learner improve the results (Table 3).

The baseline deepQuest (Ive et al., 2018) use bidirectional gated recurrent unit type recurrent neural networks to model QET. RTM + deepQuest combination results in Task 2 use linear interpolation of RTM and deepQuest results with weights $0 \leq \lambda \leq 1$ and $1 - \lambda$ respectively as well as polynomial function fits to find the best combination model optimized on the development set. The most common function fit found is $f(x) =$

		r_P	MAE	RMSE
Task 1	en-de	**0.246**	0.5312	0.7699
	en-zh	**0.228**	4.4588	4.559
	et-en	0.13	2.9666	3.0942
	ne-en	0.087	**3.6449**	**3.8997**
	ro-en	0.376	3.1361	3.2656
	ru-en	0.347	0.9238	1.2276
	si-en	0.066	2.0869	2.3426
	en-cs	**0.053**	**7.0391**	**7.1159**
	en-ja	**-0.01**	**6.9076**	**6.9553**
	km-en	**0.032**	**5.6718**	**5.7694**
	ps-en	**-0.159**	**7.1563**	**7.27**
Task 2	en-de	0.125	0.1614	**0.237**
	en-zh	**-0.052**	**0.516**	**0.5648**
	et-en	**0.24**	0.2147	0.276
	ne-en	**0.299**	0.1797	0.2293
	ro-en	0.276	0.5562	0.603
	ru-en	0.143	0.2186	0.3197
	si-en	0.171	0.307	0.3713
	en-cs	**-0.108**	**0.7076**	**0.7535**
	en-ja	0.013	0.4636	0.5456
	km-en	0.008	0.5161	0.5928
	ps-en	**-0.064**	**0.4854**	**0.5671**

Table 3: RTM test results in sentence-level MTPP in tasks 1 and 2 using super learner. Improved results are shown in **bold**.

$a^x + bx^3 + cx^2 + dx + e$ (Table 4).

Task 3 results are in Table 5.

4 Conclusion

Referential translation machines pioneer a language independent approach and remove the need to access any task or domain specific information or resource and can achieve good results in automatic, accurate, and language independent prediction of translation scores. We present RTM ensemble results with super learner.

		r_P	MAE	RMSE
Task 2 deepQuest	en-de	0.225	0.1635	0.211
	en-zh	0.206	0.3033	0.3442
	et-en	0.39	0.1763	0.2256
	ne-en	0.366	0.1958	0.2441
	ro-en	0.558	0.1707	0.219
	ru-en	0.273	0.2062	0.2822
	si-en	0.338	0.2046	0.2542
Task 2 RTM + deepQuest	en-de	0.205	**0.1558**	0.2289
	en-zh	0.129	0.3786	0.4244
	et-en	**0.425**	**0.1684**	**0.2147**
	ne-en	0.353	**0.1653**	**0.2112**
	ro-en	0.397	0.5252	0.5692
	ru-en	-0.121	0.3633	0.4558
	si-en	0.277	0.2512	0.3111

Table 4: RTM test results in sentence-level MTPP in Task 2 using deepQuest and results combining deepQuest with super learner results.

		MCC	F1 BAD	F1 GOOD	F1 MULTI
Task3	en-cs	0.0508	0.81	0.24	0.1944
	en-de	0.0778	0.7874	0.2634	0.2074
	en-ja	-0.0523	0.1639	0.1418	0.0232
	en-zh	-0.0052	0.6059	0.2401	0.1455

Table 5: RTM test results in sentence-level MTPP in Task 3 using super learner.

References

Ergun Biçici. 2018. RTM results for predicting translation performance. In *Proc. of the Third Conf. on Machine Translation (WMT18)*, pages 765–769, Brussels, Belgium.

Ergun Biçici. 2019. Machine translation with parfda, Moses, kenlm, nplm, and PRO. In *Proc. of the Fourth Conf. on Machine Translation (WMT19)*, pages 122–128, Florence, Italy.

Ergun Biçici. 2020. RTM ensemble learning results at quality estimation task. In *Proc. of the Fifth Conf. on Machine Translation (WMT20)*, Online.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*, 27(3-4):171–192.

Ergun Biçici and Andy Way. 2015. Referential translation machines for predicting semantic similarity. *Language Resources and Evaluation*, pages 1–27.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proc. of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

Vladimir Cherkassky and Yunqian Ma. 2004. Practical selection of svm parameters and noise estimation for svm regression. *Neural Networks*, 17(1):113–126.

Sandrine Dudoit and Mark J. van der Laan. 2005. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2(2):131–154.

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 *51st Annual Meeting of the Assoc. for Comp. Ling.*, pages 690–696, Sofia, Bulgaria.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepQuest: A framework for neural-based quality estimation. In *Proc. of the 27th Intl. Conf. on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA.

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 *45th Annual Meeting of the Assoc. for Comp. Ling.*, pages 177–180, Prague, Czech Republic.

Ludmila I. Kuncheva and Juan J. Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275.

Michael Perrone and Leon Cooper. 1992. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, Brown Univ. Providence RI Inst. for Brain and Neural Systems.

Eric C. Polley and Mark J. van der Laan. 2010. Super learner in prediction. Technical report, U.C. Berkeley Division of Biostatistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Assoc. for Machine Translation in the Americas*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proc. of the Sixth Conf. on Machine Translation*, Online. Association for Comp. Ling.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

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.

HW-TSC's Participation at WMT 2021 Quality Estimation Shared Task

Yimeng Chen[1*], **Chang Su**[1*], **Yingtao Zhang**[1*], **Yuxia Wang** [1], **Xiang Geng** [2],
Hao Yang [1], Shimin Tao [1], Jiaxin Guo [1], Minghan Wang [1], Min Zhang [1], Yujia Liu [1], Shujian Huang [2]
[1] Huawei Translation Services Center, Beijing, China
[2] Nanjing University, Nanjing, China
{chenyimeng,suchang8,zhangyintao9,wangyuxia5,yanghao30,taoshimin,
guojiaxin1,wangminghan,zhangmin186,liuyujia13}@huawei.com
{gx@smail, huangsj@}nju.edu.cn

Abstract

This paper presents our work in WMT 2021 Quality Estimation (QE) Shared Task. We participated in all of the three sub-tasks, including Sentence-Level Direct Assessment (DA) task, Word and Sentence-Level Post-editing Effort task and Critical Error Detection task, in all language pairs. Our systems employ the framework of Predictor-Estimator, concretely with a pre-trained XLM-Roberta as Predictor and task-specific classifier or regressor as Estimator. For all tasks, we improve our systems by incorporating post-edit sentence or additional high-quality translation sentence in the way of multitask learning or encoding it with predictors directly. Moreover, in zero-shot setting, our data augmentation strategy based on Monte-Carlo Dropout brings up significant improvement on DA sub-task. Notably, our submissions achieve remarkable results over all tasks.

1 Introduction

Quality Estimation (QE) focuses on estimating the quality of machine translation (MT) system output when no ground truth reference is available (Specia et al., 2018). QE covers wide range of tasks including word-level, sentence-level and document-level. It has wide range of applications in MT quality check and post-editing effort estimation.

In WMT2021 Quality Estimation shared task[1], there are three sub tasks — Sentence-Level Direct Assessment task, Word and Sentence-Level Post-editing Effort task and Critical Error Detection task. Each sub task involves several language pairs. Our team participated in all the above three tasks over all language pairs. We summarized our main contributions as follow:

- We employ Predictor-Estimator architecture (Kim et al., 2017b; Kim and Lee, 2016) which is a two-stage model consisting of a word prediction model trained from large-scale parallel corpora, and a estimation model trained from quality-annotated QE data. Different from the original Predictor-Estimator model in (Kim et al., 2017a), we use pre-trained XLM-Roberta large as predictor instead of RNN-based model to achieve better QE features, and use task-specific classifier or regressor as quality estimator.

- We extend PE assisted QE (PEAQE) (Kepler et al., 2019; Wang et al., 2020) by integrating real PE or addtional high-quality translation in the way of multitask learning or directly encoding it with predictor.

- We explore data augmentation method based on Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) to enhance the performance of zero-shot language pairs in Direct Assessment(DA) task.

Our methods achieve impressive performance on both word and sentence level tasks. Specifically, we peak the top-1 on sentence-level DA over English-German and English-Japanese pairs. For word and sentence-level post-editing effort task, our submissions of the majority language pairs obtain the best Pearson's correlation or Matthews correlation coefficient. We also win the first place in critical error detection task in English-Chinese and English-Japanese.

We will describe the tasks, datasets, and our methods for DA task, post-editing task, and critical error detection task in section 2, section 3, and section 4 respectively. Section 5 presents details of our experimental setup and results, with a brief discussion and conclusion in the end.

* Indicates equal contribution.
[1]http://www.statmt.org/wmt21/quality-estimation-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 890–896
November 10–11, 2021.

2 Sentence-Level Direct Assessment Task

2.1 Task Description

The sentence-level Direct Assessment task focuses on estimating sentence-level translation quality scores which are annotated with Direct Assessment (DA) scores by professional translators. The original DA scores are in scale of 0-100. The scores are then standardised using the z-score by rater. The goal is to estimate a z-standardised DA score for each translation sentence.

Sentence-level DA task is evaluated by Pearson's correlation between the predicted score and the gold human annotated z-standardised DA score. The system is assessed from two aspects: single language pair and multilingual track which takes all languages into account, including zero-shot pairs, calculating the averaged Pearson correlation overall.

2.2 Dataset

For each language, 7000, 1000 and 1000 sentence pairs are provided officially as training, development and test20 set before releasing another 1000 for the real blind test21, including high-resource English-German (En-De) and English-Chinese (En-Zh), medium-resource Romanian-English (Ro-En) and Estonian-english (Et-En), low-resource Sinhalese-English (Si-En) and Nepalese-English (Ne-En), as well as Russian-English (Ru-En). Besides, 4 language pairs — English-Czech (En-Cz), English-Japanese (En-Ja), Pashto-English (Ps-En) and Khmer-English (Km-En), are only offered blind test (1000), without training data.

2.3 Implemented Systems

The systems for DA employ Predictor-Estimator architecture. Following previous sota works (Fomicheva et al., 2020; Moura et al., 2020; Rei et al., 2020), we use a pre-trained XLM-Roberta (XLM-R)(Conneau et al., 2019) model as a predictor due to its impressive performance on cross-lingual downstream tasks.

Practically, we concatenate source(SRC) and target(MT) sentences in the format `[CLS] SRC [EOS] [SEP] MT [EOS]` following XLM-R usage, and take the embedding of pooled output of [CLS] token as features of a sentence pair. For Estimator, we simply stack two-layer FFN, taking the [CLS] feature generated above as the input to predict sentence-level DA scores.

2.3.1 PE Assisted Sentence-Level DA Prediction

Inspired by the Pseudo-PE techniques (Kepler et al., 2019; Wang et al., 2020), we take full use of post-editing sentences provided in Post-editing Effort task through multitask learning. The model jointly learns to score (SRC, MT) pair in a regression task, and distinguish between translations and post-edited sentences — which is the better translation in a classification task. In inference stage, the model only conducts regression task to predict DA score, as post-editing sentences are not available for blind test set.

The regression task applies loss function as:

$$\mathcal{L}_{reg} = (\phi(E_{s,t}) - Y_{human})^2 \quad (1)$$

where $E_{s,t}$ is the embedding of sentence pair (source, mt), ϕ is the regressor taking them as input, through a two-layer FFN to compute DA score, and Y_{human} is the Z-normalized DA score annotated by human.

The classification task forces the model to capture more expressive cross-lingual sentence representation which is paramount for DA score. In implementation, we get the model to learn which is the pair with better translation between embedding of concatenated source and target $E_{s,t}$ and embedding of concatenated source and PE $E_{s,p}$. We splice two vectors in random order and apply two stacked FFN layers to compute classification result, in which 0 means the former pair is the better (i.e. the former contains PE), 1 means the former is the worse and 2 means translation and post-edit are exactly the same. Equation (2) gives the loss function for the classification task, where M is the number of classes ($M = 3$), Y is the binary indicator (0, 1, 2) if class label c is the correct classification for observations, P is the model predicted probabilities that the observation is of classes.

$$\mathcal{L}_{cls} = -\sum_{c=0}^{M-1} Y_c \log(P_c) \quad (2)$$

2.3.2 Data Augmentation for Zero-shot Languages

Instead of directly applying the multilingual DA model trained on other 7 language pairs to zero-shot languages, we exploit a data augmentation strategy based on MC dropout to improve the performance. Specifically, we compute the expectation and variance for the set of estimated DA scores

of zero-shot languages obtained by performing N (N=30) stochastic forward passes through the well-trained but dropout-perturbed QE model. In order to control the uncertainty introduced by the disturbance, we only retain dropout in estimator and last two layers in XLM-R. We take variance as an indicator to detect observations with less uncertainty and use expectation as DA score label. Then, we mix the generated zero-shot DA data with randomly selected non-zero-shot training set to fine-tune the model. Experiments show that our data augmentation is effective to improve the performance, achieving better Pearson correlation.

3 Word and Sentence-Level Post-editing Effort Task

3.1 Task description

Word-Level QE estimates the translation quality by producing a sequence of tags for source and target. For target sentences(MT), each token is tagged as either OK or BAD, each gap between two words is tagged as BAD if one or more missing words should have been there, and OK otherwise. So the number of total tags for each target sentence is $2N + 1$, where N is the number of tokens in the target sentence. For source sentences(SRC), tokens are tagged as OK if they were correctly translated, and BAD otherwise. The number of total tags for each source sentence is M, where M is the number of tokens in the source sentence. The evaluation metrics of the word-level task is the Matthews Correlation Coefficient (MCC).

Sentence-Level QE predicts the Human Translation Error Rate (HTER). HTER is the ratio between the number of edits (insertions / deletions / replacements) needed and the reference translation length. The evaluation metrics of the sentence-level task is Pearson's correlation metric.

3.2 Dataset

The dataset in these task provides the same source and translation as DA task, with an extra post-edit sentence for each observation and task-specific token-level and sentence-level labels. Besides, we generate addition-translation sentence (AMT) for each source sentence by using well-trained machine translation systems. The motivation here is to add an additional criterion which is in the same language as the provided translation sentence. We suppose that to detect the difference between two sentence in the same language is a simpler task for model. There are some important label properties to highlight:

- The number of BAD tags and OK tags is imbalanced, especially for GAP tags.
- AMT's BLEU score is significantly lower than MT taking post-edits as reference. Its average HTER is higher than MT. It indicates that the generated AMT is less closer to post-edits than MT.

3.3 Method

The systems for QE shared task2 also employ Predictor-Estimator architecture(Kim et al., 2017b).

Predictor. Similar to Task1, we use pre-trained XLM-Roberta (XLM-R) model as predictor after fine-tuning it with mask language modeling task(Devlin et al., 2018) using the provided source and PE sentences. In order to improve the performance, refers to approach in (Wang et al., 2020), we concatenate SRC, MT, AMT sentences together in the format of `[BOS] SRC [EOS] [SEP] MT [EOS] [SEP] AMT [EOS]`.

We notate the predictor as f; SRC, MT and AMT text as X and Y and Z, corresponding features as H_x, H_y, H_z respectively:

$$H_x, H_y, H_z = f(X, Y, Z), \tag{3}$$

Estimator. We utilise 4 independent 2-layers FFN including binary three classification tasks to predict SRC word tags, MT/AMT word tags, MT/AMT gap tags respectively, and a regression task to predict HTER score of MT/AMT. All predictions are obtained by performing specific transformations ϕ. We define the predicted logits of SRC word, MT word, MT gap, AMT word, AMT gap as $\hat{V_{xw}}$, $\hat{V_{yw}}$, $\hat{V_{yg}}$, $\hat{V_{zw}}$, $\hat{V_{zg}}$; and HTER predicted score of MT and AMT as $\hat{V_{yh}}$, $\hat{V_{zh}}$. The estimator can be described as:

$$\begin{aligned}
\hat{V_{xw}} &= \phi_{xw}(H_x), \\
\hat{V_{yw}} &= \phi_{w}(H_y), \\
\hat{V_{zw}} &= \phi_{w}(H_z), \\
\hat{V_{yg}} &= \phi_{g}(f_{cat}(H_y, \hat{V_{yw}})), \\
\hat{V_{zg}} &= \phi_{g}(f_{cat}(H_z, \hat{V_{zw}})), \\
\hat{V_{yh}} &= \phi_{h}(f_{gap}(f_{cat}(H_y, \hat{V_{yw}}, \hat{V_{yg}}))), \\
\hat{V_{zh}} &= \phi_{h}(f_{gap}(f_{cat}(H_z, \hat{V_{zw}}, \hat{V_{zg}}))),
\end{aligned} \tag{4}$$

where f_{cat} is the concatenate method in the last dimension, f_{gap} is the global average pooling in

the second dimension ignoring padding tokens in a batch just like (Lin et al., 2013) 3.2.

Loss. We prepend and append two special <*pad*> labels to the original word label sequence, append a special <*pad*> label to the original gap label sequence during training, but loss of the padded labels is not computed. For all classification tasks, to deal with the problem of imbalance between OK and BAD number, we use weighted cross entropy as the loss function, and the weight is calculated as $w_i = \frac{N}{\sum C_i}$, where w_i is the inverse of the proportion of the instance with class C_i. For sentence-level HTER score loss, we use mean squared error (MSE) as the loss function. We define the tags of SRC word, MT word, MT gap, AMT word, AMT gap as V_{xw}, V_{yw}, V_{yg}, V_{zw}, V_{zg}; and HTER score of MT and AMT as V_{yh}, V_{zh}.

The model is trained under the multi-task learning framework by summing up the loss of all subtasks with specific weights:

$$loss = \sum_{\tau \in \{xw,yw,yg,zw,zg\}} \lambda_\tau log P(V_\tau | X, Y, Z) + \sum_{\tau \in \{hy,hz\}} \lambda_\tau \sqrt{\sum (V_\tau - \hat{V}_\tau)^2}, \quad (5)$$

where xw, yw, yg, zw, zg represents for classification tasks, hy, hz represents for regression tasks, λ is the weight of loss for a specific task. The multi-task framework can improve the overall performance.

4 Critical Error Detection

4.1 Task Description

This is a new QE task focusing on predicting sentence-level binary scores indicating whether or not a translation contains (at least one) critical error. The key point is to identify whether the translation will lead to misleading or more serious consequences, e.g. the translation involves critical mistranslation, hallucination or critical content deletion. Only binary prediction (whether or not any critical error contained) is required. The evaluation metrics of this task is also the MCC.

4.2 Dataset

The dataset contains 4 languages which are English-German, English-Chinese, English-Czech, English-Japenese. 7000 training, 1000 validation, and 1000 blind test sentence pairs are available for each language. Ground truth label has two classes, NOT means no catastrophic error, and ERR means at least one catastrophic error in the translation. It is noticed that the number of NOT and ERR tag is imbalanced.

Language	Baseline	+Multitask	+Ensemble
En-De	0.490	0.552	0.547
En-Zh	0.494	0.502	0.519
Ro-En	0.886	0.897	0.902
Et-En	0.798	0.805	0.814
Ne-En	0.776	0.789	0.801
Si-En	0.648	0.677	0.675
Ru-En	0.761	0.787	0.787
Average	**0.693**	**0.716**	**0.721**

Table 1: Pearson correlation between prediction of our system and human DA judgement of non-zero-shot language pairs on test20 set.

Language	Baseline	+AugData	+All
En-De	0.481	/	0.584
En-Zh	0.523	/	0.583
Ro-En	0.878	/	0.901
Et-En	0.775	/	0.808
Ne-En	0.810	/	0.858
Si-En	0.564	/	0.581
Ru-En	0.753	/	0.787
En-Cz	0.546	0.557	0.573
En-Ja	0.297	0.349	0.364
Ps-En	0.592	0.622	0.622
Km-En	0.661	0.653	0.659
Multilingual	**0.621**	/	**0.665**

Table 2: Pearson correlation between prediction of our system and human DA judgement on test21 set.

4.3 Methods

Similar as the above two tasks, our baseline system takes pre-trained XLM-R as predictor, stacked FFN layers as binary classifier. We also experimented with replacing XLM-R by mBART (Liu et al., 2020) and replacing FFN layers with TextCNN, Bi-LSTM and other types of network.

Based on the intuition that the semantic difference between two monolingual sentences are easier to distinguish than that of two cross-lingual sentences, we propose to incorporate a "good" MT of the source sentence into *(src. mt)* pair during training, so that the auxiliary information provided by the "good" MT can help the model to directly compare *mt* with MT+*src*, instead of only depending on cross-lingual *src*. With consideration of expensive overhead of manual translation, we assume that au-

Score	Method	En-Zh	En-DE	Ru-En	Ro-En	Et-En	Si-En	Ne-En
Pearsonr	baseline(dev)	0.3013	0.4910	0.4475	0.5381	0.5997	0.6062	0.5899
	+AMT(dev)	0.3481	0.6003	0.5387	0.8479	0.7832	0.8031	0.6902
	+Ensemble(dev)	0.3772	0.6678	0.5704	0.8914	0.8249	0.8573	0.7849
	All(test)	0.3681	0.6531	0.5615	0.8623	0.8094	0.8690	0.7976
SRCW	baseline(dev)	0.1991	0.3019	0.2904	0.4132	0.4173	0.3899	0.4027
	+AMT(dev)	0.2895	0.4378	0.3991	0.6027	0.5204	0.5780	0.5109
	+Ensemble(dev)	0.3128	0.4502	0.4277	0.6374	0.5396	0.6033	0.5576
	All(test)	0.3098	0.4499	0.4258	0.6140	0.5490	0.6159	0.5450
MTW	baseline(dev)	0.1354	0.3988	0.3500	0.4980	0.4533	0.5393	0.4418
	+AMT(dev)	0.3346	0.4907	0.4331	0.6642	0.6006	0.7446	0.6721
	+Ensemble(dev)	0.3726	0.5149	0.4479	0.6807	0.6177	0.8102	0.7007
	All(test)	0.3536	0.5095	0.4507	0.6664	0.6058	0.8469	0.6741
MTG	baseline(dev)	0.0998	0.1987	0.2249	0.2856	0.2017	0.2844	0.3129
	+AMT(dev)	0.1799	0.3101	0.3481	0.4379	0.3119	0.5023	0.4001
	+Ensemble(dev)	0.1822	0.3158	0.3725	0.4531	0.3280	0.5573	0.4490
	All(test)	0.1719	0.2997	0.3877	0.4457	0.3115	0.6392	0.4027

Score	Method	En-Cs	En-Jp	Ps-En	Km-En	Multilingual
Pearsonr	baseline(test)	0.2910	0.0999	0.3722	0.3571	0.5002
	+Ensemble(test)	0.4750	0.2620	0.5343	0.4750	0.6314
SRCW	baseline(test)	0.1981	0.1523	0.2344	0.3183	—-
	+Ensemble(test)	0.3128	0.2166	0.3044	0.4101	—-
MTW	baseline(test)	0.2107	0.1372	0.2789	0.3077	—-
	+Ensemble(test)	0.3801	0.2581	0.4497	0.6364	—-
MTG	baseline(test)	0.1149	0.0901	0.1342	0.2691	—-
	+Ensemble(test)	0.2126	0.1523	0.2602	0.4190	—-

Table 3: Pearsonr correlation, MCC of words in SRC, MCC of words in MT and MCC of gaps in MT between prediction of our system and labels. SRCW is SRC words MCC, MTW is MT words MCC, MTG is MT gaps MCC, Test20 set is used as training set. Results of test set are from official leaderboard.

tomatic machine translation (AMT) of top commercial machine translation tools can also be competent at this work. Practically, we apply Baidu Fanyi [2] and Google Translate [3] API, obtaining two corresponding AMTs given a source sentence. Then we concatenate it with source and original machine translation in the format of `[CLS] SRC [EOS] [SEP] MT [EOS] [SEP] AMT [EOS]`, followed by encoding the concatenated triplet to the predictor.

Voting-Based Ensemble. Finally, we ensemble several models and take their majority voting as prediction results.

5 Experimental Results

5.1 Task1: Sentence-level Direct Assessment

Experimental Settings Our system is implemented with hugging face transformers package. The pre-trained xlm-roberta-large model which has approximately 550M parameters is taken as predictor. We train the predictor and the estimator together on the multilingual QE DA dataset using Adam(Kingma and Ba, 2015) as optimizer with constant learning rate of $1e^{-6}$ and training batch size of 16. The model is trained on a Nvidia Tesla V100 GPU.

Results Table 1 shows the results on test20 set. Our baseline is the system described in section 2.3. +multitask method is introduced in section 2.3.1. To achieve more competitive scores while also maintain a relatively small number of parameters, we ensemble our result with MC dropout approach, that is to run N (N=50) pass forwards with dropout and take the expectation of the N predictions as final answers. Table 2 presents the experimental results on blind test21 set. The baseline here is the same as Table 1 baseline. +AugData is the approach mentioned in section 2.3.2. +All is our final submitted result that integrates multi-task, data augmentation and ensemble.

[2] https://fanyi.baidu.com/
[3] https://translate.google.com/

Dataset	Pre-trained Model	Classification Layer	AMT	En-Zh	En-De	En-Cs	En-Ja
Dev	baseline	FFN	/	0.1873	0.4008	0.3974	0.2193
	MBart			0.2317	0.3940	0.4112	0.2148
	XLMR-Large			**0.2989**	**0.4846**	**0.4537**	**0.2744**
	XLMR-Large	TextCNN	/	0.1820	0.2008	0.2139	0.1429
		Bi-LSTM		**0.2350**	**0.4279**	**0.4132**	0.1981
		RCNN		0.2045	0.3850	0.3463	**0.2523**
	XLMR-Large	FFN	BaiduTrans	**0.3474**	0.4623	0.4372	**0.2948**
			GoogleTrans	0.2515	**0.4732**	**0.4551**	0.2724
	Ensemble			**0.3962**	**0.5104**	**0.4854**	**0.3542**
Test	Ensemble			**0.3533**	**0.4899**	**0.4482**	**0.3184**

Table 4: MCC of all language pairs over development(dev) set and test set.

5.2 Task2: Word and Sentence-Level Post-editing Effort Task

Settings: The batch size in training stage is 8. We use Adam as optimizer with learning rate of $2e^{-5}$. Each estimator FFN layer has a 0.1 dropout. Loss weight are: ($\lambda_{yh} = 2$, $\lambda_{zh} = 2$, $\lambda_{xw} = 4$, $\lambda_{yw} = 1$, $\lambda_{yg} = 1$, $\lambda_{zw} = 1$, $\lambda_{zg} = 1$) / 12. Our model params is 560,944,640, disk footprint(in bytes, without compression) is 2,243,954,093.

Results Table 3 shows the results on dev and test21 set. Our baseline is the QE system without AMT data. +AMT method is the QE system with AMT data. In the experiments, we generate 3 different kinds of AMT data with the machine translation system trained for the WMT2021 Machine Translation of News Shared Task, Baidu Fanyi [4] and Google Translate[5]. For each kind of AMT, we run N (N=10) pass forward with dropout=0.1 using the a unified model trained with all AMT together. The expectations of 3N predictions of score and token labels is taken as the final answers.

5.3 Task3: Critical Error Detection

Table 4 shows the results of our system on development and blind test set. Experiments show that the best results obtained when applying XLMR-Large and FFN layer on development set. The involvement of AMT also brings significant improvement over all language pairs. For ensemble settings, we ensemble multiple models with different pre-trained models and classification layers using voting-based method as introduced in section 4.3.

In order to solve the problem of label imbalance, we also investigate different label weights when computing cross-entropy loss. Due to the large gap between the number of NOT and ERR labels in the dataset, the weights(NOT:ERR) are clipped as 1:6, 1:4, 1:5, 1:15 for enzh, ende, encs, enja. Meanwhile, to better fit the data in the test set and avoid over-fitting, we utilise dropout with rate of 0.1 and weight decay of $1e^{-5}$.

6 Conclusion

We present our work on WMT 2021 QE shared task in this paper. For all the three tasks to estimate sentence-level DA, token and sentence-level post-edit effort and sentence-level critical error, we employ predictor-estimator framework as our baseline. To further boost performance, we investigate the usage of additional high-quality translations. For task1, we mainly focus on introducing post-edits with multi-task learning. Also, the effect of data augmentation method based on MC dropout is studied here to improve the result of zero-shot pairs. For task 2 and 3, we generate high-quality translations for each observation using multiple well-trained machine translation systems. By directly concatenating AMT with the original source and target sentence then encoding it with pre-trained predictor, we achieved remarkable results over all language pairs and tasks. In future, we will continue to invest time and effort on studying the effect of involving additional translations into QE tasks, for example, how the additional translation quality will affect QE performance, what the better ways are to incorporate additional translations in.

[4] https://fanyi.baidu.com/

[5] https://translate.google.com/

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020. BERGAMOT-LATTE submissions for the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Fabio Kepler, Jonay Trénous, Marcos V. Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 78–84.

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*, 17:1–22.

Hyun Kim and Jong-Hyeok Lee. 2016. Recurrent neural network based translation quality estimation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 787–792.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. IST-unbabel participation in the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. *Quality Estimation for Machine Translation*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, and Liangyou Li. 2020. HW-TSC's participation at WMT 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061, Online. Association for Computational Linguistics.

Ensemble Fine-tuned mBERT for Translation Quality Estimation

Shaika Chowdhury *
University of Illinois at Chicago, US
schowd21@uic.edu

Naouel Baili
IQVIA, US
naouel.baili@iqvia.com

Brian Vannah
IQVIA, US
brian.vannah@iqvia.com

Abstract

Quality Estimation (QE) is an important component of the machine translation workflow as it assesses the quality of the translated output without consulting reference translations. In this paper, we discuss our submission to the WMT 2021 QE Shared Task. We participate in Task 2 sentence-level sub-task that challenge participants to predict the HTER score for sentence-level post-editing effort. Our proposed system is an ensemble of multilingual BERT (mBERT)-based regression models, which are generated by fine-tuning on different input settings. It demonstrates comparable performance with respect to the Pearson's correlation and beats the baseline system in MAE/ RMSE for several language pairs. In addition, we adapt our system for the zero-shot setting by exploiting target language-relevant language pairs and pseudo-reference translations.

1 Introduction

Progress in machine translation (MT) has accelerated due to the introduction of deep learning based approaches, dubbed as neural machine translation (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014). Several metrics (e.g., BLEU (Papineni et al., 2002), METEOR (Agarwal and Lavie, 2008)) are used to automatically evaluate the quality of the translations outputted by the NMT systems. However, these evaluation metrics require comparing the NMT outputs against human-prepared reference translations, which cannot be readily obtained. To tackle this predicament, recently quality estimation (QE) (Blatz et al., 2004; Specia et al., 2018) has emerged as an alternative evaluation approach for NMT systems. QE obviates the need for human judgements and hence can be efficiently integrated into the dynamic translation pipeline in the industry setting.

QE is performed at different granularity (e.g., word, sentence, document) (Kepler et al., 2019a); in this work we focus on the sentence-level post-editing effort task, which predicts the quality of the translated sentence as a whole in terms of the number of edit operations that need to be made to yield a post-edited translation, termed as HTER (Snover et al., 2006).

Sentence-level QE using neural approaches is generally treated as a supervised regression problem involving mainly two steps. In the first step, an encoder is used to learn vector representation/s of the source and translation sentences. While in the second step, the learned representations are passed through a sigmoid output layer to estimate the HTER score. These two steps can be performed either with a single model in an end-to-end fashion (e.g., Bi-RNN (Ive et al., 2018)), or using two separate models (e.g., POSTECH (Kim et al., 2017)). The different QE systems vary in their choice of the encoder, which range from RNN-based to Transformer-based models.

In this work, we leverage the fine-tuning capability of a Transformer-based encoder, namely the mBERT (Devlin et al., 2018) pre-trained model. Alongside the standard practice of feeding both the source and target (i.e., translation) sentences as the input sequence (Kepler et al., 2019a; Kim et al., 2019), we also explore other input settings based on only the target-side sentences (i.e., monolingual context). To this end, our final QE system is an ensemble of several mBERT models [1], each generated by fine-tuning on a different input combination comprising the source and/or target sentences. We experiment with the following three input settings: (1) both source and target, (2) just target and (3) both target and a randomly-sampled target sentence in the data forming the input se-

*work done during internship at IQVIA

[1] we also experimented with XLM-RoBERTa (Conneau et al., 2019) as the component model in our preliminary run; however, the results were worse compared to mBERT

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 897–903
November 10–11, 2021.

quence. Empirical analysis on 6 language pairs shows that the ensemble model is able to perform better than the individual fine-tuned models. Moreover, we provide experimental results for zero-shot QE, where training data for the test language pair is not available. This we tackle by improvising on the available training/dev data that match the target language of the test language pair and also by generating the pseudo-reference translations in that language.

2 Data

We use the WMT21 QE Shared Task 2 sentence-level data (Specia et al., 2021; Fomicheva et al., 2020a,b) for the following 7 language pairs: English-German (En-De), Romanian-English (Ro-En), Estonian-English (Et-En), Nepalese-English (Ne-En), Sinhala-English (Si-En), Russian-English (Ru-En) and Khmer-English (Km-En). Source-side data for each language pair includes sentences from Wikipedia articles, with part of the data gathered from Reddit articles for Ru-En. To obtain the translations, state-of-the-art MT models (Vaswani et al., 2017) built using fairseq toolkit (Ott et al., 2019) were used. The label for this task is the HTER score for the source-translation pair. Annotation was performed first at the word-level with the help of TER [2] tool. The word-level tags were then aggregated deterministically to obtain the sentence-level HTER score. The training, development, test and blind test data sizes for each language pair (except Km-En) are 7K, 1K, 1K and 1K instances respectively. As Km-En language pair was introduced for zero-shot prediction, only the test data containing 990 source and translation sentences was provided.

3 Our Approach

A key innovation in recent neural models lies in learning the contextualized representations by pre-training on a language modeling task. One such model, the multilingual BERT (mBERT) [3], is a transformer-based masked language model that is pre-trained on monolingual Wikipedia corpora of 104 languages with a shared word-piece vocabulary. Training the pre-trained mBERT model for a supervised downstream task, aka *finetuning*, has dominated performance across a wide spectrum of NLP tasks (Devlin et al., 2018). Our proposed approach leverages this fine-tuning capability of mBERT so as to form the component models in the ensemble QE system (Section 3.3). That is, each component model is a re-purposed mBERT that is fine-tuned for the sentence-level HTER score prediction task on one of the three input settings discussed in Section 3.2.

3.1 Fine-tuning mBERT for Regression

mBERT's model architecture is similar to BERT [4] and contains the following parameter settings: 12 layers, 12 attention heads and 768 hidden dimension per token. However, the only difference is that mBERT is trained on corpora of multiple languages instead of just on English. This enables mBERT to share representations across the different languages and hence can be conveniently used for all language pairs in the WMT21 data.

We first load the pre-trained mBERT model [5] and use its weights as the starting point of fine-tuning. The pre-trained mBERT is then trained on QE-specific input sequences (Section 3.2) for a few epochs such that the constructed sequence X is consumed by mBERT to output the contextualized representation $\mathbf{h} = (h_{CLS}, h_{x_1}, h_{x_2}, ..h_{x_T}, h_{SEP})$. Here, $[CLS]$ is a special symbol that denotes downstream classification and $[SEP]$ is for separating non-consecutive token sequences. Considering the final hidden vector of the $[CLS]$ token as the aggregate representation, it is then passed into the output layer with sigmoid activation to predict the HTER score:

$$y = sigmoid(\mathbf{W} \cdot \mathbf{h}_{CLS} + \mathbf{b}) \qquad (1)$$

$\mathbf{W}$ is a weight matrix for sentence-level QE fine-tuning that is trained along with all the parameters of mBERT end-to-end.

3.2 Input Settings

We construct the input sequence for each language pair in the following three ways:

SRC-MT: Given a source sentence $\mathbf{s} = (s_1, s_2, ...s_N)$ from a source language (e.g., English) and its translation $\mathbf{t} = (t_1, t_2, ...t_M)$ from a target language (e.g., German), we concatenate them together as $X = ([CLS], t_1, t_2, ...t_M, [SEP], s_1, s_2, ...s_N, [SEP])$ to form the input sequence.

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

[3] https://github.com/google-research/bert/blob/master/multilingual.md

[4] https://huggingface.co/bert-base-uncased

[5] https://huggingface.co/bert-base-multilingual-uncased

MT: The target sentence is only used to form the input sequence, $X = ([CLS], t_1, t_2, ...t_M, [SEP])$.

MT-MT':

Given the translation **t** for a source sentence **s**, we randomly sample another translation **t'** $= (t'_1, t'_2, ...t'_K)$ from the training data having HTER label close to **t** [6]. Although the source sentences for **t** and **t'** are different, we assume the additional monolingual context would help mBERT learn the correlating QE-specific features between **t** and **t'** for the target-side language. The resultant input sequence is $X = ([CLS], t_1, t_2, ...t_M, [SEP], t'_1, t'_2, ...t'_K, [SEP])$.

We fine-tune each of these mBERT models using AdamW optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) for two epochs with a batch size of 32 and a learning rate of $2e^{-5}$.

3.3 Ensemble Model

To take advantage of the individual strengths of the three mBERT component models fine-tuned on the aforementioned input settings, we combine their HTER score predictions by training an ensemble model. In particular, we experiment with three different ensemble models - Gradient Boosting (Friedman, 2001), AdaBoost (Freund and Schapire, 1997) and Average. For Gradient Boosting and AdaBoost we use the implementation in scikit-learn [7] with 10-fold cross validation. The settings for Gradient Boosting are: number of estimators 600, learning rate 0.01, minimum number of samples 3 and other default settings. We use the default settings for AdaBoost. In Average ensembling, we average the HTER score predictions by the three mBERT models. Our system submission to WMT21 is based on Gradient Boosting as it gave the best performance on the test data, as shown in Table 1.

3.4 Zero-Shot QE

Performing sentence-level QE in the zero-shot setting presents a unique challenge as the QE system is expected to predict HTER scores for sentences in a test language pair (e.g., Km-En) without having been trained on any instances from that test language pair. We address this by training on language pairs in the WMT21 QE data that match the target-side language (i.e., En) in the test language pair. The reason we focus on the target-side language is because the component mBERT models in the proposed ensemble QE system are fine-tuned on monolingual input sequences in the target-side language, which could potentially help the QE system generalize on the unseen test language pair. We consider the training and development data for the following language pairs in WMT21 QE data: Ro-En, Si-En, Et-En. Additionally, we augment this data by generating pseudo-references in the target language. A *pseudo-reference* (Scarton and Specia, 2014) is a translation for a source sentence that is outputted by a different NMT system than the one that produced the actual translations (e.g., transformer-based translation system proposed in (Vaswani et al., 2017)) and has shown to improve sentence-level QE performance (Soricut and Narsale, 2012). We use Google Translate [8] to get the pseudo-references in En for the Ro, Si and Et source sentences. The HTER scores for the translation and pseudo-reference pairs are then obtained using the TER tool. We train the ensemble QE system on the combined WMT21 QE data and the pseudo-reference parallel data, and test on the unseen test language pair.

Table 1: Performance of ENSBRT with different ensemble methods on the En-De test set.

	Avg	**AdaBoost**	**GradBoost**
Pearson's	0.266	0.458	**0.473**
Spearman's	0.249	0.436	**0.443**

4 Baseline

The baseline QE system (BASELINE) set by the WMT21 organizers this year is the Transformer-based Predictor-Estimator model (Kepler et al., 2019b; Moura et al., 2020). XLM-RoBERTa is used as the Predictor for feature generation. The baseline system is fine-tuned on the HTER scores and word-level tags jointly.

5 Results

Table 2 presents the experimental results of mBERT fine-tuned on the SRC-MT, MT and MT-MT'

[6] to ensure that **t'** is similar to **t**, we check that the difference between their HTER scores is within 0.1

[7] https://scikit-learn.org

[8] https://github.com/lushan88a/google_trans_new

Table 2: Performance in Pearson's correlation of mBERT fine-tuned with different input settings on the test set. ENSBRT is the proposed ensemble mBERT QE system.

	En-De	**Ro-En**	**Ru-En**	**Si-En**	**Et-En**	**Ne-En**
SRC-MT	0.389	0.793	0.400	0.526	0.601	0.489
MT	0.469	0.762	0.374	0.552	0.580	0.491
MT-MT'	0.431	0.761	0.350	0.492	0.556	0.454
ENSBRT	**0.473**	**0.802**	**0.418**	**0.576**	**0.632**	**0.525**

Table 3: Performance of BASELINE and ENSBRT on the WMT21 blind test set for different language pairs. Bold indicates ENSBRT beats BASELINE in that metric.

		En-De	**Ro-En**	**Ru-En**	**Si-En**	**Et-En**	**Ne-En**	**Km-En**
BASELINE	Pearson's ↑	0.529	0.831	0.448	0.607	0.714	0.626	0.576
	MAE ↓	0.183	0.142	0.255	0.204	0.195	0.205	0.241
	RMSE ↓	0.129	0.115	0.188	0.159	0.149	0.160	0.196
ENSBRT	Pearson's ↑	0.519	0.795	0.376	0.522	0.666	0.572	0.529
	MAE ↓	**0.171**	0.171	**0.251**	0.206	**0.171**	**0.176**	0.262
	RMSE ↓	0.129	0.141	0.189	0.162	**0.132**	**0.139**	0.197

input settings, as well as the performance of the ensemble of the three mBERT models, which we call *ENSBRT*. First, comparing among the three input settings, it seems that mBERT exhibits competitive results even when it does not have knowledge of the source-side text in the MT and MT-MT' settings, in particular for the following language pairs - En-DE, Si-En, Ne-En. While the ensemble mBERT model, ENSBRT, outperforms the independent counterparts for all the language pairs. This shows that the ensemble method can help to balance out the weakness of any component model, thereby benefiting the sentence-level QE task overall. We also visualize ENSBRT's predictions against the ground truth HTER scores in Figure 1.

Table 3 compares the QE performance between the BASELINE and ENSBRT in terms of Pearson's correlation, RMSE and MAE on the WMT21 blind test set, for which the ground truth HTER scores were not available at the time. We submitted results for 6 language pairs (En-De, Ro-En, Ru-En, Si-En, Et-En, Ne-En) in the normal QE setting and one language pair (Km-En) for zero-shot prediction. ENSBRT demonstrates comparable performance to the BASELINE for Pearson's and outperforms it in either MAE or RMSE for the following language pairs: En-De, Ru-En, Et-En and Ne-En.

6 Conclusion

In this work, we describe the *ENSBRT* system submission to the WMT21 QE Shared Task. ENSBRT is based on fine-tuning the multilingual BERT pretrained model for sentence-level translation quality score prediction. We explore three different input settings for fine-tuning which include either bilingual or monolingual context, and combine the predictions of the three models using ensemble methods as our final system. Furthermore, zero-shot QE is facilitated by using labeled data for existing language pairs and pseudo-references that align with the target language of the unseen test data.

References

Abhaya Agarwal and Alon Lavie. 2008. Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 115–118.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

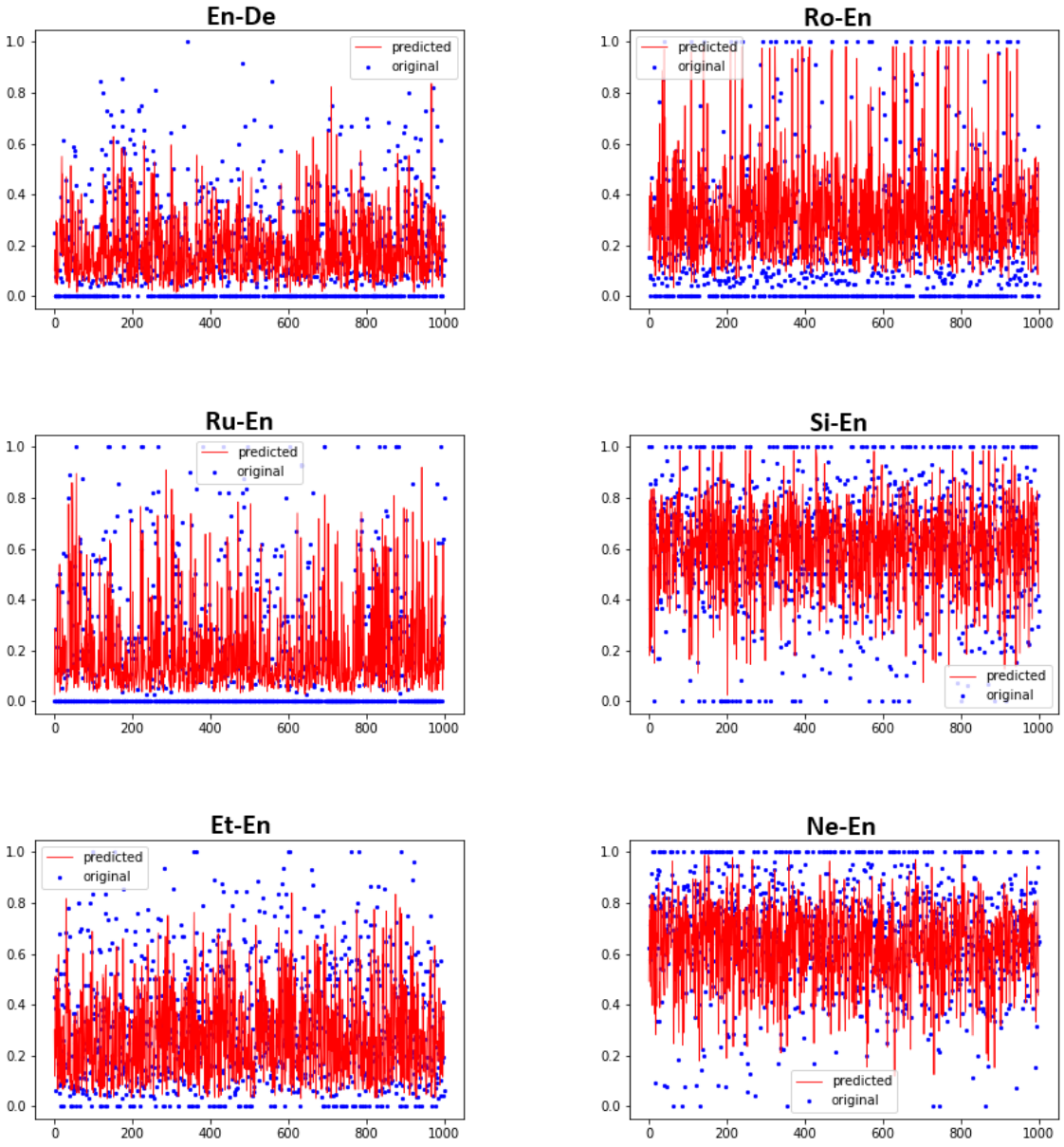


Figure 1: Visualization comparing HTER score predictions by ENSBRT (i.e., predicted (red)) against the gold labels (i.e., original (blue)) for 6 language pairs on the test set. X-axis represents each data point and Y-axis is the HTER score. The closer the corresponding red line and blue dot are to each other the better, as we expect the HTER prediction to be same as or close to the ground truth.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, pages 315–321.

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*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. MLQE-PE: A multilingual quality estimation and post-editing dataset. *arXiv preprint arXiv:2010.04480*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020b. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

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.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. Deepquest: a framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019a. Unbabel's participation in the wmt19 translation quality estimation shared task. *arXiv preprint arXiv:1907.10352*.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André FT Martins. 2019b. Openkiwi: An open source framework for quality estimation. *arXiv preprint arXiv:1902.08646*.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. 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):1–22.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. Qe bert: bilingual bert using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 85–89.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Joao Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André FT Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

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.

Carolina Scarton and Lucia Specia. 2014. Document-level translation quality estimation: exploring discourse and pseudo-references. In *Proceedings of the 17th Annual conference of the European Association for Machine Translation*, pages 101–108.

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: Technical Papers*, pages 223–231.

Radu Soricut and Sushant Narsale. 2012. Combining quality prediction and system selection for improved automatic translation output. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 163–170.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

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, 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.

The JHU-Microsoft Submission for WMT21 Quality Estimation Shared Task

Shuoyang Ding†* **Marcin Junczys-Dowmunt**‡
Matt Post†‡ **Christian Federmann**‡ **Philipp Koehn**†
† Center for Language and Speech Processing, Johns Hopkins University ‡ Microsoft
{dings, phi}@jhu.edu {marcin.junczysdowmunt, mattpost, chrife}@microsoft.com

Abstract

This paper presents the JHU-Microsoft joint submission for WMT 2021 quality estimation shared task. We only participate in Task 2 (post-editing effort estimation) of the shared task, focusing on the target-side word-level quality estimation. The techniques we experimented with include Levenshtein Transformer training and data augmentation with a combination of forward, backward, round-trip translation, and pseudo post-editing of the MT output. We demonstrate the competitiveness of our system compared to the widely adopted OpenKiwi-XLM baseline. Our system is also the top-ranking system on the MT MCC metric for the English-German language pair.

1 Introduction

In the machine translation (MT) literature, quality estimation (QE) refers to the task of evaluating the translation quality of a system without using a human-generated reference. There are several different granularities as to the way those quality labels or scores are generated. Our participation in the WMT21 quality estimation shared task focuses specifically on the *word-level* quality labels (word-level subtask of Task 2), which are helpful for both human (Lee et al., 2021) and automatic (Lee, 2020a) post-editing of translation outputs. The task asks the participant to predict one binary quality label (`OK`/`BAD`) for each target word and each gap between target words, respectively.[1]

Our approach closely follows our contemporary work (Ding et al., 2021), which focuses on en-de and en-zh language pairs tested in the 2020 version of the shared task. The intuition behind our idea is that translation knowledge is very useful for predicting word-level quality labels of translations. However, usage of machine translation models is limited in the previous work mainly due to (1) the difficulties in using both the left and right context of an MT word to be evaluated; (2) the difficulties in making the word-level reference labels compatible with subword-level models; and (3) the difficulties in enabling translation models to predict gap labels. To resolve these difficulties, we resort to Levenshtein Transformer (LevT, Gu et al., 2019), a model architecture designed for non-autoregressive neural machine translation (NA-NMT). Because of its iterative inference procedure, LevT is capable of performing post-editing on existing translation output even just trained for translation. To further improve the model performance, we also propose to initialize the encoder and decoder of the LevT model with those from a massively pre-trained multilingual NMT model (M2M-100, Fan et al., 2020).

Starting from a LevT translation model, we then perform a two-stage finetuning process to adapt the model from translation prediction to quality label prediction, using automatically-generated pseudo-post-editing triplets and human post-editing triplets respectively. All of our final system submissions are also linear ensembles from several individual models with weights optimized on the development set using the Nelder-Mead method (Nelder and Mead, 1965).

2 Method

Our system building pipeline is consisted of three different stages:

- **Stage 1**: Training LevT for translation
- **Stage 2 (Optional)**: Finetuning LevT on synthetic post-editing triplets
- **Stage 3**: Finetuning LevT on human post-editing triplets

Stage 1: Training LevT for Translation We largely follow the same procedure as Gu et al.

* Shuoyang Ding had a part-time affiliation with Microsoft at the time of this work.

[1] While there is another sub-task for predicting source-side quality labels, we do not participate in that task.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 904–910
November 10–11, 2021.

(LevT, 2019) to train the LevT translation model, except that we initialize the embedding, the encoder, and decoder of LevT with those from the small M2M-100-small model (418M parameters, Fan et al., 2020) to take advantage of large-scale pretraining. Because of that, we also use the same sentencepiece model and vocabulary as the M2M-100 model.

For to-English language pairs, we explored training multi-source LevT model. According to the results on devtest data, this is shown to be beneficial for the QE task for ro-en, ru-en and ne-en, but not for other language pairs.

Stage 2: Synthetic Finetuning During both finetuning stages, we update the model parameters to minimize the NLL loss of word quality labels and gap quality labels, for the deletion and insertion head, respectively. To obtain training targets for finetuning, we need *translation triplet data*, i.e., the aligned triplet of source, target, and post-edited segments. Human post-edited data naturally provides all three fields of the triplet, but only comes in a limited quantity. To further help the model to generalize, we conduct an extra step of finetuning on synthetic translation triplets, similar to some previous work (Lee, 2020b, *inter alia*). We explored five different methods for data synthesis, namely:

1. `src-mt-tgt`: Take the source side of a parallel corpus (`src`), translate it with a MT model to obtain the MT output (`mt`), and use the target side of the parallel corpus as the pseudo post-edited output (`tgt`).

2. `src-mt1-mt2`: Take a corpus in the source language (`src`) and translate it with two different MT systems that have clear system-level translation quality orderings. Then, take the worse MT output as the MT output in the triplet (`mt1`) and the better as the pseudo post-edited output in the triplet (`mt2`).

3. `bt-rt-tgt`: Take a corpus in the target language (`tgt`) and back-translate it into the source langauge (`bt`), and then translate again to the target language (`rt`). We then use `rt` as the MT output in the triplet and `tgt` as the pseudo post-edited output in the triplet.

4. `src-rt-ft`: Take a parallel corpus and translate its source side and use it as the pseudo post-edited output (`ft`), and round-trip translate its target side (`rt`) as the MT output in the translation triplet.

5. `Multi-view Pseudo Post-Editing (MVPPE)`: Same as Ding et al. (2021), we take a parallel corpus and translate the source side (`src`) with a multilingual translation system (`mt`) as the MT output in the triplet. We then generate the pseudo-post-edited output by ensembling two different *views* of the same model: (1) using the multilingual translation model as a translation model, with `src` as the input; (2) using the multilingual translation model as a paraphrasing model, with `tgt` as the input. The ensemble process is the same as ensembling standard MT models, and we perform beam search on top of the ensemble. Unless otherwise specified, we use the same ensembling weights of $\lambda_t = 2.0$ and $\lambda_p = 1.0$ as Ding et al. (2021).

Stage 3: Human Post-editing Finetuning We follow the same procedure as stage 2, except that we finetune on the human post-edited dataset provided by the shared task organizers for this stage.

Compatibility With Subwords As pointed out before, since LevT predicts edits on a subword-level starting from translation training, we must construct reference tags that are compatible with the subword segmentation done for both the MT and the post-edited output. Specifically, we need to: (1) for inference, convert subword-level tags predicted by the model to word-level tags for evaluation, and (2) for both finetuning stages, build subword-level reference tags. We follow the same heuristic subword-level tag reference construction procedure as Ding et al. (2021), which was shown to be helpful for task performance.

Label Imbalance Like several previous work (Lee, 2020a; Wang et al., 2020; Moura et al., 2020), we also observed that the translation errors are often quite scarce, thus creating a skewed label distribution over the `OK` and `BAD` labels. Since it is critical for the model to reliably predict both classes of labels, we introduce an extra hyperparameter μ in the loss function that allows us to upweight the words that are classified with `BAD` tags in the reference.

$$\mathcal{L} = \mathcal{L}_{OK} + \mu\mathcal{L}_{BAD}$$

Ensemble For each binary label prediction made by the model, the model will give a score $p(\texttt{OK})$,

Language Pair	Data Source	Sentence Pairs
English-German	WMT20 en-de parallel data	44.2M
English-Chinese	shared task en-zh parallel	20.3M
Romanian-English	shared task ro-en parallel	3.09M
Russian-English	shared task ru-en parallel	2.32M
Estonian-English	shared task et-en parallel	880K
Estonian-English	shared task et-en parallel + NewsCrawl 14-17	3.42M
Nepalese-English	shared task ne-en parallel	498K
Pashto-English	WMT20 Parallel Corpus Filtering Task	347K

Table 1: Source and statistics of parallel datasets used in our experiments.

which are translated into binary labels in post-processing. To ensemble predictions from k models $p_1(\texttt{OK}), p_2(\texttt{OK}), \ldots, p_k(\texttt{OK})$, we perform a linear combination of the scores for each label:

$$p_E(\texttt{OK}) = \lambda_1 p_1(\texttt{OK}) + \lambda_2 p_2(\texttt{OK}) + \cdots + \lambda_k p_k(\texttt{OK})$$

To determine the optimal interpolation weights, we optimize towards target-side MCC on the development set. Because the target-side MCC computation is not implemented in a way such that gradient information can be easily obtained, we experimented with two gradient-free optimization method: Powell method (Powell, 1964) and Nelder-Mead method (Nelder and Mead, 1965), both as implemented in `SciPy` (Virtanen et al., 2020). We found that the Nelder-Mead method finds better optimum on the development set while also leading to better performance on the devtest dataset (not involved in optimization). Hence, we use the Nelder-Mead optimizer for all of our final submissions with ensembles. We set the initial points of Nelder-Mead optimization to be the vertices of the standard simplex in the k-dimensional space, with k being the number of the models.

We find that it is critical to build ensembles from models that yield diverse yet high-quality outputs. Specifically, we notice that ensembles from multiple checkpoints of a single experimental run are not helpful. Hence, for each language pair, we select 2-8 models with different training configurations that also have the highest performance to build our final ensemble model for submission.

3 Experiments

3.1 Data Setup

LevT Training We used the same parallel data that was used to train the MT system in the shared task, except for the en-de, et-en, and ps-en language pairs. For en-de language pair, we use the larger parallel data from the WMT20 news translation shared task. For et-en language pair, we experiment with augmenting with the News Crawl Estonian monolingual data from 2014 to 2017, which was inspired by Zhou and Keung (2020). For ps-en language pair, because there is no MT system provided, we take the data from the WMT20 parallel corpus filtering shared task and applied the baseline LASER filtering method. For the multi-source LevT model, we simply concatenate the data from ro-en, ru-en, es-en (w/o monolingual augmentation) and ne-en. The resulting data scale is summarized in Table 1.

Following the setup in Gu et al. (2019), we conduct sequence-level knowledge distillation during training for all language pairs except for ne-en and ps-en[2]. For en-de, the knowledge distillation data is generated by the WMT19 winning submission for that language pair from Facebook (Ng et al., 2019). For en-zh, we train our own en-zh autoregressive model on the parallel data from the WMT17 news translation shared task. For the other language pairs, we use the decoding output from M2M-100-mid (1.2B parameters) model to perform knowledge distillation.

Synthetic Finetuning We always conduct data synthesis based on the same parallel data that was used to train the LevT translation model. For the only language pair (en-de) where we applied the `src-mt1-mt2` synthetic finetuning for shared task submission, we again use the WMT19 Facebook's winning system (Ng et al., 2019) to generate the higher-quality translation `mt2`, and the system provided by the shared task to generate the MT output in the pseudo translation triplet `mt1`. For all other combinations of translation directions, language pairs and MVPPE decoding, we use the M2M-100-

[2]The exception was motivated by the poor quality of the translation we obtained from the M2M-100 model.

Configuration	Stage 2	Stage 3	Target MCC
en-de OpenKiwi	N	default	0.337
en-de bilingual best	src-mt1-mt2	$\mu = 1.0$	0.500
en-de ensemble	N/A	N/A	0.504
en-zh OpenKiwi	N	default	0.421
en-zh bilingual best	mvppe	$\mu = 1.0$	0.459
en-zh ensemble	N/A	N/A	0.466
ro-en OpenKiwi	N	default	0.556
ro-en bilingual best	src-rt-ft	$\mu = 1.0$	0.604
ro-en multilingual best	N	$\mu = 1.0$	0.612
ro-en ensemble	N/A	N/A	0.633
ru-en OpenKiwi	N	default	0.279
ru-en bilingual best	src-rt-ft	$\mu = 3.0$	0.316
ru-en multilingual best	N	$\mu = 3.0$	0.339
ru-en ensemble	N/A	N/A	0.349
et-en OpenKiwi	N	default	0.503
et-en bilingual best	N	$\mu = 3.0$	0.556
et-en bilingual best (w/ aug)	N	$\mu = 3.0$	0.548
et-en multilingual best	N	$\mu = 3.0$	0.533
et-en ensemble	N/A	N/A	0.575
ne-en OpenKiwi	N	default	0.664
ne-en bilingual best	N	$\mu = 3.0$	0.677
ne-en multilingual best	N	$\mu = 3.0$	0.681
ne-en ensemble	N/A	N/A	0.688

Table 2: Target MCC results on `test20` dataset for all language pairs we submitted systems for (except for ps-en which is not included in `test20`). Stage 2 stands for synthetic finetuning (where N stands for not performing this stage). Stage 3 stands for human annotation finetuning. μ stands for the label balancing factor.

	Target MCC	F1-OK	F1-BAD
N	0.489	0.955	0.533
src-mt-ref	0.493	0.955	0.537
src-mt1-mt2	**0.500**	0.956	**0.544**
bt-rt-tgt	0.490	0.956	0.534
src-rt-ft	0.494	0.956	0.538
mvppe	**0.500**	**0.960**	0.540

Table 3: Analysis of different data synthesis methods on en-de language pair. All models here are initialized with M2M-100-small.

mid (1.2B parameters) model.

Human Annotation Finetuning We follow the data split for human post-edited data as determined by the task organizers and use `test20` as the devtest for our system development purposes.

Reference Tag Generation We implemented another TER computation tool[3] to generate the word-level and subword-level tags that we use as the reference for finetuning, but stick to the original reference tags in the test set for evaluation to avoid potential result mismatch.

[3] `https://github.com/marian-nmt/moses-scorers`

3.2 Model Setup

Our LevT-QE model is implemented based on Fairseq (Ott et al., 2019). All of our experiments uses Adam optimizer (Kingma and Ba, 2015) with linear warmup and `inverse-sqrt` scheduler. For stage 1, we use the same hyperparameters as Gu et al. (2019) for LevT translation training, but use a smaller learning rate of 2e-5 to avoid overfitting for all to-English language pairs. For stage 2 and beyond, we stick to the learning rate of 2e-5 and perform early-stopping based on the loss function on the development set. For stage 3, e also experiment with label balancing factor $\mu = 1.0$ and $\mu = 3.0$ for each language pair and pick the one that works the best on devtest data, while for stage 2 we keep $\mu = 1.0$ because early experiments indicate that using $\mu = 3.0$ at this stage is not helpful.

For pre-submission developments, we built OpenKiwi-XLM baselines (Kepler et al., 2019) following their `xlmroberta.yaml` recipe. Keep in mind due to the fact that this baseline model is initialized with a much smaller XLM-Roberta-base model (281M parameters) compared to our M2M-100-small initialization (484M parameters), the performance comparison is not a strict one.

3.3 Devtest Results

Our system development results on `test20` devtest data are shown in Table 2[4]. In all language pairs, our systems can outperform the OpenKiwi baseline based upon the pre-trained XLM-RoBERTa-base encoder. Among these language pairs, the benefit of LevT is most significant on the language pairs with a large amount of available parallel data. Such behavior is expected, because the less parallel data we have, the less knowledge we can extract from the LevT training process. Furthermore, the lack of good quality knowledge distillation data in the low-resource language pairs also expands this performance gap. To our best knowledge, this is also the first attempt to train non-autoregressive translation systems under low-resource settings, and we hope future explorations in this area can enable us to build a better QE system from LevT.

In terms of comparison between multilingual and bilingual models for to-English language pairs, the results are mixed, with the multilingual model per-

[4] Note that the results on en-zh also reflect a crucial bug fix on our TER computation tool that we added after the system submission deadline. Hence the results shown here are from a different system as in the official shared task results. The bug fix should not affect the results of the other language pairs.

Configuration	Stage 2	Stage 3	Target MCC	F1-OK	F1-BAD
ro-en multilingual	N	$\mu = 1.0$	**0.612**	0.949	**0.659**
ro-en multilingual	mvppe	$\mu = 1.0$	0.611	**0.951**	**0.659**
ro-en multilingual	src-mt1-mt2 (Bing mt2)	$\mu = 1.0$	0.585	0.936	0.630
ro-en bilingual (Bing KD)	N	$\mu = 1.0$	0.581	0.949	0.632
ro-en bilingual (Bing KD)	src-mt1-mt2 (Bing mt2)	$\mu = 1.0$	0.568	0.938	0.619
et-en bilingual	N	$\mu = 3.0$	0.548	0.914	0.622
et-en bilingual	mvppe	$\mu = 3.0$	0.544	**0.929**	0.615
et-en bilingual	src-mt1-mt2 (Bing mt2)	$\mu = 3.0$	**0.563**	0.919	**0.634**
et-en bilingual (Bing KD)	N	$\mu = 3.0$	0.557	0.918	0.629
et-en bilingual (Bing KD)	src-mt1-mt2 (Bing mt2)	$\mu = 3.0$	0.559	0.916	0.631

Table 4: Analysis of `src-mt1-mt2` and `mvppe` method on ro-en and et-en language pair.

forming significantly better for ru-en language pair, but significantly worse for et-en language pair. Finally, our Nelder-Mead ensemble further improves the result by a small but steady margin.

3.4 Analysis

Ding et al. (2021) already conducted comprehensive ablation studies for techniques such as the effect of LevT training step, heuristic subword-level reference tag, as well as the effect of various data synthesis methods. In this section, we extend the existing analyses by studying if the synthetic finetuning is still useful with M2M initialization, and if it is universally helpful across different languages. We also examine the effect of label balancing factor μ and take a detailed look at the prediction errors.

Synthetic Finetuning We redo the analysis on en-de synthetic finetuning with the smaller 2M parallel sentence samples from Europarl, as in Ding et al. (2021), but with the updated `test20` test set and models with M2M-100-small initialization. The results largely corroborate the trend in the other paper, showing that `src-mt1-mt2` and `mvppe` being the most helpful two data synthesis methods. We then extend those two most helpful methods to ro-en and et-en, using the up-to-date Bing Translator production model as the stronger MT system (a.k.a. `mt2`) in the `src-mt1-mt2` synthetic data. The result is mixed, with `mvppe` failing to improve performance for both language pairs, and `src-mt1-mt2` only being helpful for et-en language pair. We also trained two extra ro-en and et-en LevT models using the respective Bing Translator models to generate the KD data, which are neither helpful for improving performance on their own nor working better with `src-mt1-mt2` synthetic data.

We notice that the `mvppe` synthetic data seems to significantly improve the F1 score of the `OK` label in general, for which we don't have a good explanation yet.

Configuration	Target MCC	F1-OK	F1-BAD
ro-en $\mu = 1.0$	**0.612**	**0.949**	**0.659**
ro-en $\mu = 3.0$	0.577	0.930	0.619
ru-en $\mu = 1.0$	0.267	**0.960**	0.284
ru-en $\mu = 3.0$	**0.339**	0.943	**0.390**
et-en $\mu = 1.0$	0.478	**0.933**	0.511
et-en $\mu = 3.0$	**0.512**	0.925	**0.587**
ne-en $\mu = 1.0$	0.660	**0.885**	0.774
ne-en $\mu = 3.0$	**0.681**	0.855	**0.788**

Table 5: Analysis of different label balancing factors initialized on to-English language pairs. All results are based on the multilingual model and not performing synthetic finetuning step.

Label Balancing Factor We find the QE task performance to be quite sensitive to the label balancing factor μ, but there is also no universally optimal value for all language pairs. Table 5 shows this behavior for all to-English language pairs. Notice that while for most of the cases μ simply controls a trade-off between the performance of `OK` and `BAD` outputs, there are also cases such as ro-en where a certain choice of μ hurts the performance of both classes. This might be due to a certain label class being particularly hard to fit, thus creating more difficulties with learning when the loss function is designed to skew to this label class.

It should be noted that this label balancing factor does not correlate directly with the ratio of the `OK` vs. `BAD` labels in the training set. For example, to obtain the best performance, ne-en requires $\mu = 3.0$ while en-de requires $\mu = 1.0$, while the `OK` to `BAD` ratio for ne-en (2.14:1) is much less skewed

Lang.	Tgt. MCC	MT MCC	MT BAD (P/R/F1)			MT OK (P/R/F1)			GAP MCC	GAP BAD (P/R/F1)			GAP OK (P/R/F1)		
en-de	0.504	0.503	0.476	0.731	0.576	0.950	0.863	0.904	0.280	0.366	0.238	0.288	0.980	0.989	0.984
en-zh	0.466	0.381	0.467	0.787	0.586	0.879	0.633	0.736	0.146	0.276	0.099	0.145	0.965	0.990	0.977
ro-en	0.612	0.645	0.729	0.709	0.719	0.922	0.929	0.926	0.164	0.411	0.073	0.125	0.973	0.997	0.985
ru-en	0.349	0.329	0.296	0.675	0.411	0.945	0.775	0.852	0.167	0.265	0.123	0.168	0.978	0.991	0.985
et-en	0.575	0.553	0.676	0.681	0.679	0.875	0.873	0.874	0.251	0.426	0.169	0.242	0.967	0.991	0.979
ne-en	0.694	0.434	0.760	0.918	0.832	0.746	0.454	0.564	0.192	0.444	0.098	0.161	0.955	0.994	0.974

Table 6: Detailed evaluation metric breakdown of all submitted ensemble system on `test20` test set.

compare to en-de (10.2:1).

Detailed Error Breakdown We found it hard to develop an intuition for the model performance from the MCC metric. To further understand which label categories our models struggle with the most, we breakdown the target-side metric into a cross product of {`MT`, `GAP`} tags and {`OK`, `BAD`} classes and compute precision, recall and F1-score for each category. The breakdown is shown in Table 6. It can be seen that our model is making the most mistakes with the `GAP BAD` category, while the category with the least mistakes is the `GAP OK` category. Also, note that for MT word tags, the models often seem to suffer more from low precision rather than low recall, while for gaps it is the opposite.

Overall, we see that the highest F1 scores we can achieve for detecting bad MT words or gaps are rarely higher than 0.8, which indicates that there should be ample room for improvement. It would also be interesting to measure the inter-annotator agreement of these word-level quality labels, in order to get a sense of the human performance we should be aiming for.

4 Conclusion

In this paper, we present our WMT21 word-level QE shared task submission based on Levenshtein Transformer training and a two-step finetuning process. We also explore various ways to create synthetic data to build more generalizable systems with limited human annotations. We show that our system outperforms the OpenKiwi+XLM baseline for all language pairs we experimented with. Our official results on the blind test set also demonstrate the competitiveness of our system. We hope that our work can inspire other applications of Levenshtein Transformer beyond the widely studied case of non-autoregressive translation.

References

Shuoyang Ding, Marcin Junczys-Dowmunt, Matt Post, and Philipp Koehn. 2021. Levenshtein training for word-level quality estimation.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Dongjun Lee. 2020a. Cross-lingual transformers for neural automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 772–776, Online. Association for Computational Linguistics.

Dongjun Lee. 2020b. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online. Association for Computational Linguistics.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021. IntelliCAT: Intelligent machine translation post-editing with quality estimation and translation suggestion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 11–19, Online. Association for Computational Linguistics.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. IST-unbabel participation in the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036, Online. Association for Computational Linguistics.

John A. Nelder and R. Mead. 1965. A simplex method for function minimization. *Comput. J.*, 7(4):308–313.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

M. J. D. Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.*, 7(2):155–162.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, and Liangyou Li. 2020. HW-TSC's participation at WMT 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061, Online. Association for Computational Linguistics.

Jiawei Zhou and Phillip Keung. 2020. Improving non-autoregressive neural machine translation with monolingual data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1893–1898, Online. Association for Computational Linguistics.

TUDa at WMT21: Sentence-Level Direct Assessment with Adapters

Gregor Geigle, Jonas Stadtmüller, Wei Zhao, Jonas Pfeiffer, Steffen Eger
Technische Universität Darmstadt
{gregortheodor.geigle,jonas.stadtmueller}@stud.tu-darmstadt.de
{zhao,eger}@aiphes.tu-darmstadt.de
pfeiffer@ukp.informatik.tu-darmstadt.de

Abstract

This paper presents our submissions to the WMT2021 Shared Task on Quality Estimation, Task 1 *Sentence-Level Direct Assessment*. While top-performing approaches utilize massively multilingual Transformer-based language models which have been pre-trained on all target languages of the task, the resulting insights are limited, as it is unclear how well the approach performs on languages unseen during pre-training; more problematically, these approaches do not provide any solutions for *extending* the model to new languages or unseen scripts—arguably one of the objectives of this shared task. In this work, we thus focus on utilizing massively multilingual language models which only *partly* cover the target languages during their pre-training phase. We extend the model to new languages and unseen scripts using recent adapter-based methods and achieve on par performance or even surpass models pre-trained on the respective languages.

1 Introduction

In Machine Translation (MT), the Quality Estimation (QE) task attempts to characterize the quality of a translation, without the availability of a (gold-label) reference translation. The introduction of a QE system would consequently allow for the automatic analysis of machine-translated sentences without costly human reference translation, with numerous applications, such as: the selection of candidate translations, the estimation of human editing effort, or the detection of low-quality or misleading translations (Kepler et al., 2019). However, in order to acquire training data, professional human translators are required to score the translation quality of many examples, making labeled data difficult to obtain, especially for low-resource languages. This highlights the importance of cross-lingual zero-shot transfer of QE systems, one of the objectives of the WMT21 shared task (Specia et al., 2021), which introduces zero-shot evaluation sets of four new language pairs.

Previous approaches have predominantly focused on languages for which training data is available, such as the QE task at WMT20. The best results were obtained by fine-tuning massively multilingual Transformer-based language models (Vaswani et al., 2017) such as multilingual BERT (mBERT) (Devlin et al., 2019) or XLM-R (Conneau et al., 2020) (Specia et al., 2020; Ranasinghe et al., 2020b; Sun et al., 2020a; Nakamachi et al., 2020, *inter alia*), on the target QE tasks. These supervised methods considerably outperform unsupervised methods (Zhao et al., 2020; Fomicheva et al., 2020c; Sun et al., 2020a; Zhao et al., 2021; Song et al., 2021) even in zero-shot settings (Sun et al., 2020a). However, analyzing the applicability of fine-tuning multilingual models on the target language pairs that are covered during pre-training considerably limits the generated insights. They are only applicable to the $\sim$100 languages covered during pre-training, excluding the remaining majority of languages as the "curse-of-multilinguality" (Conneau et al., 2020) prohibits the over 7000 languages in the world (Joshi et al., 2020) to be represented within a single model

In this work, we thus aim to address these limitations by utilizing multilingual language models that only cover a subset of the target languages. Here we focus on mBERT which—in contrast to XLM-R—has not seen the languages Sinhala, Pashto, and Khmer, all part of the WMT21 shared task. As the script of Sinhala and Khmer are not included in the mBERT vocabulary, it is impossible for the corresponding tokenizer to correctly tokenize text in those languages. Following Pfeiffer et al. (2020b, 2021b) we thus propose an adapter-based approach to extend mBERT to new languages and new scripts.

Our contributions are as follows: **1)** we analyze adapter-based supervised approaches for QE

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 911–919

and demonstrate their competitive performance compared to full model fine-tuning, both in supervised as well as zero-shot settings; **2)** we use recent adapter based methods to extend mBERT to unseen languages and scripts, achieving considerable performance gains over standard mBERT for unseen languages; **3)** we demonstrate competitive performance of our adapted mBERT approach compared to XLM-R, which has seen the respective languages during pre-training. We release our code and adapters at https://github.com/Aaronsom/wmt21-qe-tudarmstadt/.

2 Method

We describe our adapter-based approaches for supervised QE and the extension to unseen languages.

2.1 Task Formulation

We model QE as a regression task. The Transformer receives as input both the source sentence and the translation hypothesis and is trained to predict the quality score for the sentence pair. For this, we take the final contextualized representation of the special [CLS]-token produced by the Transformer and feed it into a multi-layer regression head to compute the predicted quality $f(s,t)$:

$$f(s,t) = \mathbf{W}_2 \cdot (\tanh(\mathbf{W}_1 \cdot \mathbf{r}_{[CLS]}(s,t))) \quad (1)$$

with $\mathbf{W}_1 \in \mathbb{R}^{h \times h}$, $\mathbf{W}_2 \in \mathbb{R}^{1 \times h}$, tanh is the hyperbolic tangent, h is the hidden dimension of the Transformer, and $\mathbf{r}_{[CLS]}(s,t)$ is the output representation of the [CLS]-token for the source-translation input pair s,t. We train the model using mean squared error.

2.2 Adapters

Adapters are randomly initialized weights, newly introduced at every layer of the pre-trained Transformer model. During fine-tuning, *only* the adapter weights (and the regression head) are updated while the remaining model weights are kept frozen.

Houlsby et al. (2019) propose a feed-forward bottleneck adapter architecture consisting of a down-projection, a non-linearity, and finally an up-projection, both after the multi-attention as well as after the feed-forward component at every Transformer layer. We use the adapter architecture proposed by Pfeiffer et al. (2021a) which achieves on par results while reducing the number of trainable parameters of Houlsby et al. (2019) by only placing adapters after the feed-forward component (see Figure 1a). Adapters at layer l are defined as follows:

$$a_l(\mathbf{h}_l, \mathbf{r}_l) = \mathbf{U}_l \cdot (\mathrm{ReLU}(\mathbf{D}_l \cdot \mathbf{h}_l)) + \mathbf{r}_l \quad (2)$$

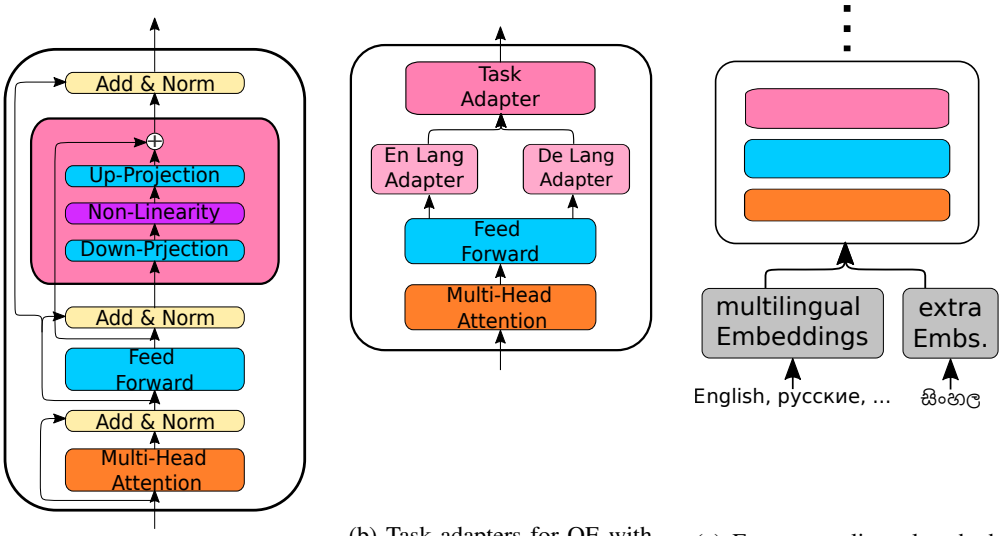
where $\mathbf{D}_l \in \mathbb{R}^{\lfloor \frac{h}{r} \rfloor \times h}$, $\mathbf{U}_l \in \mathbb{R}^{h \times \lfloor \frac{h}{r} \rfloor}$, ReLU is the rectified linear unit, $\mathbf{h}_l$ is the hidden input representation, $\mathbf{r}_l$ is the residual after the fully-connected layer, and r is the reduction factor—a hyperparameter that decides how much the adapter compresses the hidden representation.

2.3 Extending to Unseen Languages

While both XLM-R and mBERT have been pre-trained on a large number of languages, XLM-R has seen all languages appearing in the WMT21 dataset, while mBERT has not been pre-trained on Sinhala, Khmer, and Pashto. Further, the scripts of Sinhala and Khmer are not covered by mBERT's vocabulary. We thus follow Pfeiffer et al. (2020b, 2021b) to extend both the latent Transformer as well as input embedding representations to the respective languages, using adapter-based approaches.

Language Adapters. Language adapters (LAs) (Pfeiffer et al., 2020b) are trained to encode idiosyncratic, language-specific information, and transform the underlying multilingual model's latent representations to better align with the respective languages. Correspondingly, they are trained monolingually using the masked language modeling (MLM) objective on unlabeled textual data in the target language.

Extending to unseen scripts. Word piece tokenizers can (arguably inadequately (Rust et al., 2021)) tokenize unseen languages that are written in seen scripts, with a fall-back character-level tokenization. Unfortunately, these tokenizers fail for unseen scripts, as even character-level tokens are not part of the vocabulary, leaving the tokenizer only with instantiating *unknown* placeholder tokens (UNKs) as alternatives. Consequently, even by extending the overall capacity of the language model using language adapters, the model will not be able to adequately represent the respective languages. To extend the model to unseen scripts, we learn a new language-specific tokenizer and train a new embedding matrix, initialized with lexically overlapping tokens of the original embedding matrix, and random initialization for the remaining



(a) Pfeiffer adapter architecture used by us. Each adapter comprises of an down- and up-projection and is inserted after the feed-forward layer within each Transformer layer.

(b) Task adapters for QE with multiple language adapters for the multilingual input. The input parts are passed through the respective language adapter before the entire representation is passed to the task adapter.

(c) Extra monolingual embeddings for scripts and languages not included in the multilingual embeddings alongside the multilingual embeddings. The input embedding is chosen depending on the input language.

Figure 1: The architecture additions to the Transformer architecture: (a) Adapters; (b) Language and task adapters with multilingual input; (c) Extra monolingual embeddings alongside multilingual embeddings.

unseen tokens (Pfeiffer et al., 2021b). Here, language adapters are trained together with the new embedding matrix, while the pre-trained Transformer weights are frozen. Similar to standard LAs, these components are trained monolingually using the MLM objective on unlabeled textual data in the target language.

Task Adapters. For target task fine-tuning we stack task-specific adapters on top of the pre-trained LAs. For most tasks, sentences of only one language are passed through the model, while for QE the original sentence in the source language and the translation of the target language are simultaneously passed through the model. The tokens of the respective languages are thus passed through their respective LA. The subsequent task adapter is shared between the two languages (see Figure 1b). For cross-lingual transfer, the LAs of the training languages are replaced with the LAs of the evaluation languages. For this reason, not only the transformer weights but also the LAs are frozen during training and only the task adapters are fine-tuned on the target task. For languages with scripts not covered during pre-training, the new embedding matrix is used. The embedding representations are subsequently concatenated (see Figure 1c).

3 Data

The sentence-level direct assessment task of WMT21 builds upon the data of WMT20 task 1 (Fomicheva et al., 2020a). The WMT20 dataset consists of seven language pairs ranging from the high-resource English–German (En-De) and English–Chinese (En-Zh), to the medium-resource Romanian–English (Ro-En), Estonian–English (Et-En) and Russian-English (Ru-En), and the low-resource Sinhalese–English (Si-En) and Nepalese–English (Ne-En). For each pair, sentences in the source language are sampled from Wikipedia (or in the case of Russian, from Wikipedia and Reddit), translated with fairseq (Ott et al., 2019) to the target language, and then annotated by at least three professional translators with Direct Assessment (DA) (Guzmán et al., 2019). The DA scores are z-normalized for each annotator and averaged to form the final score. For each of the seven language pairs, the dataset contains 7000 training pairs and 1000 test and dev pairs.

The WMT21 dataset extends the WMT20 dataset by providing new test sets—with unpublished labels—consisting of 1000 sentences for each language pair of the WMT20 dataset. In addition, they provide testsets for four new language pairs for zero-shot evaluation, each compris-

ing of 1000 sentence pairs with unpublished labels: English–Czech (En-Cs), English–Japanese (En-Ja), Pashto–English (Ps-En), Khmer–English (Km-En).

4 Experiments

We describe our experimental setup along with the training and implementation details.

Training & Model Hyperparameters. We initialize our models with mBERT and XLM-R (both large and base-sized). We use a reduction factor r of 8 for our task adapters. Language adapters use $r = 2$ and have been trained on Wikipedia articles of the respective language. The additional embeddings for Khmer and Sinhala contain 10k tokens each and have been fine-tuned together with the respective LAs on the Wikipedia data.

We fine-tune our models using AdamW (Loshchilov and Hutter, 2019) with a linear learning rate schedule without warm-up. We simulate early stopping by storing the checkpoint with the best dev set performance—evaluating every 500 steps. For all models, we use a learning rate of 1e-4 and a batch size of 8. We train each model for 8k steps. Hyperparameters have been chosen based on the WMT20 dev set performance. We have chosen the above hyperparameters from the following values ranges: learning rate {5e-5, 1e-4, 2e-4, 5e-4}, batch size {4, 8, 16, 32, 96}, reduction factor r for the task adapters {4, 8, 16}, and training steps {2k, 3k, 5k, 8k, 10k}.

Implementation Details. To train adapters, we use the AdapterHub framework (Pfeiffer et al., 2020a) which builds upon the Hugging Face Transformers library (Wolf et al., 2020). In each batch we samples examples from only one language pair.

Experimental Setup. We evaluate the performance of our QE models using Pearson correlation between the predicted quality and the actual label (Specia et al., 2020).

We evaluate our adapter approaches in an ALL and a leave-one-out zero-shot setup (ZERO). In the ALL setup, we train a model on all seven language pairs with training data available and then evaluate the model on all eleven language pairs—the seven pairs with training data and the four pairs without. In the ZERO setting, for each of the seven language pairs which have a training set, we train a model with six of the pairs and then evaluate on the left-out seventh pair.

We evaluate both the large-sized XLM-R with adapters (denoted A-XLMR$_{\text{LARGE}}$) and base-sized mBERT and XLM-R with adapters (denoted A-MBERT and A-XLMR$_{\text{BASE}}$ respectively). For mBERT, we use both language adapters (+LA) and additional embeddings for Sinhala and Khmer (+EMB). We denote the setup with both as A+LA+EMB-MBERT. We also consider adapter ensembles for XLM-R. Here, we train five adapters in the ALL setup using different random seeds. During the evaluation, we average the predictions of the five adapters for the final prediction.

5 Results & Discussion

We present the Pearson correlation results for our models on the WMT21 test set. The reported values are obtained from the CodaLab competition.[1]

5.1 Language Extension Results

We present our results on the WMT21 test set for our two setups. The results for the ALL setup where we train with all seven pairs that have training data and then evaluate the model on all eleven pairs, i.e. the seven with training data and the four which are zero-shot, are found in Table 1. The leave-one-out ZERO results where we train on six of the seven pairs with training data and then evaluate in a zero-shot setup on the left-out pair are in Table 2.

We consider how our language extension methods improve the results for the unseen languages Sinhala, Khmer, and Pashto. We first evaluate how much we gain by representing input in the unseen script with extra embeddings instead of simply replacing all by the [UNK]-token. For this, we compare A-MBERT with A+EMB-MBERT. When we train with the Si-En data in ALL, the additional embeddings only give a relatively small performance boost of 0.04 points on top of already quite good results. This is unexpected since half the input is not correctly represented. We investigate this in more detail in §5.2. In zero-shot (Table 2 for Si-En and Table 1 for Km-En), the extra embeddings result in greatly improved results for Si-En by 0.25 points and by 0.05 points for Km-En.

Next, we compare models with and without language adapters in both setups. For the languages seen by mBERT during pre-training, there is little difference between A(+EMB)-MBERT and A+LA(+EMB)-MBERT in both setups. This

[1] https://competitions.codalab.org/competitions/33411

	Unseen			Seen							
	Si-En	***Km-En***	***Ps-En***	**Ne-En**	**Et-En**	**Ro-En**	**Ru-En**	**En-De**	**En-Zh**	***En-Cs***	***En-Ja***
A-MBERT	0.44	*0.37*	–	–	–	–	–	–	–	–	–
A+EMB-MBERT	0.48	*0.42*	*0.22*	0.73	0.68	0.84	0.63	0.36	0.50	*0.41*	*0.24*
A+LA+EMB-MBERT	0.51	*0.49*	*0.50*	0.74	0.68	0.84	0.64	0.33	0.48	*0.47*	*0.23*
A-XLMR$_{\text{BASE}}$	0.52	*0.57*	*0.53*	0.71	0.68	0.82	0.68	0.33	0.49	*0.45*	*0.27*
A-XLMR$_{\text{LARGE}}$	0.56	*0.62*	*0.59*	0.80	0.78	0.87	0.73	0.47	0.54	*0.54*	*0.33*
A-XLMR$_{\text{LARGE}_{\text{ENSEMBLE}}}$	0.57	*0.64*	*0.61*	0.83	0.79	0.89	0.76	0.43	0.56	*0.55*	*0.32*

Table 1: Pearson correlation results of the ALL setup for trained results for the seven pairs with training set and zero-shot results for the four pairs without. We group the language pairs in those unseen and seen by mBERT during pre-training and we mark the zero-shot results of the pairs without training set with *italic*. We report the results for our adapters with mBERT, XLM-R (base), and XLM-R (large). For mBERT, we extend the model with language adapters (+LA) and additional embeddings for Sinhala and Khmer (+EMB)

	Unseen	Seen					
	Si-En	**Ne-En**	**Et-En**	**Ro-En**	**Ru-En**	**En-De**	**En-Zh**
A-MBERT	0.03	–	–	–	–	–	–
A+EMB-MBERT	0.28	0.63	0.63	0.76	0.55	0.41	0.43
A+LA+EMB-MBERT	0.46	0.64	0.65	0.75	0.54	0.37	0.39
A+XLMR$_{\text{BASE}}$	0.51	0.67	0.63	0.68	0.56	0.33	0.38
A+XLMR$_{\text{LARGE}}$	0.55	0.79	0.75	0.81	0.66	0.43	0.56

Table 2: Pearson correlation results of the leave-one-out ZERO setup for zero-shot results of the seven language pairs with training set. We report the results for our adapters with mBERT and XLM-R (base & large). For mBERT, we extend the model with language adapters (+LA) and additional embeddings for Sinhala and Khmer (+EMB)

aligns with the findings by Pfeiffer et al. (2020b) and suggests that language adapters are less helpful for seen languages. For the three pairs with unseen languages, the language adapters can greatly improve the performance. In zero-shot situations (Table 2 for Si-En and Table 1 for the other two), we gain 0.18 points for Si-En, 0.07 for Km-En, and 0.28 points for Ps-En. Similar to extra embeddings, when we train with the Si-En data in ALL, we only gain 0.03 points more with language adapters.

Finally, we compare mBERT with language adapters and additional embeddings (A+LA+EMB-MBERT) to a base-sized XLM-R A-XLMR$_{\text{BASE}}$. This comparison is not ideal due to the differences in pre-training between the Transformers—training set, selected languages, etc.—but we can assume that for the unseen languages, XLM-R serves as an estimated upper bound for the performance. For seen language pairs (i.e., not Si-En, Km-En, and Ps-En), both methods perform comparably. For unseen languages, our adapter-based extensions to mBERT close the gap to XLM-R for most languages, except for Km-En where there is still a noticeable performance difference.

	Si-En	**Ne-En**	**Et-En**	**Ro-En**	**Ru-En**	**En-De**	**En-Zh**	**Avg**
A-MB$_{\text{S+T}}$	0.54	0.68	0.69	0.85	0.63	0.42	0.43	0.61
A-MB$_{\text{T}}$	0.52	0.52	0.61	0.70	0.58	0.40	0.38	0.53
A-MB$_{\text{S}}$	0.19	0.53	0.56	0.63	0.60	0.33	0.30	0.45

Table 3: Pearson correlation for mBERT with adapters (A-MB)—without language extensions—on the WMT20 test set trained with all pairs where we use both source and translation (S+T), only the translation (T), or only the source (S) during training. Evaluation is performed with both source and translation.

5.2 Analysis of Results on Trained Pairs

For the three unseen languages, we achieve large performance gains in zero-shot scenarios. However, while we witness large performance gains in zero-shot scenarios of the adapter-based methods, the difference considerably smaller when training data in the target language is available. Intuitively, we would expect a larger boost, considering half the input is in an unknown language and mostly not encoded. However, these results align with previous findings. Sun et al. (2020b) show for WMT19 and WMT18 that training with *only* the translation still results in strong results—77-100% of the per-

	Si-En	Ne-En	Et-En	Ro-En	Ru-En	En-De	En-Zh	Avg
A+LA+EMB-MBERT$_{ALL}$	0.59	0.69	0.71	0.85	0.65	0.44	0.43	0.62
BERGAMOT-LATTE (mBERT)	0.53	0.69	0.70	0.85	0.65	0.42	0.45	0.61
A-XLMRBASE$_{ALL}$	0.59	0.67	0.70	0.81	0.68	0.41	0.41	0.61
A-XLMRLARGE$_{ALL}$	0.65	0.75	0.78	0.88	0.75	0.48	0.46	0.68
A-XLMRLARGE$_{ENSEMBLE}$	0.66	0.79	0.80	0.89	0.77	0.47	0.47	0.69
TransQuest (XLM-R)	0.65	0.76	0.76	0.89	0.75	0.44	0.46	0.67
BERGAMOT-LATTE (XLM-R)	0.67	0.78	0.80	0.89	0.78	0.50	0.49	0.70
TransQuest (best)	0.68	0.82	0.82	0.91	0.81	0.55	0.54	0.72
BERGAMOT-LATTE (best)	0.68	0.81	0.83	0.91	0.80	0.54	0.53	0.72
A+LA+EMB-MBERT$_{ZERO}$	0.54	0.57	0.65	0.77	0.53	0.44	0.33	0.55
A+XLMRBASE$_{ZERO}$	0.56	0.61	0.63	0.67	0.59	0.35	0.32	0.53
A+XLMRLARGE$_{ZERO}$	0.63	0.74	0.76	0.80	0.69	0.41	0.41	0.63
BERGAMOT-LATTE (zero-shot)	0.68	0.76	0.75	0.80	0.68	0.45	0.42	0.65

Table 4: Pearson correlation on the WMT20 test set for the ALL and ZERO setup. We group the results in the setups in base-sized and large models. TransQuest and BERGAMOT-LATTE use fully fine-tuned models. TransQuest results are taken from (Ranasinghe et al., 2020b), BERGAMOT-LATTE from (Sun et al., 2020a)—their best models are the winners of the WMT20 shared task and additionally use ensembles.

formance of training with the complete pair. We are able to reproduce these findings for WMT20 in Table 3, and achieve similar results for Si-En when passing *only* the English translation as input to the model, compared to when training on both inputs. However, when training with only the (Sinhala) source, we witness the expected drop in performance. It is likely that in the zero-shot setup, the model cannot learn to exploit the statistical cues that allow it to function without the source sentence. Hence, we obtain more appropriate representations with adapter-based methods where the language-specific word-embedding representations result in considerable performance gains.

5.3 Ensembles

Ensembles have been used in previous work with great success (Ranasinghe et al., 2020a; Fomicheva et al., 2020b; Nakamachi et al., 2020). With an adapter ensemble, the underlying Transformer weights are re-used resulting in a very parameter-efficient setup—our ensemble with five adapters adds only 6.5% more parameters on top of the large XLM-R Transformer. However, our adapter ensemble A-XLMRLARGE$_{ENSEMBLE}$ only brings a slight performance boost, smaller than the reported boost by the ensembles of previous works. More work is needed here to investigate why this is the case.

5.4 Comparison to Fully Fine-Tuned Models

We evaluate the general performance of adapters for the QE task in comparison to fully fine-tuned models. For this, we compare our models on the WMT20 test set against the top submissions of the WMT20 shared task in Table 4. We find that they achieve competitive results with fully fine-tuned models that do not employ additional techniques like ensembles in both the ALL and ZERO setups. Our highest-scoring submission, A-XLMRLARGE$_{ENSEMBLE}$, places in the midfield for the WMT21 competition.

5.5 Parameter Count

Adapters are considerably more parameter efficient with respect to the number of *fine-tuned* parameters, compared to fully fine-tuned models. The number of adapter parameters is equivalent to only 1.3% of the Transformer parameters for our models. This makes adapters very lightweight for model sharing or for loading multiple adapters on the same GPU, e.g., for language adapters or for multiple task adapters in a pipeline (Nguyen et al., 2021; Rücklé et al., 2021). The extension for the unseen languages for mBERT also adds only a small number of parameters: 2.4% for each language adapter and 1.4% for each monolingual embedding.

6 Conclusion

In this work, we proposed the use of adapters to fine-tune massively multilingual Transformers for the sentence-level QE task. We demonstrated that adapters are able to achieve competitive results with fully fine-tuned models. However, as fully fine-tuned approaches are limited to the languages seen during pre-training, we have employed recent language extension methods to integrate languages

unseen by mBERT. We extended mBERT with language adapters and monolingual embeddings for Sinhala, Khmer, and Pashto. These methods greatly improved the zero-shot performance of the model and largely closed the gap to XLM-R which has been pre-trained on all languages appearing in WMT21. This demonstrates that our approach is applicable, not only to languages seen during pre-training, but also to unseen languages, even with unseen scripts. This suggests that our method is able to extend multilingual models to a wider range of language not covered during pre-training.

We suggest that future shared tasks should consider disentangling languages which massively multilingual language models have been pre-trained on, from those that are unseen during pre-training, to more closely reflect realistic scenarios, as the majority of languages cannot be represented within a single model (Conneau et al., 2020).

Acknowledgements

This work has been partly supported by the German Research Foundation as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) at the Technische Universität Darmstadt under grant No. GRK 1994/1. Jonas Pfeiffer is supported by the LOEWE initiative (Hesse, Germany) within the emergenCITY center.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Erick R. Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. MLQE-PE: A multilingual quality estimation and post-editing dataset. *arXiv preprint*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020b. BERGAMOT-LATTE submissions for the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 1010–1017. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020c. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6097–6110. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6282–6293. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos V. Treviso, Miguel Vera, and André F. T. Martins. 2019. Openkiwi: An open source framework for quality estimation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 117–122. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Akifumi Nakamachi, Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. TMUOU submission for WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 1037–1041. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021*, pages 80–90. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 487–503. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 46–54. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7654–7673. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. UNKs Everywhere: Adapting Multilingual Language Models to New Scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Online, November , 2021*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. Transquest at WMT2020: sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 1049–1055. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5070–5081. International Committee on Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the Efficiency of Adapters in Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Online, November , 2021*.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021, Online, August 1-6, 2021*. Association for Computational Linguistics.

Yurun Song, Junchen Zhao, and Lucia Specia. 2021. Sentsim: Crosslingual semantic evaluation of machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3143–3156. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Rocha Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 743–764. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shuo Sun, Marina Fomicheva, Frédéric Blain, Vishrav Chaudhary, Ahmed El-Kishky, Adithya Renduchintala, Francisco Guzmán, and Lucia Specia. 2020a. An exploratory study on multilingual quality estimation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing,*

AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020, pages 366–377. Association for Computational Linguistics.

Shuo Sun, Francisco Guzmán, and Lucia Specia. 2020b. Are we estimating or guesstimating translation quality? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6262–6267. 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. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2021. Inducing language-agnostic multilingual representations. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 229–240, Online. Association for Computational Linguistics.

Wei Zhao, Goran Glavas, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1656–1671. Association for Computational Linguistics.

Quality Estimation Using Dual Encoders with Transfer Learning

Dam Heo* **WonKee Lee*** **Baikjin Jung*** **Jong-Hyeok Lee*,†**
*Departmet of Computer Science and Engineering,
†Graduate School of Artificial Intelligence,
Pohang University of Science and Technology (POSTECH), Republic of Korea
{dammy, wklee, bjjung, jhlee}@postech.ac.kr

Abstract

This paper describes POSTECH's quality estimation systems submitted to Task 2 of the WMT 2021 quality estimation shared task: Word and Sentence-Level Post-editing Effort. We aim to improve the stability of recently proposed quality estimation models, which usually have a single encoder based on the self-attention mechanism to simultaneously process both of the two input data: a source sequence and its machine translation; considering that such models are not propped up by pre-trained language models' monolingual word representations, which are generally accepted as reliable representations for various natural language processing tasks. Therefore, our model first uses two pre-trained monolingual encoders and then exchanges their output information through two additional cross attention networks. According to the official leaderboard, our systems outperform the baseline systems in terms of the Matthews correlation coefficient for machine translations' word-level quality estimation and in terms of the Pearson's correlation coefficient for sentence-level quality estimation by 0.4126 and 0.5497 respectively.

1 Introduction

Quality estimation (QE) is the task of estimating the quality of given machine translations without regard to their reference translations (Blatz et al., 2004; Specia et al., 2009). As reference translations are generally unavailable in real life, QE should help to treat output texts of machine translation (MT) systems. QE can be categorized into several subtasks, and this round of the WMT QE task has three subtasks, yet we focus on Task 2: Word and Sentence-Level Post-editing Effort. In Task 2, while sentence-level QE aims to predict the Human-Targeted Translation Edit Rate (HTER, Snover et al. 2006), which measures the edit distance between an MT output (*mt*) and its human post-edited text (*pe*), word-level QE aims to predict OK–BAD tags for three sequences of tokens: the sequence of words in a source text (*src*) depending on whether they are correctly translated referring to *mt*; the sequence of words in *mt* depending on their correctness; and <GAP> tokens, which each represent the gap between two adjacent words, depending on the existence of any missing words (Specia et al., 2020).

As other recent QE models do, our method also applies transfer learning, considering that pre-trained language models (LM) have been successfully applied to various natural language processing (NLP) tasks including QE; many previous studies (Fomicheva et al., 2020; Hu et al., 2020; Wu et al., 2020; Lee, 2020; Moura et al., 2020; Nakamachi et al., 2020; Rubino, 2020) that apply pre-trained LMs to QE have adopted multilingual or cross-lingual LMs such as multilingual-BERT (Pires et al., 2019), XLM (Conneau and Lample, 2019), and XLM-R (Conneau et al., 2020) to process the two input data *src* and *mt*. Such cross-lingual LMs have a single Transformer (Vaswani et al., 2017) encoder using only the self-attention mechanism to create vector representations of the input data and predict the labels. However, it appears possible to further improve the stability of those models, considering that they are not propped up by pre-trained LMs' monolingual word representations, which are generally accepted as reliable representations for various NLP tasks.

With this background, we propose a QE model that has two separate pre-trained encoders that each produce monolingual representations of *src* and *mt*, respectively. On top of each encoder, we add a cross attention network for the learning of the cross-lingual context between *src* and *mt*; these networks will produce two sets of cross-lingual representations for QE. We conduct simple experiments to compare the performance of our systems and ensembles of them with that of the baseline systems and that of other submitted systems for

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 920–927
November 10–11, 2021.

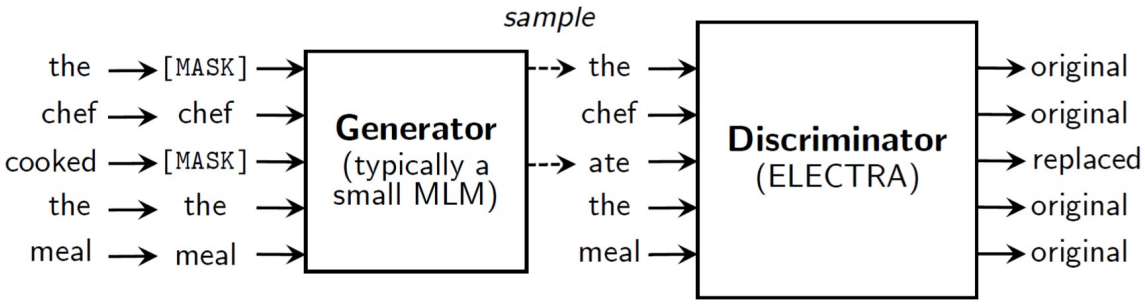


Figure 1: A diagram depicting the training task of ELECTRA (Clark et al., 2020)

Task 2. Experimental results imply that although our systems do not always outperform the baseline systems, they do in terms of the Matthews correlation coefficient (MCC) for *mt*'s word-level QE and in terms of the Pearson's correlation coefficient (PCC) for sentence-level QE by 0.4126 and 0.5497 respectively.

2 Related Work

Because our model does not confine its monolingual encoders to specific pre-trained LMs, all pre-trained LMs can be considered relevant. Among them, most of the recently proposed pre-trained LMs are denoising autoencoders, of which the pre-training task is usually to select about 15% of tokens in unlabeled input sequences and apply the attention mechanism to those tokens (Yang et al., 2019) or is to mask certain tokens (Devlin et al., 2019) and then restore them. However, in our experiments, our systems use ELECTRA (Clark et al., 2020). ELECTRA introduces "replaced token detection" as an additional pre-training task and let the language model learn to distinguish between real input tokens and specious but artificially generated tokens. In detail, when the generator network predicts the tokens in the masked positions, some of the predicted tokens are corrupted, and then this output sequence is fed into a Transformer-based discriminator network, which predicts whether each token in the fed sequence is the same as the original one or is a replaced one (Figure 1). We suppose that this process and QE are similar to each other in that both of them predict the soundness of the given tokens, so ELECTRA would be one of the most appropriate pre-trained LMs for our QE model's monolingual encoders, especially for Task 2.

3 Model Description

Our model uses two ELECTRAs: one ELECTRA[1] that is pre-trained with English corpora and the other ELECTRA[2] pre-trained with German corpora. Figure 2 depicts the overall structure of our model.

3.1 Dual Monolingual Encoders

Our model has dual encoders: a pre-trained English ELECTRA processing *src* and a pre-trained German ELECTRA processing *mt*. These encoders will produce reliable monolingual representations of *src* and *mt* respectively to provide these refined representations to the upper cross attention networks.

Because unlike other pre-trained QE models that have a single Transformer encoder being fed with the concatenation of *src* and *mt*, our model lets the two different encoders process the two input data respectively, we exclude the segment embeddings, which are used to distinguish one language from another, and assign different positional embeddings to each input data. In addition, for sentence-level QE, *mt*'s special token `<CLS>` is used to predict the HTER.

3.2 Cross Attention Networks

We attach a cross attention network to each pre-trained encoder; it learns the cross-lingual context information by using the encoders' refined monolingual representations of the two input data. Although the structure of a cross attention network is identical to that of the encoders, the cross attention networks are not pre-trained, so we train them after the random initialization of their parameters. We

[1] https://huggingface.co/google/electra-base-discriminator

[2] https://huggingface.co/german-nlp-group/electra-base-german-uncased

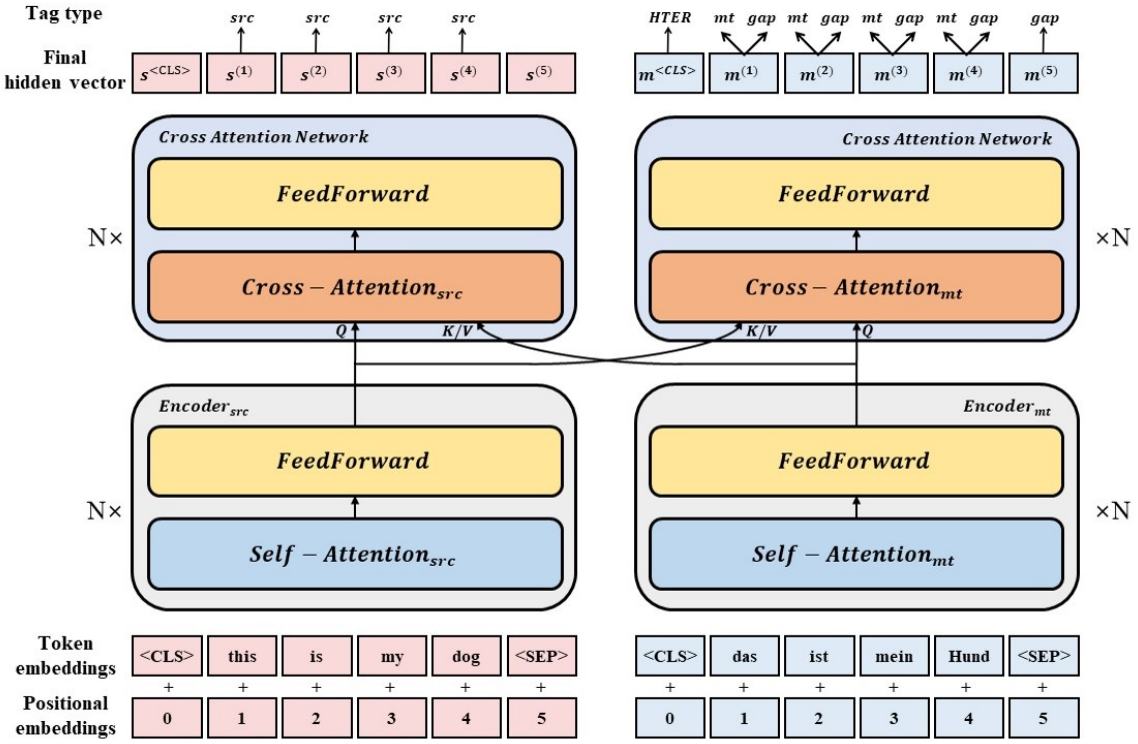


Figure 2: The overall structure of our model

find that applying transfer learning to the cross attention networks is not available due to the absence of pre-trained language models that are pre-trained to perform cross attention on cross-lingual input data by using one side as a query vector and the other side as both a key vector and a value vector just as the Transformer decoder performs multi-head attention on the output of the Transformer encoder.

3.2.1 Sentence-Level QE

To predict the HTER for sentence-level QE we employ the final hidden vector $m^{\texttt{<CLS>}}$ of the *mt*-side cross attention network, which is the final representation of the `<CLS>` token, as the representation of the *mt* sequence as a whole. After this representation passes through double linear layers with the GELU (Hendrycks and Gimpel, 2016) activation function, the HTER of the given *mt* sentence is estimated as follows.

$$\mathbf{l} = \mathbf{W}_h m^{\texttt{<CLS>}} + \mathbf{b}_0$$
$$\hat{y}_{\text{HTER}} = \mathbf{w}_h^T \text{GELU}(\mathbf{l}) + b_1 \quad (1)$$

We have trainable parameters $\mathbf{W}_h \in \mathbb{R}^{H \times H}$, $\mathbf{w}_h \in \mathbb{R}^H$, $\mathbf{b}_0 \in \mathbb{R}^H$, and $b_1 \in \mathbb{R}$; H denotes hidden vectors' dimension.

We use the mean squared error of this estimator, that is, the difference between the estimated HTER $\hat{y}_{\text{HTER}}$ and the ground truth HTER value y_{HTER}, as the training loss

$$\mathcal{L}_{\text{HTER}} = \text{MSE}(\hat{y}_{\text{HTER}}, y_{\text{HTER}}). \quad (2)$$

3.2.2 Word-Level QE

***src*-Side Prediction** We use the final hidden vector $s^{(i)}$ ($i \in \{1, ..., |S|\}$, where $|S|$ is the number of tokens in the tokenized *src* sequence) of the *src*-side cross attention network corresponding to each token in *src* to predict OK or BAD in the token's position. After each of these representations passes through a linear layer, the word-level probability of the corresponding token being OK or BAD is predicted with a sigmoid activation function:

$$P_s^{(i)} = \text{sigmoid}(\mathbf{w}_s^T s^{(i)}), \quad (3)$$

where $\mathbf{w}_s \in \mathbb{R}^H$ is a trainable parameter.

We use the binary cross-entropy loss function; we also introduce an extra hyperparameter k_s to prevent our model from being overfitted because the statistics of the ratio between the number of OK tags and that of BAD tags in our training data (Table 1) can misguide the model to have the tendency

		# of words	OK	BAD
Artificial data	*src*	54.6M	34.3M (62.81%)	20.3M (37.19%)
	mt.word	50.5M	29.7M (58.82%)	20.8M (41.18%)
	mt.gap	53.5M	50.6M (94.53%)	2.9M (5.47%)
WMT 21 train	*src*	115K	84K (73.05%)	31K (26.95%)
	mt.word	112K	81K (71.85%)	32K (28.15%)
	mt.gap	119K	114K (95.41%)	5K (4.59%)
WMT 21 dev	*src*	16K	12K (74.21%)	4K (25.79%)
	mt.word	16K	12K (72.49%)	4K (17.51%)
	mt.gap	17K	16K (95.83%)	0.7K (4.17%)

Table 1: Statistics of QE datasets used in our experiments.

of outputting OK even when it should output BAD. The *src*-side loss is as follows:

$$\mathcal{L}_{src} = \frac{1}{|S|}\sum_{i=1}^{|S|}\Big\{k_s y_s^{(i)} log P_s^{(i)} + (1-y_s^{(i)}) log(1-P_s^{(i)})\Big\}, \quad (4)$$

where $y_s^{(i)}$ is a ground-truth OK–BAD tag.

***mt*-Side Prediction** We use the final hidden vector $m^{(i)}$ ($i \in \{1, ..., |M|\}$, where $|M|$ is the number of tokens in the tokenized *mt* sequence) of the *mt*-side cross attention network corresponding to each token in *mt* to predict OK or BAD in the token's position. We estimate the probabilities of the word tokens

$$P_m^{(i)} = \text{sigmoid}(\mathbf{w}_m^T m^{(i)}), \quad (5)$$

where $\mathbf{w}_m \in \mathbb{R}^H$ is a trainable parameter.

We also use the final hidden vector $m^{(j)}$ ($j \in \{1, ..., |M|+1\}$ including the vector in the position of the last <SEP> token to predict OK or BAD for the last <GAP> token. We estimate the probabilities of the <GAP> tokens

$$P_g^{(j)} = \text{sigmoid}(\mathbf{w}_g^T m^{(j)}), \quad (6)$$

where $\mathbf{w}_g \in \mathbb{R}^H$ is a trainable parameter.

The *mt*-side prediction loss equals the sum of the losses for word tokens and <GAP> tokens:

$$\mathcal{L}_{mt} = \mathcal{L}_m + \mathcal{L}_g, \quad (7)$$

where

$$\mathcal{L}_m = \frac{1}{|M|}\sum_{i=1}^{|M|}\Big\{k_m y_m^{(i)} log P_m^{(i)} + (1-y_m^{(i)}) log(1-P_m^{(i)})\Big\}, \quad (8)$$

and

$$\mathcal{L}_g = \frac{1}{(|M|+1)}\sum_{j=1}^{|M|+1}\Big\{k_g y_g^{(j)} log P_g^{(j)} + (1-y_g^{(j)}) log(1-P_g^{(j)})\Big\}, \quad (9)$$

$y_m^{(i)}$ and $y_g^{(j)}$ being ground-truth OK–BAD tags for a word token and a <GAP> token respectively and hyperparameters k_m and k_g being introduced for the same reason why we introduce k_s.

Finally, we define the word-level loss and the overall QE loss of our model as follows.

$$\mathcal{L}_{\text{word}} = \mathcal{L}_{src} + \mathcal{L}_{mt} \quad (10)$$

$$\mathcal{L}_{\text{QE}} = \mathcal{L}_{\text{word}} + \mathcal{L}_{\text{HTER}} \quad (11)$$

4 Experiments

4.1 Datasets

In our experiments, we used the eSCAPE (Negri et al., 2018) dataset, which is a collection of data triplets each of which is composed of *src*, *mt*, and *pe*; we used this dataset to make artificial QE training data. In this process, to make our artificial data have a similar statistics as those of WMT 2021's official training data, we filtered eSCAPE triplets according to various criteria, such as the sequence lengths of *src* and *mt*, the sequence length ratio

Encoder	PCC↑	*src*-side MCC↑	*mt*-side	
			Words MCC↑	<GAP>s MCC↑
BERT	0.4832	0.3092	0.3684	0.03617
ELECTRA	0.5109	0.3100	0.4104	0.1401

Table 2: Single Dual-Encoder model's performance respect to pre-trained model applied to its encoders for the WMT 2020 English-German QE task2.

Systems	PCC↑	*src*-side MCC↑	*mt*-side	
			Words MCC↑	<GAP>s MCC↑
Baseline	0.5285	0.3220	0.3696	0.1157
Single	0.5038	0.3200	0.4126	0.1096
Top4-ens	0.5458	0.3165	0.4296	0.1096
Top6-ens	0.5497	0.3186	0.4271	0.1225

Table 3: Our systems' performance for the WMT 2021 English–German QE Task 2. Single is a single system modelled on our proposed model. Top4-ens and Top6-ens are the ensembles of the top four and the top six single systems respectively in terms of their performance on the validation dataset.

between *src* and *mt*, and TER. Then, we created a tuple of labels (T_{src}, T^{word}_{mt}, T^{gap}_{mt}, the TER (Snover et al., 2006)) for each triplet [3]. Finally, we tokenized and truncated both of the artifical data and the WMT 2021 official data by using a pre-trained tokenizer based on WordPiece (Wu et al., 2016).

4.2 QE Pre-training

After obtaining about three million of artificial triplets, we made the final QE pre-training data by joining the artificial training data and the official human-labeled data together; especially, we augmented the quantity of the latter by replication to allow our systems to learn from both kinds of training data relatively more evenly during the QE pre-training. Our systems learn to predict all kinds of labels jointly (L_{QE}, Eqn. 11) considering the close correlation among the subtasks in Task 2. We used 1,000 triplets in the WMT 2021's official development dataset as validation data.

4.3 Fine-Tuning

We used only the WMT 2021 human-labeled data for fine-tuning. In contrast with the QE pre-training, we fine-tuned our systems to each subtask: the prediction of the sentence-level task (L_{HTER}, Eqn. 2) and the word-level task (L_{word}, Eqn. 10) Considering the overproportion of OK tags in our training data (Table 1), we set a large k_s, k_m, and k_g (§ 3.2.2) in our experiments.

4.4 Ensemble Learning

Besides single fine-tuned systems, we also made ensembles of our best fine-tuned systems, each of which has a different random seed from that of the others. In detail, after fine-tuning several single systems with different random seeds, for each seed, we picked out the top two systems, each of which is different from the other in certain variable training conditions such as how its cross attention networks have been randomly initialized in that instance, in terms of their performance on our validation dataset. Finally, we averaged the weights of the systems element-wisely for better generalization and made the ensembles.

4.5 Hyperparameters

We used ELECTRA-base (Clark et al., 2020)s as pre-trained monolingual LMs for our dual monolingual encoders[4]. In the QE pre-training, we used `get_schedule_with_warmup`[5] as our learning rate scheduler with 3,000 warm-up steps. We used the AdamW (Loshchilov and Hutter, 2018) optimizer that has a weight decay with λ=0.5, β_1=0.9, β_2=0.999, and ϵ=1e-8, together with gradient clipping. Setting a batch size of 64 for both the QE pre-training and fine-tuning, we set a learning rate of $1e{-}5$ and a tuple of ($k_s = 1$, $k_m = 1$, $k_g = 3$) for the QE pre-training and a learning rate of $5e{-}5$ and a tuple of ($k_s = 2$, $k_m = 2$, $k_g = 4$) for the

[3] https://github.com/deep-spin/qe-corpus-builder

[4] Our encoders are available at https://huggingface.co/models

[5] https://huggingface.co/transformers/main_classes/optimizer_schedules.html

Systems	PCC↑	RMSE↑	MAE↑	Disk Footprint (GB)↓	Model Params↓
HW-TSC	0.6531	0.1513	0.1079	2.0898	560.9M
IST-Unbabel	0.6173	0.1715	0.1163	2.1373	569.4M
ACBU-NMT	0.5773	0.1743	0.1154	2.0894	560.1M
POSETCH (Ours)	0.5497	0.1741	0.1304	1.4540	390.2M
Baseline	0.5285	0.1828	0.1291	1.0640	281.3M
ENSBRT	0.5199	0.1711	0.1287	1.2700	502M

Table 4: The reported sentence-level QE performance of the systems submitted to the WMT 2021 English–German QE Task 2 according to the official leaderboard.

Systems	*mt*-side		*src*-side MCC	Disk Footprint (GB)↓	Model Params↓
	Words MCC	<GAP>s MCC			
JHU-Microsoft	0.5231	0.2559	-	6.3918	484.4M
HW-TSC	0.5095	0.2997	0.4499	2.0898	560.9M
IST-Unbabel	0.4661	0.1833	0.4042	2.1373	569.4M
ACBU-NMT	0.4368	-	0.3915	2.0894	560.1M
POSETCH (Ours)	0.4126	0.1096	0.3200	1.4540	390.2M
Baseline	0.3696	0.1157	0.3220	1.0640	281.3M

Table 5: The reported word-level QE performance of the systems submitted to the WMT 2021 English–German QE Task 2 according to the official leaderboard. A hyphen indicates that no corresponding score exists.

fine-tuning, respectively. We validated the performance of our systems on our validation set every 5,000 steps during the QE pre-training and every 200 steps during the fine-tuning, respectively; we applied early stopping with a patience of 30.

4.6 Results

In comparison with our single system, our ensembles report an improved PCC, *mt*-side words MCC, and *mt*-side <GAP>s MCC of about 0.5497, 0.4296, and 0.1225 respectively (Table 2). Compared to other systems submitted to the WMT 2021 English–German QE Task 2, our systems outperform the baseline systems in terms of the sentence-level PCC (Table 3) and the *mt*-side words MCC (Table 4). Our systems are inferior to the baseline systems in terms of the *src*-side MCC and the *mt*-side <GAP>s MCC by a narrow margin (Table 4). However, because our systems have a smaller number of parameters than other submitted systems, we expect that it is possible to improve the performance of our systems by adopting larger pre-trained LMs such as ELECTRA-large (Clark et al., 2020).

5 Conclusion

We model our systems submitted to Task 2 of the WMT 2021 QE shared task on our proposed model, which uses dual pre-trained monolingual encoders and two additional cross attention networks to process the two input data *src* and *mt* more effectively considering that the latest Transformer-based QE models are not propped up by pre-trained monolingual word representations. We expect that the cross attention networks enable the two pre-trained monolingual encoders to exchange cross-lingual information without losing their stability and to learn the subtasks of Task 2 jointly and also separately. Experimental results partially supports this expectation: according to the official leaderboard, our systems outperform the baseline systems in terms of the *mt*-side words MCC and the sentence-level PCC by 0.4126 and 0.5497 respectively, although they do not in terms of the *src*-side MCC and the *mt*-side <GAP>s MCC. Neverhteless, it appears possible to improve the performance of our systems by adopting larger pre-trained LMs, and thus, our future work will explore such aspects and other related new methods.

Acknowledgements

This work was carried out as part of the HPC Support Project supported by the Ministry of Science and ICT (MSIT) and the National IT Industry Promotion Agency (NIPA), and was funded by the Institute of Information & Communications Technology Planning & Evaluation (IITP) supported by the Korean government (MSIT): Grant No. 2019-0-01906, Graduate School of Artificial Intelligence

(POSTECH).

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, pages 315–321.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020. Bergamot-latte submissions for the wmt20 quality estimation shared task. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Chi Hu, Hui Liu, Kai Feng, Chen Xu, Nuo Xu, Zefan Zhou, Shiqin Yan, Yingfeng Luo, Chenglong Wang, Xia Meng, et al. 2020. The niutrans system for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1018–1023.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Joao Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André FT Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036.

Akifumi Nakamachi, Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. Tmuou submission for wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1037–1041.

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)*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Raphael Rubino. 2020. Nict kyoto submission for the wmt'20 quality estimation task: Intermediate training for domain and task adaptation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1042–1048.

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: Technical Papers*, pages 223–231.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. Association for Computational Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *EAMT*, volume 9, pages 28–35.

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.

Haijiang Wu, Zixuan Wang, Qingsong Ma, Xinjie Wen, Ruichen Wang, Xiaoli Wang, Yulin Zhang, Zhipeng Yao, and Siyao Peng. 2020. Tencent submission for wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1062–1067.

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*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

ICL's Submission to the WMT21 Critical Error Detection Shared Task

Genze Jiang **Zhenhao Li** **Lucia Specia**
Language and Multimodal AI (LAMA) Lab, Imperial College London, UK
{genze.jiang20, zhenhao.li18, l.specia}@imperial.ac.uk

Abstract

This paper presents Imperial College London's submissions to the WMT21 Quality Estimation (QE) Shared Task 3: Critical Error Detection. Our approach builds on cross-lingual pre-trained representations in a sequence classification model. We improve the base classifier by (i) adding a weighted sampler to deal with imbalanced data and (ii) introducing feature engineering, where features related to toxicity, named-entities and sentiment, which are potentially indicative of critical errors, are extracted using existing tools and integrated to the model in different ways. We train models with one type of feature at a time and ensemble those models that improve over the base classifier on the development set. Our official submissions achieve very competitive results, ranking second for three out of four language pairs.

1 Introduction

Critical Error Detection (CED) is a new task which has been introduced in the WMT21 Quality Estimation (QE) Shared Task.[1] The purpose of CED is to address a challenging problem in Machine Translation (MT): translations produced by state-of-the-art MT systems can be grammatical and fluent but do not always retain the meaning of the source text. More importantly, incorrect translations can be misleading and even have catastrophic consequences such as health, safety, legal, or financial implications. However, these can be hard errors to capture by general QE architectures, which have been shown to be prone towards relying mainly on the translated sentence (Sun et al., 2020).

According to the Shared Task definition, a critical translation error is a type of error that occurs when the meaning of the translation deviates from source sentence in a critical way. The task data (Section 2.1) includes five categories of such errors: deviation in toxicity (TOX), in named entities (NAM), in sentiment polarity or negation (SEN), or in numbers (NUM), or introduction of health or safety risks (SAF).

The baseline model for this task utilises the XLM-RoBERTa (Conneau et al., 2020) for sequence classification model, following the Mono-TransQuest architecture proposed by Ranasinghe et al. (2020). Inspired by the fact that these five critical error types refer to specific linguistic phenomena, we aim to bring additional information to the models on the presence of such phenomena. The intuition is that sentences containing certain types of linguistic features, such as named entities or dates, are more likely to lead to errors. Therefore, we first process the dataset to extract features reflecting the sentences' toxicity, sentiment and named entities, using off-the-shelf toolkits or APIs (Section 2.2). We then enhance the baseline architecture with this additional information.[2]

We experiment with two approaches to take the additional features into account, at token and hidden state levels. We build multiple models taking one type of feature at a time and finally ensemble "promising" models. Promising models are those that lead to improvements over the baseline on the dev set (Section 2.3).

Our results comparing different features show that some of the features are indeed useful, but there is no general pattern that applies to all language pairs (Section 3.1). The official submission, which uses an ensemble of the models that lead to improvements over the baseline on the dev set for each language shows that ensembling only models with promising features are better than ensembling models with all kinds of features (Section 3.2). Upon manual inspection, we observed that additional features indeed help the model to make predictions but this is subject to the accuracy of features (Section 3.3).

[1] http://statmt.org/wmt21/quality-estimation-task.html

[2] Our code and data are available from https://github.com/conanjgz/critical-error-detection-for-MT

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 928–934
November 10–11, 2021.

2 Experiment Settings

2.1 Dataset

According to the description of WMT21 CED Shared Task, the dataset for this task was collected from Wikipedia comments (Wulczyn et al., 2017) in English with translations generated by the ML50 multilingual translation model (Tang et al., 2020), consisting of four language pairs: English-Czech (En-Cs), English-German (En-De), English-Japanese (En-Ja) and English-Chinese (En-Zh). The number of data samples in the training set differs for the four language pairs but is around 6500-8000. Each language pair has 1000 data samples in the dev set and 1000 data samples in the test set. For each sentence pair in the dataset, there are three labels given by three human annotators. The three labels are aggregated to the final label of the dataset using majority strategy. The final label is either `ERR` or `NOT`, where `ERR` means the translation has at least one critical error and `NOT` means the translation does not have a critical error.

Pair	Dataset	Count	Label	Count
En-Cs	Train set	7476	NOT	6188
			ERR	1288
	Dev set	1000	NOT	840
			ERR	160
	Test set	1000	–	–
En-De	Train set	7878	NOT	5674
			ERR	2204
	Dev set	1000	NOT	719
			ERR	281
	Test set	1000	–	–
En-Ja	Train set	7658	NOT	6939
			ERR	719
	Dev set	1000	NOT	904
			ERR	96
	Test set	1000	–	–
En-Zh	Train set	6859	NOT	5749
			ERR	1110
	Dev set	1000	NOT	859
			ERR	141
	Test set	1000	–	–

Table 1: Statistics of datasets for four language pairs. The distribution of labels for the test set is unknown as this is a blind evaluation task.

The dataset information for each language pair can be found in Table 1. As can be seen, the data is very imbalanced, with the En-Ja dataset suffering the most: the `ERR` label only accounts for 9.4% in the training set. The En-De dataset is the least imbalanced compared to other three language pairs, where the proportion of `ERR` label in En-De training set reaches 27.9%.

2.2 Features

We extract features reflecting sentences' toxicity score, sentiment and named entities. The expectation is that these features could be helpful in detecting critical errors since these errors stem from issues with the translation/introduction of these and other linguistic phenomena. Ideally we would have wanted to extract this information for both source and translated sentences to be able to perform some sort of comparison between the two, for example, presence of toxicity in the translation but not in the source sentence. However, we are limited by the availability of tools in the four language pairs, as we explain below.

For all features, our goal is to have a discrete representation which will allow us to easily incorporate them to the architecture, as will be explained in Section 2.3.2. Therefore, we need to threshold some of these features.

The toxicity score is produced by Perspective API,[3] which supports only English and German amongst our five languages. Based on some manual inspection of the predictions by Perspective, we consider that if the toxicity score of a sentence is greater than 0.5, the sentence will be regarded as toxic. We leave for future work experiments varying this threshold. Since this API does not support Czech, Japanese and Chinese, we were only able to extract a toxicity feature in the source sentences for En-Cs, En-Ja and En-Zh.

The sentiment score is produced by Google Cloud Natural Language API,[4] which supports English, German, Japanese and Chinese. Therefore, we can get the sentiment feature of both source sentence and translation for En-De, En-Ja and En-Zh. The score returned by this API is a float number ranged from -1 to 1. Empirically, we consider a sentence to be negative if the score is smaller than -0.2, and positive if the score is greater than 0.2, otherwise the sentence's sentiment is neutral. In our experiments, the sentiment feature is not applied to En-Cs because Czech is not supported by this API.

The information of named entities (NE) is extracted using spaCy,[5] which can recognise over 15 NE types. We count the number of named entities for each NE type and finally choose seven NE types with the highest counts as features. The description

[3] https://www.perspectiveapi.com/
[4] https://cloud.google.com/natural-language
[5] https://spacy.io/

of the seven NE types can be found in Table 2. We extract named entities in both source sentence and translation for En-De, En-Ja and En-Zh. However, Czech is not supported by spaCy, therefore we do not use NE features for En-Cs.

Type	Description	Abbr.
ORG	Organisation name	ORG
PERSON	Person name	PER
DATE	Year, month or day	DAT
CARDINAL	Numerals	CRD
ORDINAL	Ordinal numerals	ORD
NORP	Religious group, etc	NRP
GPE	Geographical name	GPE

Table 2: Descriptions of seven types of NE features and their abbreviations.

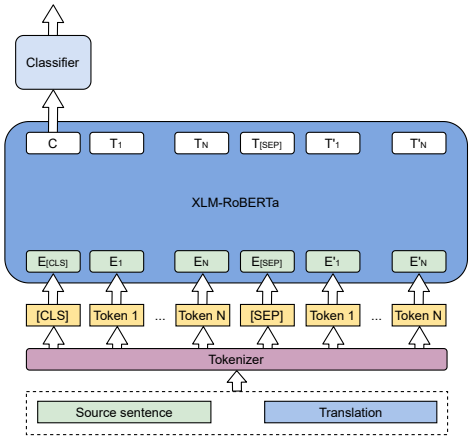


Figure 1: Architecture of baseline model. This is a MonoTransQuest model where we pass the output of the `[CLS]` token to a classifier.

2.3 Models

2.3.1 Baseline Model

The baseline model employs the MonoTransQuest framework (Ranasinghe et al., 2020), which is proposed for general quality estimation (QE) tasks and is shown in Figure 1. Essentially this is used to produce the baseline score of CED Shared Task. The model is based on a pre-trained XLM-RoBERTa transformer model (Conneau et al., 2020) and is used to perform sentence-level classification tasks. The model takes a sequence of tokens as input which starts with `<s>`, denoting `[CLS]` token, followed by tokens for source sentence and translation and ended with `</s>` token. The source sentence and its translation, separated by `[SEP]` token, are fed into one single transformer encoder at the same time. Then the output of the transformer encoder is fed into a classification head where cross-entropy is adopted as the loss function. We use pre-trained XLM-RoBERTa models released by HuggingFace's model repository (Wolf et al., 2020) for the implementation.

To alleviate the influence of imbalanced training data, a weighted sampler can be applied to the data loader during training. The weighted sampler is to make the label distribution in the training batch as balanced as possible. The weight of the sampler is computed as reciprocals of label proportions.

2.3.2 Model with Features

To utilise the features mentioned in Section 2.2, we proposed two different approaches.

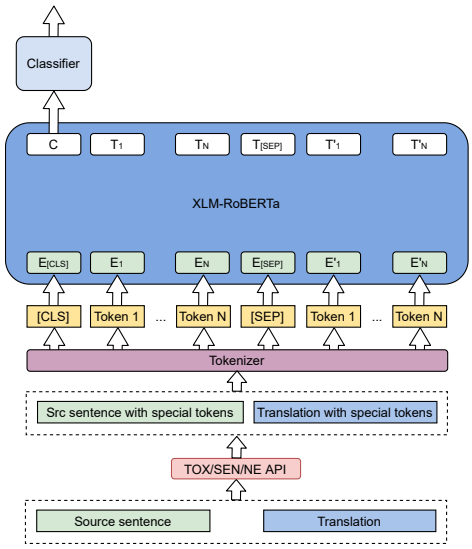


Figure 2: Architecture of the first approach (adding special tokens). We insert TOX/SEN/NE information to the source sentence and its translation as special tokens, and then feed sentences with special tokens to the baseline architecture.

The first approach (shown in Figure 2) is to **add special tokens**. Here the features (toxicity, sentiment, named entities) are directly inserted as special tokens to the input source sentence and, where available, its translation before getting tokenised. To correctly tokenise sentences with features, these special tokens are also added to the XLM-RoBERTa tokeniser. The remaining architecture is the same as the baseline model except for the dimension of model's word embeddings as the model's token embeddings should be resized when adding new tokens.

For the toxicity feature, a special token `[TOX]` is added to the beginning of the input token sequence if and only if the sentence is toxic. If the sentence is not toxic, the `[TOX]` token will not be

added. For En-De, the [TOX] token is applied to both source sentence and translation. But for other three language pairs it is only applied to the source sentence (English), because the Perspective API does not support Czech, Japanese and Chinese.

For the sentiment feature, there are three special tokens, [SEN_POS], [SEN_NEG], [SEN_NEU], representing positive, negative and neutral sentiment respectively. Each time only one token denoting sentence's sentiment is added to the beginning of that sentence. All the sentences should have one sentiment token at the beginning. The sentiment token is applied to both source sentence and translation for En-De, En-Ja and En-Zh. We do not perform experiments on sentiment feature for En-Cs due to lack of support on Czech from sentiment analysis API.

For named-entities feature, there are seven special token pairs corresponding to seven types of named entities generated by SpaCy API, e.g. [ORG] and [/ORG], [DAT] and [/DAT], etc. We use special token pairs to encase the named entities with relevant type in sentences at word level. Similarly to sentiment feature, the tokens of named-entities feature are also applied to both source and translation for En-De, En-Ja and En-Zh. Czech is not supported by spaCy, hence we do not apply this feature to En-Cs.

By adding extra features to the texts, we expect to guide the model with the toxicity/named-entities/sentiment information on the source sentence or the discrepancy of such information between the source sentence and the translation, which might indicate the existence of critical translation errors.

The second approach (shown in Figure 3) is to **modify hidden states**, where the extracted features are presented as numerals and appended to the hidden states of [CLS] token. Due to limited time, we only experimented with NE features using this approach. Since some named entity types are similar, they can be grouped as one type. In this approach, except for DAT, which is an independent category, we group ORG and PER as a category, CRD and ORD as a category, NRP and GPE as a category so that finally we have four categories. The feature that is used here is the count of the four NE types in source and target sentences. It is presented as a vector of length 8, where the first 4 numbers represent the counts of these NE categories for the source sentence, and the last 4 numbers are for the translation. First we feed the source sentence and its translation into the XLM-RoBERTa encoder, then we append the vector of counts to the output of the [CLS] token. The modified hidden states is then fed to the classification head.

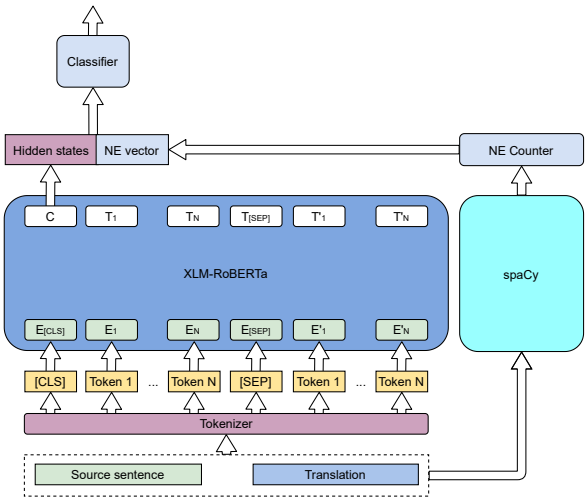


Figure 3: Architecture of the second approach (modifying hidden states). The sentence pair is fed into the XLM-RoBERTa encoder and into spaCy to generate NEs, resulting in the NE vector with the count of name entities of different types. We concatenate the output of [CLS] with the NE vector and send the modified hidden states to the classifier.

Our expectation is that the additional information (vector of counts) could guide the classifier to give more accurate predictions, because a deviation in named entity counts may be indicative of critical errors. For example, if the source sentence contains 3 named entities and the translation contains only 1 named entity, the translation may be missing some named entities.

2.3.3 Ensemble

To boost the performance, we ensemble several models to produce the final predictions. We experiment with two ensemble strategies. One strategy is label-level (late) ensemble. We first obtain the label predictions generated by different models using different features, then combine these predicted labels by performing majority vote to get a final label. The other strategy is logit-level ensemble, where we average the logits produced by different models and then produce the final label using the averaged logits.

3 Results

This section presents the evaluation results of the proposed methods. Except for the baseline score on

Method		En-Cs	En-De	En-Ja	En-Zh
Baseline		0.393	0.413	0.217	0.255
Baseline	best	0.397	0.422	0.231	0.276
(with sampler)	average	**0.396**	0.418	0.224	0.262
Adding TOX token	best	0.391	0.448	0.254	0.284
	average	0.385	**0.435**	0.220	0.261
Adding SEN token	best	–	0.429	0.228	0.296
	average	–	0.416	0.226	**0.276**
Adding NE (ORG) token	best	–	0.430	0.237	0.248
	average	–	0.422	0.211	0.239
Adding NE (PER) token	best	–	0.446	0.229	0.265
	average	–	0.415	0.224	0.225
Adding NE (DAT) token	best	–	0.439	0.220	0.259
	average	–	0.427	0.212	0.232
Adding NE (CRD) token	best	–	0.438	0.248	0.222
	average	–	0.426	0.195	0.217
Adding NE (ORD) token	best	–	0.442	0.236	0.262
	average	–	0.430	0.214	0.239
Adding NE (NRP) token	best	–	0.440	0.247	0.252
	average	–	0.420	0.193	0.247
Adding NE (GPE) token	best	–	0.429	0.220	0.251
	average	–	0.418	0.186	0.247
Modifying hidden states	best	–	0.455	0.257	0.280
	average	–	**0.435**	**0.238**	0.253

Table 3: Matthews's Correlation Coefficient (MCC) between predictions and gold labels using different methods on development set, trained on XLM-RoBERTa-base model. "Best" and "average" stand for the highest score and average score of three runs respectively. The bold numbers are the best result for the average of three runs in that language pair. For En-Cs, we only experiment on two cases due to lack of feature availability for Czech.

test set in Section 3.2 which is produced by MonoTransQuest using pretrained XLM-RoBERTa-base model with batch size of 128, learning rate of 2e-5, and a linear learning rate warm-up ratio of 10%, all the other scores (including baseline score on dev set in Section 3.1) are produced using following hyperparameters: 64 for batch size, 2e-5 for learning rate, 30% for the warm-up ratio.

3.1 Results on Dev Set

As described in Section 2.2, we explore nine feature types: source and target toxicity, source and target sentiment and 7 types of source and target named entities. We trained our model using the first approach (adding special token) for each of the nine feature types and the second approach (modifying hidden states) for named entities only. For each method or feature, we run the model for three times with different seeds and report average performance, as well as the performance of the best of the three models. The results on the development set are shown in Table 3.

The results follow our expectation that En-De could achieve the highest MCC score among the four language pairs as the training set of En-De is more balanced, compared to other three language pairs. Meanwhile, En-Ja has the lowest MCC score, as the dataset is the most imbalanced. The results also show that adding a weighted sampler to deal with imbalanced data improves the models' performance in most cases. As for the additional features, some of them are useful, but it depends on the language pair. For example, the toxicity feature can improve the score in En-De but cannot improve performance in En-Ja and En-Zh, while the sentiment token is helpful in En-Ja and En-Zh but not boost the score in En-De.

We note that the results may be affected by fluctuations because of different random seeds. Sometimes multiple runs of the same case will produce fairly different results. This is a general problem of neural models for QE as well as other tasks and requires further investigation. For example, the results of three runs of adding NE (NRP) feature in En-Ja vary a lot. The best score from the three runs is 0.247 which is over the baseline score, but the average score is 0.193 which is largely below the baseline.

3.2 Results on Test Set

We use ensembling to produce final results. The different models to ensemble are trained using different features, and hence focus on difference types of errors, thus potentially leading to different predictions. Not all these models lead to improvements over the base (no features) model; in fact, adding

some features decreases the performance for some languages. Therefore, we tested ensembles of models with all feature and ensembles of only models with features which achieve higher score on the development set in our ablation experiments (Table 3). We found that ensembling all models leads to a lower score than ensembling the best few models.

In our experiments, ensembling models with better performance than the base model improves the results of all languages except En-De. For that language pair, the best result is achieved by only adding NE (DAT) token to the XLM-RoBERTa-large base architecture. The final results of submission for CED shared task and ranks are shown in Table 4.

Method	**En-Cs**	**En-De**	**En-Ja**	**En-Zh**
Baseline	0.388	0.397	0.214	0.187
Ours (*gjiang*)	0.473 (2nd) reached by ensembling following: • 4 × baseline models (base) • 1 × baseline model (large)	0.498 (2nd) reached by single model: • 1 × NE (DAT) model (large)	0.314 (2nd) reached by ensembling following: • 1 × baseline models (base) • 1 × NE (CRD) model (base) • 1 × TOX model (large)	0.305 (4th) reached by ensembling following: • 1 × baseline models (base) • 2 × SEN models (base) • 1 × TOX model (base) • 1 × baseline model (large) • 1 × NE (DAT) model (large) • 1 × TOX model (large)

Table 4: Final results and ranks of CED shared task in WMT 2021. These results can also be found on CodaLab result page where the name of our team is *gjiang*. The metric in the table is Matthews's Correlation Coefficient (MCC) between predictions and gold labels. Below our score for each language pair is the model(s) that we used to achieve this score. "(base)" and "(large)" denotes this model is trained using XLM-RoBERTa-base and XLM-RoBERTa-large respectively.

Source	YOU SUCK IT!!! AS YOU'RE USED TO SUCK PHALLUS, NAZI HINDUIST LIKE HITLER!!!
Translation	你吸了它, 就像你以前吸过帕卢斯一样, 纳西迷人就像希特勒一样!
Baseline label	NOT
Our label	ERR
True label	ERR
Analysis	The source sentence is toxic and has negative sentiment. But the sentiment of translation is positive. Therefore, there is a deviation in sentiment between source and translation and this is a critical error.
Source	Upon further research I have found irrefutable proof that he got the nickname for the masterful way he cleaves beavers with his massive member.
Translation	经过进一步的研究, 我发现了不可否认的证据, 那就是他用他巨大的成员把贝弗切开的巧妙方法获得了绰号。
Baseline label	NOT
Our label	ERR
True label	ERR
Analysis	The source sentence is not toxic and the sentiments of both sentences are neutral. However, the machine translator mistakenly regards "beavers" as a name and produce a name in Chinese, which is detected by spaCy. The translation introduces one named entity which does not exist in source sentence. Therefore, this is a critical error.
Source	REDIRECT Talk:Historical Archive of the City of Cologne
Translation	主题演讲: 科隆市历史档案
Baseline label	NOT
Our label	ERR
True label	NOT
Analysis	In this case, spaCy does not report "Cologne" as a named entity in source sentence, but in translation it reports the city name in Chinese as a named entity (GPE). Therefore, our model regards the translation introduces a new named entity. There is a deviation in named entities between source and translation and this is mistakenly classified as a critical error.
Source	Goanikontes is an oasis is hidden within the Goanikontes Region.
Translation	戈亚尼科恩特是戈亚尼科恩特地区内的一个绿洲。
Baseline label	NOT
Our label	ERR
True label	NOT
Analysis	Similarly to previous case, spaCy correctly detects "Goanikontes" as a location name, but in translation spaCy mistakenly reports the corresponding location name in Chinese as person's name. Hence, our model thinks there is a deviation in named entities and predicts this case to a critical error. The mistakes from APIs are likely to lead the model to give wrong predictions.

Table 5: Case study: comparison of baseline predictions and our ensembled predictions in En-Zh

3.3 Qualitative Analysis

We conducted manual inspection on En-Zh in an attempt to understand whether the additional features actually contribute to better performance. The choice of the language pair that we analysed was determined by the availability of understanding languages in both sides. We compared our final submitted predictions on test set with the baseline predictions. We found that, compared to the baseline result, our final model predicts more `ERR` labels. 82 out of 1000 samples' label in test set are flipped from `NOT` to `ERR`, among which 35 samples are correct change (from false negative to true positive), 47 are incorrect (from true negative to false positive). We give some examples in Table 5 to compare our predictions with the baseline results. These examples show that feature engineering actually pushes the model to predict more `ERR`s. Overall this improves the performance to some extent, but is subjected to the correctness of the feature extractor. Inaccurate results from APIs will give the model wrong information and limit the improvement of performance of our models.

4 Conclusions

This paper describes our submission to sentence-level CED task in WMT21. Our work extends the baseline MonoTransQuest architecture by exploring feature engineering and model ensembling, as well as weighted sampling to deal with imbalanced datasets. Potentially due to the skewed distribution of labels in the dataset, the model performance varies substantially over different runs. However, our results averaged over multiple random seeds show that our feature engineering and ensembling lead to large improvements over the baseline. Our official submission achieves the 2nd position in En-Cs, En-De, En-Ja, and the 4th postion in En-Zh.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shuo Sun, Francisco Guzmán, and Lucia Specia. 2020. Are we estimating or guesstimating translation quality? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6262–6267, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Wikipedia Talk Corpus.

Papago's Submission to the WMT21 Quality Estimation Shared Task

Seunghyun S. Lim
Papago, Naver Corp.

Hantae Kim
Papago, Naver Corp.

Hyunjoong Kim
Papago, Naver Corp.

{shaun.lim, hantae.kim, soy.lovit}@navercorp.com

Abstract

This paper describes Papago's submission to the WMT 2021 Quality Estimation `Task 1: Sentence-level Direct Assessment`. Our multilingual Quality Estimation system explores the combination of Pretrained Language Models and Multi-task Learning architectures. We propose an iterative training pipeline based on pretraining with large amounts of in-domain synthetic data and finetuning with gold (labeled) data. We then compress our system via knowledge distillation in order to reduce parameters yet maintain strong performance. Our submitted multilingual systems perform competitively in multilingual and all 11 individual language pair settings including zero-shot.

1 Introduction

Quality Estimation (QE) evaluates the quality of machine translated output without human reference translation (Blatz et al., 2004). QE has a variety of applications in the Machine Translation (MT) pipeline and is particularly useful in industry settings by informing translation quality to end-users. High performance in sentence-level QE (Specia et al., 2020) is achieved by building a model on top of Pretrained Language Model (PLM); XLM-RoBERTa-large (Conneau et al., 2020) performs particularly well as shown in previous WMT sentence-level QE Shared Task. However, such PLMs contain extremely large number of parameters. This year's task is different from the previous years' task as submitted systems are ranked based on both model size[1] and model performance[2]. For concurrent work Gajbhiye et al. (2021) applies knowledge distillation (Hinton et al., 2015) from a PLM-based QE architecture to a much lighter BiRNN-based architecture, reducing memory requirements. Data scarcity is another issue relevant to QE tasks where there are often limited amount of gold training data. Previous WMT systems incorporate data augmentation techniques and show improvements in model performance when training with additional sources of data (Baek et al., 2020; Ranasinghe et al., 2020a).

Our system builds a model on top of PLM and trains with Multi-task Learning (MTL) (Caruana, 1997). Similar to Hoang et al. (2018); Zhang et al. (2018) where back-translation is iteratively applied to the same monolingual corpus to successively generate higher quality synthetic training data in the context of Neural Machine Translation (NMT), our proposed approach consists of an iterative knowledge transfer procedure which aims to repeatedly produce better quality pseudo labels for large amounts of synthetic training data. During the final stage of our training pipeline, knowledge distillation is applied from teacher to student model in order to reduce model size while maintaining competitive performance. We participate in WMT 2021 Quality Estimation (Specia et al., 2021) `Task 1` for multilingual and all individual language pair settings. Our system is a single multilingual sentence-level QE model that performs very strongly in both multilingual and individual language pair settings.

2 Data

In this year's task, participants are provided with 7K train set (`Train`), 1K development set (`Dev`), and 1K test set (`Test20`) for 7 language pairs: high-resource English-German (En-De) and English-Chinese (En-Zh), medium-resource Romanian-English (Ro-En) and Estonian-English (Et-En), and low-resource Sinhalese-English (Si-En) and Nepalese-English (Ne-En), as well as Russian-English (Ru-En). The source side sentences of language pairs excluding Ru-En are col-

[1] Disk space without compression and number of parameters.

[2] Pearson's correlation coefficient, root mean square error (RMSE), mean absolute error (MAE).

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 935–940

lected from Wikipedia data; the source side sentences of Ru-En is collected from a combination of Wikipedia articles and Reddit articles. Target side sentences are collected by translating source side sentences using NMT models and each sentence pair is annotated by at least three professional translators with a score between 0-100 according to the perceived translation quality. Systems are required to inference z-standardized direct assessment (DA) scores for 1K blind test set for each language pair. This year's task also include zero-shot scenario for 4 new language pairs: English-Czech (En-Cs), English-Japanese (En-Ja), Pashto-English (Ps-En), and Khmer-English (Km-En). As additional resource, participants are also provided with parallel data used to train NMT models (except for Ru-En and zero-shot language pairs) and NMT models used to generate target side sentences of the dataset.

3 Approach

Figure 1 summarizes our approach. Below we describe relevant components to our sentence-level QE model.

3.1 Base Model Architecture

Our QE model stacks feed-forward layers on top of feature vector extracted from Pretrained Language Models (PLM). Our choices of PLM are XLM-RoBERTa-base ($L = 12$) and XLM-RoBERTa-large ($L = 24$). Given source sentence src^X in language X and target sentence tgt^Y in language Y, the concatenation of src^X and tgt^Y are fed as input to the PLM and feature vector CLS_{cat} is produced by taking the concatenation of [CLS] representations from all layers of the PLM; our feature vector is based on using [CLS] token representation due to its superior performance over other pooling strategies (Ranasinghe et al., 2020b; Fomicheva et al., 2020). QE model f predicts direct assessment scores as follows:

$$\begin{aligned} f(src^X, tgt^Y) = W_{score} \cdot LeakyReLU(\\ W_2 \cdot LeakyReLU(\\ W_1 \cdot CLS_{cat} + b1) + b2) \end{aligned} \quad (1)$$

where $W_{score} \in \mathbb{R}^{1\times512}$, $W_2 \in \mathbb{R}^{512\times2048}$, $b_2 \in \mathbb{R}$, $W_1 \in \mathbb{R}^{2048\times N}$, $b_1 \in \mathbb{R}$, and N is XLM-RoBERTa's hidden dimension size (1024) times number of layers (L).

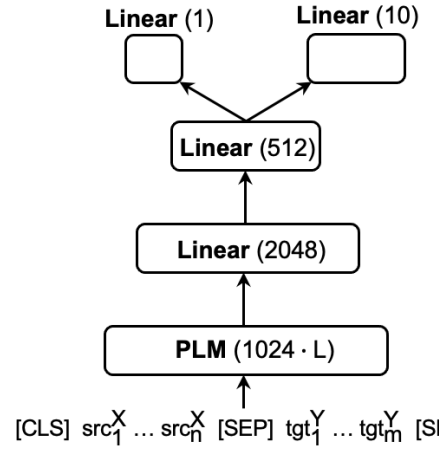


Figure 2: The network architecture for Multi-task Learning (§3.2) with XLM-RoBERTa as PLM. Concatenation of source and target sentences (with special tokens) are tokenized and fed as input to the PLM. Numbers in parenthesis denote the output dimension size of each network block.

3.2 Multi-task Learning (MTL)

We train our QE model in multi-task fashion by adding a classification objective to the base model architecture (§3.1). As shown in Figure 2, a classification layer W_{class}, where $W_{class} \in \mathbb{R}^{10\times512}$, is stacked next to W_{score} in equation (1). Given the n_{th} train set sample's z-standardized DA score $score_n$, we scale $score_n$ by applying min-max normalization and assign bin (class) labels to each sample. For our experiments, the number of bins is set to 10. Note that min-max scaling is applied to each language pair data set in order to account for different scales of $score_n$ per data set. The model is trained with a combined loss of mean squared error and cross entropy loss as shown in equation (2), with λ set to 0.6. Our intuition is that QE is inherently a complex task even for humans such that human-labeled DA scores may contain noise. We expect that training with an auxiliary classification loss, where bin labels are less susceptible to noise, can make training more robust and produce a model that is more generalizable.

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{mse} + (1 - \lambda) \cdot \mathcal{L}_{ce} \quad (2)$$

3.3 Data Augmentation

We create large amounts of synthetic direct assessment samples for 7 language pairs (non zero-shot) using parallel data and NMT models which both are provided as additional resource. For data augmentation, we utilize source side sentences from parallel data. We sub-sample from parallel data

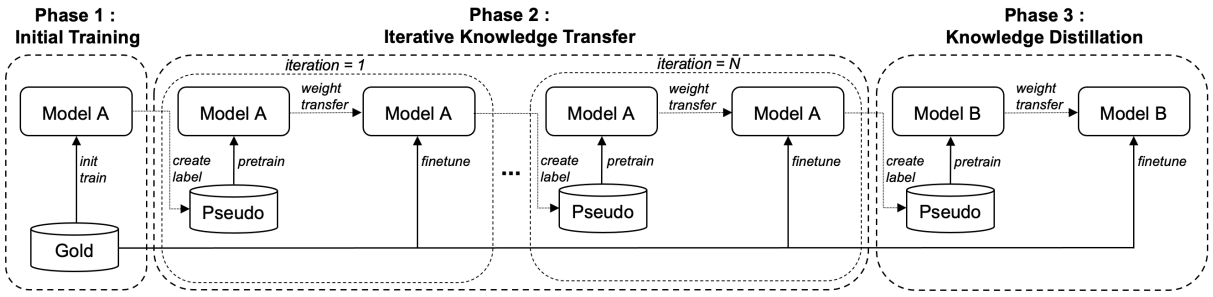


Figure 1: Pipeline of our proposed approach. `Gold` refers to `Train` set provided by task organizers (§2). `Pseudo` refers to synthetic sentence pairs generated as described in §3.3, while labels for `Pseudo` are created as described in §3.4. `Phase 2` and `Phase 3` each refer to §3.4 and §3.5 respectively. `Model A` and `Model B` each refer to large and small models in terms of model size. In our experiments, the architecture for `Model A` is a base model architecture (§3.1) with MTL (§3.2) using XLM-RoBERTa-large as PLM, which we denote as **Base$_{\text{large}}$ + MTL**; `Model B` instead uses XLM-RoBERTa-base as its PLM and we denote it as **Base$_{\text{small}}$ + MTL**.

Language	# parallel data	# sampled
En-De	19,298,476	400,000
En-Zh	15,178,232	400,000
Ro-En	3,901,626	400,000
Et-En	879,922	400,000
Ne-En	498,272	73,207
Si-En	646,781	400,000
Ru-En	12,061,155	400,000

Table 1: Number of parallel data provided in WMT2021 Task 1 and number of synthetic sentence pairs sampled as augmented data. For Ru-En, we collect parallel data from the Commoncrawl dataset.

for each language pair such that the distribution of sampled source sentences follows the distribution of source side sentences of gold data (§2) in terms of sentence length; this is to reduce the discrepancy between actual data and synthetic data. We then forward-translate source side sentences to target using provided NMT models to collect approximately 2.4M pseudo sentence pair data which are used as additional training resource. Table 1 shows the total amount of parallel data provided and the amount of synthetic sentence pairs generated. We describe how pseudo labels for synthetic data are created in the next section (§3.4).

3.4 Iterative Knowledge Transfer (IKT)

Given a QE model that is initially trained only on gold data (refer to `Phase 1` in Figure 1), iterative knowledge transfer aims to produce higher quality training signals or pseudo labels for synthetic data by iteratively performing pretraining and finetuning as shown in `Phase 2` in Figure 1. For pretraining, the model is always initialized with random weights (PLM weights are loaded from HuggingFace[3]) and is trained using synthetic direct assessment sentence pair data collected from §3.3. Pseudo labels for synthetic data in the current iteration are created with score predictions from model trained in the prior phase or iterative step. The aim of pretraining is to expose our model to large amounts of in-domain synthetic training data with sub-optimal labels. Similar to Sellam et al. (2020), the key aspect of the pretraining technique is to "warm up" the model before finetuning on gold data. At the start of finetuning, the model is initialized with parameter weights from the pretrain stage and is trained only with gold data. Because psuedo labels for synthetic data are newly generated for each iterative step in `Phase 2`, we expect the quality of "warm up" during pretraining to increase in each successive iteration. We stop the iterative process when the model's Pearson's correlation performance does not improve on `Test20`; we empirically find that performance does not improve after the second iteration.

3.5 Knowledge Distillation (KD)

`Phase 3` in Figure 1 demonstrates knowledge distillation from a large to smaller model. Akin to `Phase 2` (§3.4), a 2 stage pretrain-to-finetune training procedure is conducted and pseudo labels for synthetic data is generated using a teacher model which is the model produced from the last iteration of `Phase 2`. As our results will show, the compressed model performs on par with our baseline large model with approximately less than half the number of model parameters.

[3]https://huggingface.co/

Model	Data	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	Avg	# params
$\text{Base}_{\text{small}}$ (§3.1)	Dev	0.447	0.475	0.841	0.722	0.715	0.631	0.685	0.645	297M
	Test20	0.428	0.437	0.843	0.742	0.706	0.611	0.720	0.641	
+ MTL (§3.2)	Dev	0.452	0.496	0.847	0.737	0.730	0.639	0.683	0.655	297M
	Test20	0.473	0.449	0.854	0.740	0.729	0.625	0.732	0.657	
$\text{Base}_{\text{large}}$	Dev	0.488	0.496	0.891	0.788	0.794	0.703	0.715	0.696	611M
	Test20	0.481	0.473	0.882	0.803	0.762	0.664	0.764	0.690	
+ MTL	Dev	0.530	0.489	0.901	0.796	0.788	0.706	0.737	0.707	611M
	Test20	0.563	0.486	0.892	0.812	0.795	0.667	0.786	0.715	
+ $\text{IKT}_{\text{iter=1}}$ (§3.4)	Dev	0.550	0.527	0.906	**0.809**	0.798	**0.716**	**0.751**	0.722	611M
	Test20	0.543	**0.502**	**0.903**	0.814	**0.806**	0.676	0.791	0.719	
+ $\text{IKT}_{\text{iter=2}}$	Dev	**0.576**	**0.535**	**0.910**	0.807	**0.801**	0.714	0.742	**0.726**	611M
	Test20	**0.583**	0.497	0.901	**0.817**	0.792	**0.678**	**0.803**	**0.724**	
+ KD (§3.5)	Dev	0.523	0.522	0.880	0.773	0.758	0.680	0.712	0.692	297M
	Test20	0.544	0.488	0.883	0.770	0.764	0.662	0.756	0.695	

Table 2: Pearson's correlation with human judgments on the `Dev` and `Test20` set. Model names starting with + sign indicates approaches that are cumulative.

4 Settings

For all training phases and experiments, we train our model in data parallelism on multiple NVIDIA Tesla V100 GPUs for 3 epochs with batch size of 8 and is optimized with Adam (Kingma and Ba, 2015) with a learning rate of $7e^{-6}$. Dropout (Srivastava et al., 2014) with 0.15 is applied to activation function outputs in equation 1. Each model variant is trained 3 times with different random seeds, and for each model variant the best performing system in terms of Pearson's correlation coefficient is reported.

All models trained within the scope of this paper are multilingual QE models. We concatenate the `Train` set of each individual language pair to create a single multilingual train set for training. We apply the same for `Dev` and `Test20` set such that validation, model selection and evaluation can be performed at a multilingual level.

5 Results

In this section, we present results of our architectures described in §3. Pearson's correlation coefficient between predictions and gold standard scores is the main evaluation metric to measure performance; this year's task also considers model size to rank systems. Table 2 shows the Pearson's correlation with human judgments on the development and test set (`Dev` and `Test20`). Each row in Table 2 corresponds to model variants derived from certain phases of the training pipeline as described in Figure 1. We first observe that incorporating **MTL** (§3.2) improves over both our small baseline model $\textbf{Base}_{\textbf{small}}$ and large baseline model $\textbf{Base}_{\textbf{large}}$ with respect to all language pair settings. We observe further improvements in performance using Iterative Knowledge Transfer (§3.4) where the average performance of second iterative model $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}_{\textbf{iter=2}}$ is better than the first iterative model. Comparing $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}_{\textbf{iter=2}}$ to our large baseline model $\textbf{Base}_{\textbf{large}}$, the average performance gain is 3.4 percentage point but gain with respect to individual language pairs varies, with 10.2 percentage point increase for En-De being the greatest.

Our final compressed model $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}_{\textbf{iter=2}}\textbf{+KD}$ not only outperforms $\textbf{Base}_{\textbf{small}}\textbf{+MTL}$ in all language pairs but also outperforms our large baseline model $\textbf{Base}_{\textbf{large}}$ in 4 out of 7 language pair settings with less than half the number of model parameters.

Table 3 compares performance between the organizer's baseline model and two of our submitted systems. We submit two systems: $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}$ and $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT+KD}$. Systems can be evaluated on two ranking schemes: `R1` indicates overall ranking[4] which considers both model performance and size, while `R2`[5] ranks systems based only on model performance. As shown

[4]Overall ranking is computed by taking the average of individual ranks of the following metrics: Pearson's correlation coefficient, root mean square error, mean absolute error, disk space without compression and number of parameters.

[5]Ranking scheme based on Pearson's correlation coefficient

	+ IKT (§3.4)		+ KD (§3.5)		Organizer's
	Pearson	R2	Pearson	R1	Pearson
Multi	0.6577	4th	0.6132	3rd	0.5411
En-De	0.5677	3rd	0.5511	**2nd**	0.4025
En-Zh	0.5668	4th	0.5534	3rd	0.5248
Ro-En	0.9008	**2nd**	0.8786	3rd	0.8175
Et-En	0.7941	4th	0.7588	3rd	0.6601
Ne-En	0.8530	4th	0.8233	3rd	0.7376
Si-En	0.5947	3rd	0.5819	**1st**	0.5127
Ru-En	0.7927	**2nd**	0.7436	4th	0.6766
En-Cs	0.5722	4th	0.4969	5th	0.3518
En-Ja	0.3315	4th	0.2755	5th	0.2301
Ps-En	0.6368	**2nd**	0.5816	3rd	0.4760
Km-En	0.6616	**2nd**	0.6251	6th	0.5623
# params	611M		297M		281M
Disk space	2,503MB		1,249MB		1,142MB

Table 3: Submission results on the `Test21` blind set. `+IKT` refers to model from either row 5 or 6 of Table 2; `+KD` refers to model from row 7 of Table 2.

	Supervised							Zero-shot			
	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	En-Cs	En-Ja	Ps-En	Km-En
Δ Pearson	-0.016	-0.013	-0.022	-0.035	-0.029	-0.012	-0.049	-0.075	-0.056	-0.055	-0.036
Δ RMSE	+0.007	+0.019	+0.034	+0.039	+0.040	+0.022	+0.043	+0.017	+0.011	+0.032	+0.064
Δ MAE	+0.007	+0.020	+0.034	+0.040	+0.040	+0.023	+0.043	+0.017	+0.012	+0.033	+0.064
% Δ Pearson	-2.8	-2.3	-2.4	-4.4	-3.4	-2.1	-6.1	-13.1	-16.9	-8.6	-5.5
% Δ RMSE	+1.2	+3.0	+8.6	+7.6	+7.6	+2.9	+7.5	+2.2	+1.2	+4.3	+7.3
% Δ MAE	+1.2	+3.2	+8.6	+ 7.8	+7.6	+3.0	+7.5	+2.2	+1.4	+4.4	+7.8

Table 4: Changes in performance on the `Test21` blind set when transitioning from `+IKT` (before compression) to `+KD` (after compression). `Supervised` indicates 7 language pairs that are provided in `Train`, `Dev` and `Test20`; `Zero-shot` indicates 4 zero-shot language pairs that are only evaluated with `Test21` blind set. Δ `metric` (row 1 to 3) measures the change in performance; % Δ `metric` (row 4 to 6) measures the percentage change.

in Table 3, when ranking systems based purely on performance (`R2`), **Base$_{\text{large}}$+MTL+IKT** performs strongly. However, when systems are ranked based on both performance and size (`R1`), our compressed model **Base$_{\text{large}}$+MTL+IKT+KD** ranks very competitively. Moreover, our compressed model outperforms the organizer's baseline in all language pair settings with a great margin using approximately 5.7% more parameters.

We observe in Table 3 that our compressed model is relatively less competitive under zero-shot than in supervised settings when ranked based on `R1`. As demonstrated in Table 4, model compression causes performance degradation in all language pairs with respect to all three performance metrics. In particular, the amount of degradation in terms of Pearson's correlation coefficient is greater under zero-shot than in supervised settings. Interestingly, this trend does not apply to other performance metrics (RMSE, MAE) where the amount of degradation under zero-shot and supervised settings is not significantly different. This indicates that model compression degrades the strength of correlation particularly more under zero-shot than in supervised settings, while degradation in performance measured by magnitude of error is not significantly different between two settings.

6 Conclusions

In this paper, we describe our submission to the WMT 2021 Quality Estimation `Task 1: Sentence-level Direct Assessment`. We introduce a QE model architecture trained with multi-task objective and show improvements in performance. We show that iterative knowledge transfer techniques applied in QE tasks can further

improve model's performance and demonstrate that knowledge distillation is effective for building a competitive lighter-weight QE model, making it more suitable for practical use. Although our submitted systems show strong performance in general, we observe that our compressed model becomes relatively less competitive under zero-shot settings. Further analysis of this phenomenon and improvements on zero-shot are challenges that we need to overcome in future work.

References

Yujin Baek, Zae Myung Kim, Jihyung Moon, Hyunjoong Kim, and Eunjeong Park. 2020. PATQUEST: Papago translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 991–998, Online. Association for Computational Linguistics.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020. BERGAMOT-LATTE submissions for the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017, Online. Association for Computational Linguistics.

Amit Gajbhiye, Marina Fomicheva, Fernando Alva-Manchego, Frédéric Blain, Abiola Obamuyide, Nikolaos Aletras, and Lucia Specia. 2021. Knowledge distillation for quality estimation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5091–5099, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. 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(56):1929–1958.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

NICT Kyoto Submission for the WMT'21 Quality Estimation Task: Multimetric Multilingual Pretraining for Critical Error Detection

Raphael Rubino and **Atsushi Fujita** and **Benjamin Marie**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
raphael.rubino,atsushi.fujita,bmarie@nict.go.jp

Abstract

This paper presents the NICT Kyoto submission for the WMT'21 Quality Estimation (QE) Critical Error Detection shared task (Task 3). Our approach relies mainly on QE model pretraining for which we used 11 language pairs, three sentence-level and three word-level translation quality metrics. Starting from an XLM-R checkpoint, we perform continued training by modifying the learning objective, switching from masked language modeling to QE oriented signals, before finetuning and ensembling the models. Results obtained on the test set in terms of correlation coefficient and F-score show that automatic metrics and synthetic data perform well for pretraining, with our submissions ranked first for two out of four language pairs. A deeper look at the impact of each metric on the downstream task indicates higher performance for token oriented metrics, while an ablation study emphasizes the usefulness of conducting both self-supervised and QE pretraining.

1 Introduction

This paper describes the NICT Kyoto submission to the WMT'21 Quality Estimation (QE) shared task. We participated in Task 3 "Critical Error Detection" involving four language pairs, namely English–Chinese, English–Czech, English–Japanese and English–German. A critical error is defined as a translation error falling into one of the following five categories: toxicity, health or safety risk, named entity, sentiment polarity and number or unit deviation.[1]

The objective of the task is to classify a sequence pair, composed of a sentence in the source language and its automatic translation in the target language, in a binary fashion whether it contains or not at least one of the five types of critical errors. This task differs from the other QE tasks as not all translation errors should be detected but only critical ones. Labels were produced by majority vote over three annotators for each pair leading to two possible classes: *ERR* (or class 1) when at least one critical error is spotted and *NO* (or class 0) when no critical errors are present.

Our approach relies mainly on QE model pretraining leveraging a large amount of synthetic data produced using parallel corpora and MT systems. Because annotating translations for critical error is costly, we propose to pretrain a model on translation quality scores computed with automatic metrics. To capture multiple translation error granularities during pretraining, we employ multiple metrics and evaluate their performance individually on the downstream task. Additionally, we pretrain the QE model jointly on all WMT QE shared tasks language pairs as a data augmentation method. Transfer learning is then conducted for each language pair by finetuning the pretrained model on the downstream task with the officially released training data annotated with critical errors.

The remainder of this paper is organized as follows. In Section 2, we introduce our approach involving multimetric and multilingual pretraining. In Section 3, the data, tools and training procedure are presented, followed by the experimental results and their analysis in Section 4, before the conclusion in Section 5.

2 Multimetric & Multilingual Pretraining

Multilingual pretrained masked language models (LMs) were shown to perform well in several downstream natural language processing tasks (Devlin et al., 2019; Conneau et al., 2020; Liu et al., 2020). Starting from an XLM-R checkpoint (Conneau et al., 2020), we performed continued (or intermediate) training (Phang et al., 2018; Rubino and Sumita, 2020) with large amount of automatically

[1] More details about these categories and the task itself can be found here: http://statmt.org/wmt21/quality-estimation-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 941–947
November 10–11, 2021.

translated source language texts (thereafter called *synthetic data*), replacing the masked LM objective with QE oriented ones. Because XLM-R is multilingual and all languages in this model share a common vocabulary of sub-words, we decided to conduct QE pretraining on the 11 language pairs from all subtasks of WMT'21 QE. These language pairs all share English, whether on the source or target side, and this method can be seen as a data augmentation approach to increase vocabulary coverage.

The objective of QE Task 3 is to classify sentence pairs in a binary fashion. Formally, given a source sequence s and its translation t, we want to learn a function f: $f_\theta(s,t) \rightarrow y$ where $y \in \{0,1\}$ is the class associated with the sequence pair (s,t) and θ represents the model parameters. While fine-tuning a pretrained model on the official QE task 3 data allows us to directly learn model parameters approximating y given (s,t), we do not have such classes for synthetic data. We decided to use MT automatic metric scores as objective instead, assuming that critical error classes could correlate with translation quality scores at least in extreme cases (e.g. no translation errors also means no critical errors).

Several automatic metrics are used by the research community to evaluate the performance of MT systems by measuring translation accuracy against a human-produced reference at different granularity levels. We opted for metrics capturing quality information at the character (chrF (Popović, 2017)), token (TER (Snover et al., 2006)) and token n-gram (BLEU (Papineni et al., 2002)) levels. For the latter, the smoothed sentence-level BLEU was chosen (Chen and Cherry, 2014). In addition to sentence-level metrics, token-level binary tags were also extracted following the usual procedure to determine post-editing effort (Specia et al., 2020).[2]

To allow for sentence-level QE predictions, we added a feed-forward layer on top of XLM-R for each of the three metrics employed without parameter sharing, following:

$$\hat{y}_s = \tanh\left(\phi(h)W_{s1} + b_{s1}\right)W_{s2} + b_{s2} \quad (1)$$

where $\hat{y}_s \in \mathbb{R}^1$ is the sentence-level score, $W_{s1} \in \mathbb{R}^{d\times d}$, $b_{s1} \in \mathbb{R}^d$, $W_{s2} \in \mathbb{R}^{d\times 1}$ and $b_{s2} \in \mathbb{R}^1$ are parameters of the model with dimensionality $d = 1,024$, ϕ is a pooling function and $h \in \mathbb{R}^{n\times d}$ is the set of contextual embeddings corresponding to the n tokens in (s,t). The pooling function is the *class* token added at the beginning of each input sequence. For token-level predictions, we used a linear transformation from contextual embeddings to two-dimensional output (for binary token-level classes): $\hat{y}_t = \text{softmax}\left(hW_t + b_t\right)$, with $\hat{y}_t \in \mathbb{R}^{n\times 2}$ are token-level scores, $W_t \in \mathbb{R}^{d\times 2}$ and $b_t \in \mathbb{R}^2$ are the parameter matrix and bias. Parameters of the model are learned with mini-batch stochastic gradient descent based on losses computed for sentence-level and token-level predictions. For the former loss, we used mean squared error, while cross-entropy was used for the latter. All losses are linearly summed with equal weights before back-propagation. The parameters of the classification and regression heads are optimized along with XLM-R.

3 Data and Tools

This section presents the data used in our experiments, including the synthetic data produced for pretraining and the official QE task 3 corpora, along with the tools required to train our models and the procedure employed for both pretraining and fine-tuning.

3.1 Datasets

In order to gather as much data as possible for many language pairs, we collected all parallel data from the QE shared tasks (from all subtasks). Additionally, we retrieved parallel data from the WMT news translation task (Barrault et al., 2020) and from OPUS (Tiedemann, 2016).[3] The source side of these parallel corpora was translated using publicly available neural MT models based on the Transformer architecture (Vaswani et al., 2017). For Estonian–English (et–en), Nepalese-English (ne–en), Romanian–English (ro–en), Russian–English (ru–en), Sinhala–English (si–en), English–German (en–de) and English–Chinese (en–zh), we used the MT systems made available by the shared task organizers,[4] while for English–Czech (en–cs), English–Japanese (en–ja), Khmer–English (km–en) and Pashto–English (ps–en), we used the mBART50

[2]Scripts and procedure available at https://github.com/deep-spin/qe-corpus-builder

[3]The corpora from OPUS used in our experiments are: Common Crawl, ParaCrawl, OpenSubtitles, DGT, IWSLT, KFTT and XLEnt.

[4]Links to models available at https://github.com/facebookresearch/mlqe/blob/master/nmt_models/README-models.md

Lang. src	Lang. tgt	Sent.	Token src	Token tgt	Type src	Type tgt
Synthetic Data (pretraining)						
en	cs	14.1M	244.4M	220.2M	2.3M	2.5M
en	de	22.3M	477.5M	442.9M	2.5M	4.6M
en	ja	3.3M	64.7M	86.7M	1.2M	732.1k
en	zh	16.2M	407.2M	350.4M	1.1M	1.1M
et	en	14.8M	143.3M	176.8M	2.3M	0.9M
km	en	3.7M	47.7M	34.8M	1.3M	480.5k
ne	en	0.9M	10.1M	8.5M	307.6k	343.2k
ps	en	1.0M	11.6M	10.2M	332.6k	190.3k
ro	en	2.3M	55.7M	51.9M	331.7k	261.1k
ru	en	5.0M	82.1M	90.1M	1.8M	0.9M
si	en	1.4M	17.6M	12.8M	366.7k	344.4k
Official QE Task 3 Data (finetuning)						
en	cs	7.5k	122.2k	125.9k	23.6k	22.5k
en	de	7.9k	127.7k	154.6k	24.7k	19.6k
en	ja	7.7k	126.3k	213.7k	24.6k	12.8k
en	zh	6.9k	110.7k	122.9k	21.9k	12.9k

Table 1: Number of sentences (*Sent.*), tokens and types in the source (*src*) and target (*tgt*) corpora used in our experiments (M stands for millions and k for thousands).

model (Liu et al., 2020; Tang et al., 2020).[5]

Statistics about the synthetic corpora after translation are presented in Table 1, along with the official QE data for Task 3 released by the shared task organizers. After deduplicating and cleaning the synthetic corpora produced to conduct QE pretraining, the total amount of data reached 72.3M triplets (source, translation and reference sentences).

3.2 Tools

Data preprocessing was conducted using the tokenizer and truecaser from the Moses distribution (Koehn et al., 2007), except for Chinese, Japanese, Nepalese and Sinhala, for which the tokenization was conducted using *jieba*,[6] *KyTea*[7] and FLORES (Goyal et al., 2021) respectively.

To compute the sentence-level and token-level scores, we used automatic metrics implementations available in the tools *SacreBLEU* (Post, 2018) for BLEU and chrF and *tercom* (Snover et al., 2006) for TER and token-level classes.

The XLM-R checkpoint used was the *xlm-roberta-large* from HuggingFace Transformers library (Wolf et al., 2020). We used in-house Pytorch (Paszke et al., 2019) code and V100 GPUs hardware for QE pretraining and finetuning, 8 GPUs for the former step and 1 GPU for the latter.

3.3 Training Procedure

Model pretraining on synthetic data was conducted for one epoch (approx. 500k updates) with batches of 128 source and target sequences for a total training time of 3 days. The AdamW optimizer (Loshchilov and Hutter, 2019) was used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-6}$, while the weight decay was set to 0. A linear learning rate warmup was used during the first 50k updates to reach a maximum value of 5×10^{-6}, which remained without decay until the end of the first epoch. The dropout rates were set to 0.1 for both the embeddings and the transformer blocks (feed-forward and attention layers). A total of four models were pretrained with different random seeds before being finetuned on the official QE Task 3 data.

To conduct finetuning, we added a classification layer on top of XLM-R following:

$$\hat{y}_e = \text{softmax}(\tanh(\phi(h)W_{e1} + b_{e1})W_{e2} + b_{e2}) \quad (2)$$

where $\hat{y}_e \in \mathbb{R}^2$ is the sentence-level probability distribution over the two classes, $W_{e1} \in \mathbb{R}^{d \times d}$, $b_{e1} \in \mathbb{R}^d$, $W_{e2} \in \mathbb{R}^{d \times 2}$ and $b_{e2} \in \mathbb{R}^2$ are parameters of the model with $d = 1,024$. The pooling function ϕ is the same as the one employed during pretraining presented in Section 2. Due to the class imbalance of the critical error dataset, we used the weighted cross-entropy loss function to finetune our models. The weight given to the error class (the least populated) was tuned on the validation set in a grid-search manner, with integer values ranging from 1 to 8.

During finetuning, which lasted 40 minutes per model, we used the validation set to select the best performing models according to the Matthews correlation coefficient (MCC), which is the main metric chosen by the shared task organizers for the final evaluation. One model per seed was selected and a total of four models were ensembled to produce our final submission to the shared task.

4 Results and Analysis

We present in this section the main results obtained on the official shared task test set as reported by the organizers, followed by an analysis with ablation study and various pretraining objectives.

[5] More details about the model available at https://github.com/pytorch/fairseq/tree/master/examples/multilingual

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

[7] http://www.phontron.com/kytea/

Lang.	MCC	F1 ERR	F1 NOT	F1 Multi
Official Baseline				
en–cs	0.3875	0.8992	0.4768	0.4287
en–de	0.3974	0.8484	0.5317	0.4511
en–ja	0.2139	0.9505	0.2439	0.2318
en–zh	0.1873	0.8980	0.2694	0.2419
Our Baseline				
en–cs	0.4030	0.8984	0.4985	0.4478
en–de	0.5204	0.8687	0.6495	0.5642
en–ja	0.2523	0.9294	0.3191	0.2966
en–zh	0.2413	0.8667	0.3714	0.3219
Our Ensemble				
en–cs	0.5105	0.9132	0.5949	0.5433
en–de	0.5464	0.8767	0.6667	0.5845
en–ja	0.2375	0.9447	0.2896	0.2736
en–zh	0.3109	0.8833	0.4260	0.3763

Table 2: Results obtained on the test set for the WMT'21 QE shared task, Task 3 "Critical Error Detection". *F1 ERR* denotes the F-score obtained on the error class, *F1 NOT* denotes the F-score obtained on the non-error class, *F1 Multi* stands for the multiplication of *F1 ERR* and *F1 NOT*.

Lang.	MCC	F1 ERR	F1 NOT	F1 Multi
No Checkpoint				
en–cs	0.3844	0.4847	0.8996	0.4360
en–de	0.3796	0.5575	0.8219	0.4582
en–ja	0.1963	0.2047	0.9461	0.1937
en–zh	0.2461	0.3513	0.8948	0.3143
No QE Pretraining				
en–cs	0.4728	0.5593	0.9132	0.5107
en–de	0.5182	0.6192	0.8804	0.5451
en–ja	0.2999	0.3439	0.9441	0.3247
en–zh	0.3649	0.4633	0.8897	0.4122
Checkpoint + QE Pretraining				
en–cs	0.5271	0.6000	0.9266	0.5560
en–de	0.5501	0.6615	0.8829	0.5840
en–ja	0.3286	0.3497	0.9499	0.3322
en–zh	0.3833	0.4784	0.8905	0.4260

Table 3: Results obtained on the WMT'21 QE Task 3 "Critical Error Detection" validation set. All results are obtained with ensemble of 4 models. *No Checkpoint* denotes QE pretraining of randomly initialized XLM-R without usual masked LM pretraining, followed by finetuning, *No QE Pretraining* denotes direct finetuning of an XLM-R checkpoint on the official task specific training data, *Checkpoint + QE Pretraining* is our submission to the shared task based on XLM-R and QE pretraining with finetuning.

4.1 Shared Task Results

The official results reported by the shared task organizers are presented in Table 2. We compare our final ensemble results, obtained with four models trained on different seeds, to our baseline, obtained with a single model. We also include the official baseline provided by the shared task organizers. All our submissions outperform the official baseline and our ensembles reach the highest performance according to the correlation score and F-measure. One exception, however, is for the English–Japanese language pair. Despite several attempts to improve our ensembling method for this pair, we could not improve over our baseline.

A comparison with other shared task participants in terms of MCC and F1 scores shows that our submissions were ranked first for English–Czech and English–German, third for English–Chinese and sixth for English–Japanese. We assume that the smaller amount of synthetic data, as well as a possible preprocessing mismatch between the official data and our synthetically generated corpora, could be the reason behind the low performance of the two latter language pairs. More precisely, the data preprocessing pipeline for English, German and Czech are commonly based on the Moses tokenizer and truecaser, and it is possible to infer the parameters used with these tools by looking at the official training data released for the task. For Chinese and Japanese, however, due to the lack of details given by the shared task organizers, it was not possible to use the same preprocessing tools and parameters with certainty.

4.2 Impact of Pretraining Steps

While our approach relied on a two-step process, QE pretraining on synthetic data followed by finetuning on the task specific training set, we still made use of a pretrained XLM-R model by initiating QE pretraining from a checkpoint. Overall, three steps are thus required to obtain the results presented in Table 2. XLM-R and QE pretraining, as well as producing synthetic data, are the most computationally expensive steps, whereas finetuning is relatively cheap to perform due to the small amount of task specific data. Therefore, we performed an ablation study aiming at evaluating the impact of each pretraining step and ran two sets of experiments following the same experimental setup employed for our main submission to the shared task.

For the first set of experiments, no pretraining of XLM-R was conducted, meaning that we did not start QE pretraining from an existing checkpoint, but instead randomly initialized XLM-R parameters and ran QE pretraining *from scratch* (this setup is noted *No Checkpoint*). For the second set of

Lang.	MCC	F1 ERR	F1 NOT	F1 Multi
TER pretraining				
en–cs	0.4725	0.5605	0.9235	0.5176
en–de	0.5092	0.6378	0.8786	0.5604
en–ja	0.2891	0.3628	0.9490	0.3443
en–zh	0.3284	0.4324	0.9158	0.3960
BLEU pretraining				
en–cs	0.4760	0.5629	0.9266	0.5216
en–de	0.4917	0.6290	0.8725	0.5488
en–ja	0.2982	0.3636	0.9513	0.3459
en–zh	0.3442	0.4450	0.9061	0.4032
chrF pretraining				
en–cs	0.4200	0.4988	0.9210	0.4594
en–de	0.4122	0.5911	0.8540	0.5048
en–ja	0.2375	0.3163	0.9496	0.3004
en–zh	0.2925	0.3838	0.9242	0.3547
All sentence-level pretraining				
en–cs	0.4700	0.5539	0.9258	0.5128
en–de	0.5229	0.6609	0.8726	0.5767
en–ja	0.2982	0.3636	0.9496	0.3453
en–zh	0.3660	0.4639	0.9207	0.4271
All word-level pretraining				
en–cs	0.4697	0.5556	0.9172	0.5096
en–de	**0.5323**	0.6667	0.8728	0.5819
en–ja	0.3100	0.3743	0.9505	0.3558
en–zh	**0.3756**	0.4688	0.9127	0.4279
All metrics pretraining				
en–cs	**0.5015**	0.5796	0.9289	0.5384
en–de	0.5276	0.6431	0.8779	0.5646
en–ja	**0.3131**	0.3824	0.9507	0.3635
en–zh	0.3546	0.4391	0.9112	0.4001

Table 4: Results obtained on the WMT'21 QE Task 3 "Critical Error Detection" validation set with single models (no ensemble) based on various learning objectives used during pretraining. Results in bold indicate the best MCC scores among the pretraining configurations for a given language pair.

experiments, we finetuned the XLM-R checkpoint directly on the task specific data, without conducting QE pretraining. This alleviates the need to produce large amount of synthetic QE data (this setup is noted *No QE Pretraining*). We conducted an additional set of experiments based on XLM-R and QE pretraining without finetuning on the official training set but the obtained results were subpar compared to the baseline, due to the randomly initialized parameters of the classification layer (see eq. (2)) which was not tuned for the task following this configuration. We present the results of the two ablation experiments in Table 3.

While combining both the use of a pretrained XLM-R with masked LM and QE pretraining on synthetic data leads to the best results on the four language pairs, *No QE Pretraining* performs better than the *No Checkpoint* configuration. These results emphasize the usefulness of large self-supervised LM pretraining. The amount of data used for QE pretraining is smaller compared to the large quantity of monolingual and parallel data used to train *xlm-roberta-large*, which could be an explanation for the difference in downstream performances according to the MCC and F1 metrics.

4.3 Impact of Pretraining Objectives

As an additional analysis, we propose to evaluate the impact of different metrics used as pretraining objectives on the downstream critical error detection task. Several independent QE pretraining were conducted for this purpose: one for each sentence-level translation quality metrics, one for the combination of sentence-level metrics and finally one for word-level metrics which includes source, target and gap error predictions as described in Section 2. The finetuning step for each pretrained model is identical, only the learning objective during pretraining differs. The results obtained on the validation set for the critical error detection task are presented in Table 4.

Based on MCC scores, using sentence-level metrics during pretraining is not leading to the best downstream performance compared to using word-level metrics or combining both sentence and word-level quality indicators. From the three sentence-level metrics used as learning objectives during pretraining, TER and BLEU outperform chrF. For English–German and English–Chinese, using word-level metrics outperforms the combination of all metrics, while it is the opposite for English–Czech and English–Japanese. These results show that the optimal quality indicator for QE pretraining depends on the language pair and the translation direction, and should therefore be considered as a hyper-parameter to be optimized. However, due to the costly nature of large model pretraining, combining multiple translation quality indicators in a multi-task learning fashion appears to be an efficient solution, in addition to using masked LM pretrained model as shown in the results presented in Section 4.2.

5 Conclusion

This paper presented the NICT Kyoto submission for the WMT'21 QE Task 3 "Critical Error Detection". Our submissions were ranked first for two out of four language pairs. Our approach relies mainly on model pretraining with large amount of

synthetic data, followed by finetuning on the official data released for the shared task. We proposed a novel QE pretraining approach which allows for a multimetric learning objective based on relatively cheap to compute MT automatic metrics. An analysis of each automatic metric used during QE pretraining shows the complementarity of metrics both at level of sentences and words. The ablation study emphasized the usefulness of both self-supervised and QE pretraining. Future work focuses on exploring additional metrics and their performance on various downstream QE tasks.

Acknowledgements

We would like to thank the reviewers for their insightful comments and suggestions. A part of this work was conducted under the commissioned research program "Research and Development of Advanced Multilingual Translation Technology" in the "R&D Project for Information and Communications Technology (JPMI00316)" of the Ministry of Internal Affairs and Communications (MIC), Japan, and supported by JSPS KAKENHI grant numbers 20K19879 and 19H05660.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. 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, Baltimore, USA. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, USA. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The FLORES–101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *arXiv preprint arXiv:2106.03193*.

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 Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

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, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pages 8026–8037. Curran Associates, Inc.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks. *arXiv preprint arXiv:1811.01088*.

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.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Raphael Rubino and Eiichiro Sumita. 2020. Intermediate Self-supervised Learning for Machine Translation Quality Estimation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4355–4360, Barcelona, Spain (Online). International Committee on 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: Technical Papers*, pages 223–231, Cambridge, USA. Association for Machine Translation in the Americas.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 Shared Task on Quality Estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual Translation with Extensible Multilingual Pretraining and Finetuning. *arXiv preprint arXiv:2008.00401*.

Jörg Tiedemann. 2016. OPUS – Parallel Corpora for Everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

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. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

QEMind: Alibaba's Submission to the WMT21 Quality Estimation Shared Task

Jiayi Wang*, **Ke Wang***, **Boxing Chen, Yu Zhao, Weihua Luo, Yuqi Zhang**†
Alibaba Group, Hangzhou, China
{joanne.wjy,moyu.wk,boxing.cbx}@alibaba-inc.com,
kongyu@taobao.com, {weihua.luowh,chenwei.zyq}@alibaba-inc.com

Abstract

Quality Estimation, as a crucial step of quality control for machine translation, has been explored for years. The goal is to investigate automatic methods for estimating the quality of machine translation results without reference translations. In this year's WMT QE shared task, we utilize the large-scale XLM-Roberta pre-trained model and additionally propose several useful features to evaluate the uncertainty of the translations to build our QE system, named *QEMind*. The system has been applied to the sentence-level scoring task of Direct Assessment and the binary score prediction task of Critical Error Detection. In this paper, we present our submissions to the WMT 2021 QE shared task and an extensive set of experimental results have shown us that our multilingual systems outperform the best system in the Direct Assessment QE task of WMT 2020.

1 Introduction

Quality estimation (QE) aims to predict the quality of a machine translation (MT) system's output without any access to ground-truth translation references or human intervention (Blatz et al., 2004; Specia et al., 2009, 2018). Automatic methods for QE are highly appreciated in MT applications when we expect to efficiently obtain the quality indications for a larget amount of machine translation outputs in a short time, or even at run-time. This paper describes Alibaba's submissions to the WMT 2021 Quality Estimation Shared Task. We developed a novel QE system, called QEMind, that have been applied to two tasks this year, the sentence-level direct assessment (DA) and binary score prediction of Critical Error Detection (CED).

Common approaches in the previous years heavily focus on human-crafted rule-based feature engineering mode such as QuEst++ (Specia et al., 2015).

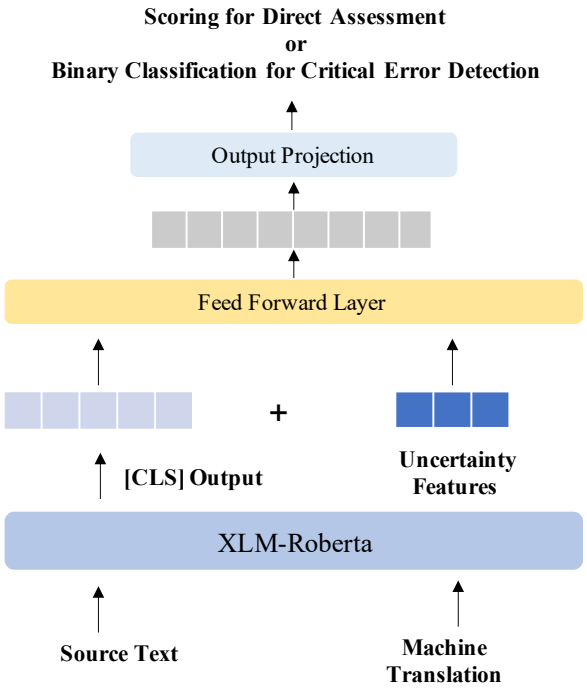


Figure 1: Structure of the uncertainty quantification feature-enhanced model.

The features extracted are usually fed into traditional machine learning algorithms such as a support vector regression for the sentence-level scoring or a sequence-labeling model with conditional random fields for the word-level labeling respectively. With the development of neural networks applied in machine translation and other NLP tasks, a neural predictor-estimator framework for QE was proposed and achieved better results in WMT 2017 and WMT 2018 QE shared tasks (Fan et al., 2019; Kim et al., 2017). This framework extensively requires a pre-training procedure with a large amount of parallel corpora in the predictor mode and stacks a downstream estimator mode with additional layers for a supervised regression or classification task. Since 2019, state-of-the-art (SOTA) QE systems (Kepler et al., 2019; Ranasinghe et al., 2020) have

* indicates equal contribution.
† indicates corresponding author.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 948–954
November 10–11, 2021.

hit record high with transfer learning by leveraging SOTA pre-trained NLP neural network models, for example, mBERT (Pires et al., 2019) and XLM-Roberta (Conneau et al., 2019). Till then, only "black-box" QE methods had been mainly used in WMT QE shared tasks.

Furthermore, with the accessibility to the NMT systems, some "glass-box" QE features have been explored and verified to bring improvements upon "black-box" approaches (Moura et al., 2020). In addition, Fomicheva et al. (2020) have showed that useful information that are extracted from the MT systems performs good correlation with human judgements of quality. Inspired by these works, we propose more useful features in this paper, among which, some are derived from the NMT systems and others are created via utilizing the masked language model of XLM. We develop our QE systems by incorporating all the features that can potentially evaluate the uncertainty of the machine translations into a supervised QE model based on the transfer learning from XLM-Roberta. We evaluate our method on the Direct Assessment QE tasks of WMT 2020 and WMT 2021 and our experiment results demonstrate the efficiency and versatility of the features we have proposed on the quality estimation in different language pairs.

2 Task & Data Set

We participate the sentence-level Direct Assessment task and Critical Error Detection tasks of this year's QE shared task. (1) For the DA task, we merge 7000 and 1000 labeled data in the training and development data sets as our training set and treat the test20 data set as our development set for each of the seven language pairs. However, for the four zero-shot language pairs, we only have the blind test sets. (2) For the CED task, we observed that the distributions of two classes, NOT and ERR, are extremely unbalanced for all four language pairs. Therefore, we simply up-sample the samples with ERR labels to get a relatively balanced training set. This strategy of data augmentation has also been empirically verified to be valid.

3 Methodology

In this section, we provide a complete view of our uncertainty feature enhanced approach, including:

(1) The overall framework of QEMind is carried out in Section 3.1: how uncertainty features are combined with a pre-trained multilingual language model to enhance transfer learning;

(2) Uncertainty features used in QEMind are described in Section 3.2: how uncertainty features are defined and extracted for translation quality estimation;

(3) Strategies we applied in the WMT QE shared task to further improve the system's performance, such as data augmentation and model ensemble, are explained in Section 3.3.

3.1 QEMind Framework

QEMind follows the general transfer learning procedure while allowing extra meta features to enhance the model. We concatenate the source text and machine translation and feed them into the pre-trained XLM-Roberta model to get the output representation of the special [CLS] token. Afterwards, the output representation is combined with the normalized uncertainty features described in Section 3.2. They are fed into a simple linear regression/classification layer to predict the continuous or binary quality score. The architecture of our feature enhanced model is shown in Figure 1. This model is equivalent to TransQuest's (Ranasinghe et al., 2020) when no extra feature is used. Considering the size of the training set is small, we have not added extra parameters, such as bottleneck adapter layers used in Moura et al. (2020), to fuse uncertainty features and the output from XLM-Roberta.

3.2 Uncertainty Features

Fomicheva et al. (2020) proposed several "glass-box" features extracted from the NMT model. Estimating translation quality with these features achieves state-of-the-art results as an unsupervised approach. However, the performances of this approach are still far below those of the supervised model from transfer learning (Ranasinghe et al., 2020). Moura et al. (2020) combined limited "glass-box" features with the hidden state of a bottle-neck adapter layer attached on the output from the XLM-Robert, and the results indicate that these features can bring slight but significant improvements to the transfer learning model. Wang et al. (2021) proposed more unsupervised "glass-box" and "black-box" QE features and investigated further on the contributions of each one to the QE model's performance via a feature-enhanced supervised model.

Inspired by their work, we explored deeply in the aspect of uncertainty quantification to obtain uncertainty features in this section to enhance the transfer learning model. First, we extend "glass-box" features in Fomicheva et al. (2020) to the *Decoding Probability Features* and the *Monte Carlo Dropout Features*. And then, the *Noised Data Features* are proposed similar to the *Monte Carlo Dropout Features*.

Decoding Probability Features. For autoregressive sequence generating models like Transformers (Vaswani et al., 2017), the decoding probability at each step can be extracted from the softmax layer directly in a "glass-box" setting:

$$P_{step}^{(\mathbf{x},t,\theta)} = \log P(y_t|\mathbf{y}_{<t}, \mathbf{x}, \theta) \tag{1}$$

where $\mathbf{x}$ represents the input source text and $\mathbf{y}$ is the output machine translation. P_{step} is a probability sequence with the same length of the generated sequence $\mathbf{y}$. Three statistical indicators of P_{step} can be used to estimate the uncertainty of the output: expectation, standard deviation, and the combined ratio of them:

$$\mathbb{E}(P_{step}|\mathbf{x}, \theta) = \frac{1}{T}\sum_{t=1}^{T} P_{step}^{(\mathbf{x},t,\theta)} \tag{2}$$

$$\begin{aligned}&\sigma(P_{step}|\mathbf{x}, \theta)\\ =&\sqrt{\mathbb{E}(P_{step}^2|\mathbf{x}, \theta) - \mathbb{E}^2(P_{step}|\mathbf{x}, \theta)}\end{aligned} \tag{3}$$

$$Combo(P_{step}|\mathbf{x}, \theta) = \frac{\mathbb{E}(P_{step}|\mathbf{x}, \theta)}{\sigma(P_{step}|\mathbf{x}, \theta)} \tag{4}$$

Intuitively, larger expectation, smaller deviation and larger combined ratio of P_{step} indicate lower uncertainty and higher quality. P_{step} is an extended version of the TP feature in Fomicheva et al. (2020) and the expectation of P_{step} is the same as TP.

Monte Carlo Dropout Features. Monte Carlo (MC) Dropout sampling, that has been exploited in Gal and Ghahramani (2016), is an efficient "glass-box" approach to estimate uncertainty. It enables random dropout on neural networks during inference and the predictive probabilities through different sampling paths are used to obtain measures of uncertainty (Fomicheva et al., 2020). The output sequences $\hat{\mathbf{y}}$ sampled across stochastic forward-passes by MC dropout with different sampled model parameters $\hat{\theta}$ can be different as well. If $\mathbf{y}$ is a high-quality output with low uncertainty, the Monte Carlo sampled outputs $\hat{\mathbf{y}}$ should be close to $\mathbf{y}$ and the variance of $\hat{\mathbf{y}}$ should be low. Hence, two measurements of sampling based on text similarity are carried out here:

$$MC\text{-}Sim = Sim(\mathbf{y}, \hat{\mathbf{y}}_i) \tag{5}$$

$$MC\text{-}Sim\text{-}Inner = \frac{1}{N}\sum_{j=1}^{N} Sim(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_j) \tag{6}$$

where $\hat{\mathbf{y}}_i$ is the i-th sample of $\hat{\mathbf{y}}$, and $1 \leq i \leq N$. For the similarity score function, as in Fomicheva et al. (2020), Meteor metric (Denkowski and Lavie, 2014) is applied. Besides, as a sentence-level probability score, $\mathbb{E}(P_{step})$ can also be calculated with different model parameters $\hat{\theta}$ by MC dropout sampling:

$$MC\text{-}P_{step} = \mathbb{E}(P_{step}|\mathbf{x}, \hat{\theta}) \tag{7}$$

The expectation, standard deviation, and combined ratio of $MC\text{-}Sim$, $MC\text{-}Sim\text{-}Inner$ and $MC\text{-}P_{step}$ are calculated over all MC dropout samples and will be used as "glass-box" uncertainty features. Among them, $\mathbb{E}(MC\text{-}P_{step})$, $\sigma(MC\text{-}P_{step})$, $Combo(MC\text{-}P_{step})$, and $\mathbb{E}(MC\text{-}Sim\text{-}Inner)$ are equivalent to $D\text{-}TP$, $D\text{-}Var$, $D\text{-}Combo$, and $D\text{-}Lex\text{-}Sim$ in Fomicheva et al. (2020)

Noised Data Features. Monte Carlo Dropout approaches mentioned above can be regarded as a robustness test of the NMT model. Due to its validity in Fomicheva et al. (2020), it is rational to believe that a similar way with appropriate noise in the input of MT may perform comparably. Therefore, we define the following uncertainty features similar to $MC\text{-}Sim$, $MC\text{-}Sim\text{-}Inner$ and $MC\text{-}P_{step}$. The differences are: (1) the NMT model weights are fixed θ without MC dropout sampling; (2) the model decodes translations $\tilde{\mathbf{y}}$ with a noised input $\tilde{\mathbf{x}}$.

$$Noise\text{-}Sim = Sim(\mathbf{y}, \tilde{\mathbf{y}}_i) \tag{8}$$

$$Noise\text{-}Sim\text{-}Inner = \frac{1}{N}\sum_{j=1}^{N} Sim(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) \tag{9}$$

$$Noise\text{-}P_{step} = \mathbb{E}(P_{step}|\tilde{\mathbf{x}}, \theta) \tag{10}$$

One crucial point in designing this type of features is how to generate noised input $\tilde{\mathbf{x}}$. One solution is

Algorithm 1 Generate Noise Input with "Post-Editing"

Require: input $\mathbf{x} = \{x_t | t = 1, 2, ..., T\}$, hyper-parameters R, p_i, p_d.

1: Initialize $\mathbf{x}_{mask} = \mathbf{x}$
2: **for** $r = 1, ..., R$ **do**
3: $\mathbf{x}_{mask}$ = randomly delete tokens from $\mathbf{x}_{mask}$ with probability p_d
4: $\mathbf{x}_{mask}$ = randomly insert special <*mask*> tokens into $\mathbf{x}_{mask}$ with probability p_i
5: **end for**
6: $\tilde{\mathbf{x}} = MLM(\mathbf{x}_{mask})$, where MLM is a pre-trained masked language model.
7: **return** $\tilde{\mathbf{x}}$

a "black-box" way that takes the advantage of the masking strategy of the pre-trained XLM-Roberta. Basically, we can mask some words in the source text and get a noised source text by the predictions from the pre-trained model in the masked positions. This simple approach only conducts substitutions on $\mathbf{x}$ with the [mask] token, but it limits the diversity of the noised sample inputs. To enrich the variety of $\mathbf{x}$, we adjust the imitation learning algorithm in Wang et al. (2020) to a simplified version to obtain noised input $\tilde{\mathbf{x}}$. We "post-edit" the input $\mathbf{x}$ by randomly deleting tokens and inserting masks for several rounds to get $\mathbf{x}_{mask}$. Then, the pre-trained XLM-R is used as a masked language model to predict the tokens in the masked positions of $\mathbf{x}_{mask}$ to get the post-edited $\tilde{\mathbf{x}}$. Pseudo codes of this "post-editing" algorithm is provided in Algorithm 1.

3.3 Strategies

Multilingual Training. Considering zero-shot language pairs in the DA task, we mix up all seven language pairs' training data to fine-tune the XLM-Roberta model and predict on the whole test set including zero-shot language pairs. We have tried two different ways of mixing up training data from different language pairs to fine-tune XLM-Roberta: (1) source sentence + translation sentence; (2) English sentence + non-English sentence. Our experimental results demonstrate that multilingual models usually perform better than bilingual models trained on a single language pair, but there is no prominent difference in performance of the two different multilingual strategies. We keep both multilingual models and bilingual models for model ensemble.

Data Augmentation. Two data augmentation strategies are applied for the CED task. First, considering the imbalance between positive and negative samples in the CED dataset, we up-sample the data with*ERR* labels in each language pair to obtain a balanced dataset. Secondly, inspired by examples provided by the organizer, we have also tried to replace the original machine translation with a back-translated sentence and hope that the gap between the source sentence and the back-translated sentence can provide insights of the detection of potential critical errors. The back translations come from the released ML50 multilingual translation model (Tang et al., 2020).

Model Ensemble. For the DA task, models trained with different multilingual strategies and different uncertainty features are ensembled by averaging predicted scores. While for the CED task, we average classification probability outputs from models trained with different data augmentation strategies and uncertainty features to obtain ensemble results. We apply a greedy ensemble strategy. First, all models are sorted by their performance on the development sets. Then, upon the best single model, we take one more model into the ensemble at each step until there is no more performance gain on the development sets or the maximum step is reached. We set the maximum step to avoid overfitting on the development sets.

4 Experiments

4.1 Model Settings

We follow the model settings of Transquest (Ranasinghe et al., 2020) to fine-tune our QE model based on the XLM-Roberta large model with a classification/regression head on a single P100 GPU. The training batch size is set to 8 and the training process takes about 2 hours to convergence. For the DA task, the total number of parameters of QE-Mind with uncertainty features is 560981507; if no uncertainty features are used, it is 560941571. And for the CED task, the numbers of parameters with and without uncertainty features are 560982532 and 560942596 respectively.

4.2 Experiments of DA task

We conduct all experiments and evaluate our model on last year's test sets to optimize model configurations for each language pair. In particular, the model performed best on all seven language pairs in average is selected to generate submissions for

Model	High-Resource		Mid-Resource			Low-Resource	
	En-De	En-Zh	Et-En	Ro-En	Ru-En	Si-En	Ne-En
OpenKiwi (Official Baseline)	0.1455	0.1902	0.4770	0.6845	0.5459	0.3737	0.3860
TransQuest Single	0.4669	0.4779	0.7748	0.8982	0.7734	0.6525	0.7914
QEMind-Bi	0.4463	0.4471	0.7569	0.8961	0.7990	0.6443	0.7988
QEMind-Multi	0.5107	0.4762	0.8031	0.9009	0.7984	0.6775	0.7958
QEMind-Multi + UNC	**0.5746**	**0.5094**	**0.8156**	**0.9039**	**0.8044**	**0.6843**	**0.8130**
TransQuest Ensemble	0.5539	0.5373	0.8240	0.9082	0.8082	0.6849	0.8222
QEMind Ensemble	**0.6054**	**0.5445**	**0.8410**	**0.9173**	**0.8273**	**0.7079**	**0.8374**

Table 1: The Pearson's correlation between model predictions and human DA judges on the WMT 2020 QE test sets

Model	High-Resource		Mid-Resource			Low-Resource	
	En-De	En-Zh	Et-En	Ro-En	Ru-En	Si-En	Ne-En
Official Baseline	0.4025	0.5248	0.6601	0.8175	0.6766	0.5127	0.7376
QEMind Single	0.5281	0.5635	0.7909	0.8954	0.7893	0.5769	0.8406
QEMind Ensemble	**0.5666**	**0.6025**	**0.8117**	**0.9082**	**0.8060**	**0.5956**	**0.8667**

Table 2: Pearson's correlations results of 2021 DA task on non-zero-shot language pairs

Model	En-Ja	En-Cs	Km-En	Ps-En
Official Baseline	0.2301	0.3518	0.5623	0.4760
QEMind Single	0.3354	0.5456	0.6509	0.6159
QEMind Ensemble	**0.3589**	**0.5816**	**0.6787**	**0.6474**

Table 3: Pearson's correlations results of 2021 DA task on zero-shot language pairs

Model	En-Cs	En-De	En-Ja	En-Zh
QEMind	0.3915	0.4629	0.2559	0.2629
QEMind + BK	**0.4257**	**0.4914**	0.2471	0.2800
QEMind + UNC	0.4111	0.4859	**0.2606**	**0.2897**
QEMind Ensemble	**0.4864**	**0.5257**	**0.3325**	**0.3587**

Table 4: Matthews correlation results of WMT 2021 CED task on development sets

Model	En-Cs	En-De	En-Ja	En-Zh
Official Baseline	0.3875	0.3974	0.2139	0.1873
QEMind-Single	0.4129	0.4257	0.2139	0.2356
QEMind Ensemble	**0.4539**	**0.4797**	**0.2601**	**0.2777**

Table 5: Matthews correlation results of WMT 2021 CED task on test sets

zero-shot language pairs.

The Pearson's correlations between our model's predictions and the human DA judges (z-standardized mean DA score) are shown in Table 1. *TransQuest Single* and *TransQuest Ensemble* are the best single and ensemble models of Ranasinghe et al. (2020), which is the winner system of last year's DA task. *QEMind-Bi* and *QEMind-Multi* are models without uncertainty features, between which, the difference is that the model is trained on bilingual data or mixed multilingual data. *QEMind-Multi + UNC* is the complete QEMind model enhanced by various uncertainty features described in Section 3.2. Finally, predictions from bilingual models, multilingual models, and uncertainty features enhanced models are ensembled following Section 3.3, marked as *QEMind Ensemble* in the table.

Results on the DA test sets of WMT 2020 show that: (1) multilingual strategies work well on this task, especially for high-resource language pairs; (2) the uncertainty features enhanced multilingual model achieves the highest performance among all single models, which verifies that these uncertainty features are useful to all language pairs and can be fused in multilingual models. (3) ensemble of multiple models of different settings can further improve the performance of QEMind systems.

We pick the best single and ensemble models

for each language pair and produce predictions on the newly released blind test sets of WMT 2021, including the 4 zero-shot language pairs. Results of Pearson's correlations are shown in Table 2 and Table 3.

4.3 Experiments of CED task

We test different strategies and uncertainty features on the CED development sets. Brief results of Matthews correlations (MCC) on development sets are shown in Table 4. All models are trained on up-sampled training data of each language pair. From the results observations, compared to *QEMind*, which only applies up-sampling on the training data, the strategies of back-translation (*QEMind + BK*) and uncertainty features (*QEMind + UNC*) can achieve comparable or better performances. The ensemble of all these models makes a significant improvement. Similar to the DA task, the best single and ensemble models are picked to generate our final submissions. Results on test sets of this year are listed in Table 5.

5 Conclusion

This paper introduces our machine translation quality estimation model, QEMind, for the sentence-level Direct Assessment and Critical Error Detection tasks of WMT 2021. We propose novel features to estimate the uncertainty of machine translations and incorporate them into the transfer learning from the large-scale pre-trained model, XLM-Roberta. Besides, three important strategies are particularly utilized for improving the QE system's performance such as multilingual training, data augmentation and model ensemble. Our system has achieved the first ranking in average Pearson correlation across all languages, including the zero-shot ones in the multilingual DA task of WMT 2021.

Acknowledgements

This work is supported by National Key R&D Program of China (2018YFB1403202).

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, pages 315–321.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6367–6374.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019. Unbabel's participation in the wmt19 translation quality estimation shared task. *arXiv preprint arXiv:1907.10352*.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. 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):1–22.

Joao Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André FT Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Tharindu Ranasinghe, Constantin Orăsan, and Ruslan Mitkov. 2020. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. Association for Computational Linguistics.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

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*.

Ke Wang, Yangbin Shi, Jiayi Wang, Yuqi Zhang, Yu Zhao, and Xiaolin Zheng. 2021. Beyond glass-box features: Uncertainty quantification enhanced quality estimation for neural machine translation.

Ke Wang, Jiayi Wang, Niyu Ge, Yangbin Shi, Yu Zhao, and Kai Fan. 2020. Computer assisted translation with neural quality estimation and auotmatic post-editing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2175–2186.

Direct Exploitation of Attention Weights for Translation Quality Estimation

Lisa Yankovskaya and **Mark Fishel**
Institute of Computer Science
University of Tartu, Estonia
{lisa_y,mark}@tartunlp.ai

Abstract

The paper presents our submission to the WMT2021 Shared Task on Quality Estimation (QE)[1]. We participate in sentence-level predictions of human judgments (Task 1) and post-editing effort (Task 2). We propose a glass-box approach based on attention weights extracted from machine translation systems. In contrast to the previous works, we directly explore attention weight matrices without replacing them with general metrics (like entropy). We show that some of our models can be trained with a small amount of a high-cost labelled data. In the absence of training data our approach still demonstrates a moderate linear correlation, when trained with synthetic data.

1 Introduction

Quality Estimation (QE, Blatz et al., 2004; Specia et al., 2009) is an essential part of the machine translation (MT) pipeline, which estimates the quality of the translation output without relying on any reference.

Unlike the previous year, three QE sentence-level tasks were presented in the WMT2021 Shared Task (Specia et al., 2021). The goal of Task 1 is to predict direct assessments (DA), i.e. human judgments of translation quality (Graham et al., 2015), whereas in Task 2, the task is to estimate the post-editing effort required to obtain a correct translation which is measured by the HTER metric (Snover et al., 2006). The goal of Task 3 is to determine if the translation output contains at least one critical error.

We propose a lightweight glass-box approach that can be applied to Task 1 and Task 2. The approach is based on using the encoder-decoder attention weight matrices as input features for supervised translation quality estimation. Next we describe our approach (Section 2) and evaluate it experimentally (Sections 3–5).

2 Approach

There are several QE models based on attention weights of neural MT systems described earlier (Yankovskaya et al., 2018; Fomicheva et al., 2020a,b). Their main idea is to compute the entropy of encoder-decoder attention weights for each target token and then average these entropies to get a sentence-level metric:

$$Entropy = -\frac{1}{I}\sum_{i=1}^{I}\sum_{j=1}^{J}\alpha_{ji}\log\alpha_{ji},$$

where α represents attention weights, I is the number of target tokens and J is the number of source tokens.

Yankovskaya et al. (2018) work with attention weights extracted from LSTM (Hochreiter and Schmidhuber, 1997) MT systems. As LSTM has only one attention matrix, the approach of computing entropies is straightforward. However, neural MT models based on Transformer (Vaswani et al., 2017) have several layers and heads, so the number of computed entropies equals [Layers $\times$ Heads] for each sentence, which introduces some difficulties in this approach. To overcome it, Fomicheva et al. (2020a) summarise entropies by taking the average or minimum value to get an unsupervised attention-based QE metric. Fomicheva et al. (2020b) use the obtained entropies as features of regression models.

In this article, we propose another approach of using attention weights obtained from Transformer MT models: instead of summarising them into one metric, we feed all encoder-decoder attentions weights into a convolutional neural network (CNN) to get a QE score. We test the approach in a supervised setting, and also show that it can be applied in a zero-shot scenario when training data for the required language pair is not available.

[1] http://www.statmt.org/wmt21/quality-estimation-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 955–960
November 10–11, 2021.

3 Data

This year, three sentence-level tasks are available: predicting human judgments, post-editing effort and critical errors. In this work, we have focused on the first two tasks.

`Task 1` and `2` include eleven language pairs, seven of which have training (7 000 sentences), development (1 000 sentences) and two test sets (WMT2020 and WMT2021, 1 000 sentences each). For the other four languages only test sets (1 000 sentences) are available, which is called the zero-shot subtask. WMT2020 test set includes gold-labels whereas WMT2021 is the usual blind test without labels before submission.

To test our approach, we have focused on two language pairs with training data: English-German (En-De) and Estonian-English (Et-En), as well as one language pair without training data: English-Czech (En-Cz).

Besides data provided by the shared task organizers, we used additional parallel corpora to train CNN networks: the OpenSubtitles (Lison and Tiedemann, 2016), JRC Acquis (Steinberger et al., 2006), EuroParl (Koehn, 2005), DGT and EMEA (Tiedemann, 2012) corpora.

4 Settings

To compare the performance of our approach with CNN models to a previous baseline, we also ran experiments with models based on machine learning algorithms with entropies as input features (ML-Ent).

Below we present the experimental settings which we used for training ML-Ent and CNN models.

4.1 Machine Learning models with entropies as input features (ML-Ent)

There are two machine learning methods that we used. Random Forest (Ho, 1995) was chosen as a relatively easy and fast approach. We used the `sklearn`[2] library, set a randomized search on the hyperparameters and performed 5-fold cross-validation.

The second method is ensemble building based on (Caruana et al., 2004). The main idea behind the method is doing a greedy search over all trained models to find such models that would improve the ensemble's performance when added. We used the `mljar`[3] library, Random Forest and CatBoost (Prokhorenkova et al., 2018) algorithms, set Pearson as the evaluation metric and ran 5-fold cross-validation.

For both models and both tasks, we combined the proposed training and development sets (8 000 sentences in total) and used `[Heads × Layers]` (in our case 48) entropies for each translation as input.

4.2 CNN-based models

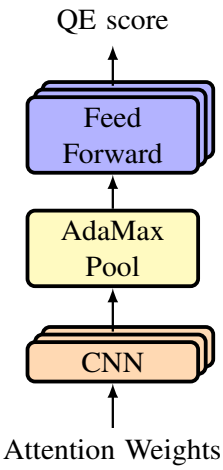


Figure 1: The proposed architecture of the QE model.

The base architecture of proposed CNN models is presented on Figure 1. The model's input is attention weights with shape (`[Heads × Layers]`, number of the source tokens, number of the target tokens). The number of `[Heads × Layers]`[4] is constant for all weights obtained from the same system, whereas the number of source and target tokens of each sentence can vary noticeably. To reduce the amount of padding added to each batch, we sort all sentences by the number of source/target tokens (max(src, tgt)) and only after that form a batch. Each CNN-based model consists of two or three CNN blocks, each of them comprises 2D-CNN, Batch Normalization, MaxPooling and Dropout Layers. We use `Relu` as the activation function. To handle the variable size of input batches, we use the Adaptive Max pooling layer. The last block of the model consists of three feed-forward layers. As a result, the model is trained to produce the desirable score: DA or HTER. We optimised our neural models with Adam (Kingma

[2] `https://scikit-learn.org/stable`

[3] `https://supervised.mljar.com/`

[4] 8 × 6 for En-Et and En-De, and 16 × 12 for En-Cs NMT systems

	En-De		Et-En		En-Cs
	wmt20	wmt21	wmt20	wmt21	wmt21
ML-Ent-RF	0.373	0.301	0.499	0.455	
ML-Ent-Ensemble	**0.395**	0.341	0.517	0.48	
CNN-DA	0.22	0.21	0.518	0.464	
CNN-BLEURT	0.383	0.357	0.577	0.526	0.299
CNN-BLEURT+	0.381	**0.369**	**0.599**	**0.547**	

Table 1: Pearson correlation coefficients between human DA scores and predicted values for WMT2020 and WMT2021 test sets (`Task 1`).

	En-De		Et-En	
	wmt20	wmt21	wmt20	wmt21
ML-Ent-RF	0.389	0.519	0.505	0.534
ML-Ent-Ensemble	0.408	**0.531**	0.519	**0.561**
CNN-HTER	**0.430**	0.503	**0.580**	0.549
CNN-HTERart	0.334	—	0.482	—

Table 2: Pearson correlation coefficients between HTER scores and predicted values for WMT2020 and WMT2021 test sets (`Task 2`).

and Ba, 2015).

`Task 1`**:** To predict DA scores, we considered three models with different training sets:

CNN-DA: we use human-labelled data provided by the shared task organizers: 7 000 for training set and 1 000 for development set;

CNN-BLEURT: we experiment with pre-training on synthetic data and for that we compute the BLEURT (Sellam et al., 2020) score for randomly chosen 300 000 sentences and use them as labels for training and development tests. We have chosen BLEURT to get artificial labels due to its good agreement with human judgments (Mathur et al., 2020);

CNN-BLEURT+: we fine-tune the model **CNN-BLEURT** on data provided by the organizers.

`Task 2` evaluates the proposed QE models for post-editing purposes.

CNN-HTER: we train a model with data provided by the shared task organizers;

CNN-HTERart: we use synthetically computed HTER between translation and reference. Though the preliminary experiments showed a poor performance compared to **CNN-HTER**, but this setting might be used in the absence of the human annotated training data.

5 Results

Below we present the obtained results and discuss the most interesting observations. To assess the performance of sentence-level QE models, Pearson correlation coefficient is used.

Table 1 shows results for the `Task 1`. For Et-En language pair, both CNN-BLEURT models show better results compared to ML-Ent models and CNN model trained only on DA score. For En-De, results are mixed. The CNN-DA model shows abysmal performance compared to both CNN-BLEURT and ML-Ent models. In contrast to Et-En, we can see that the performance of CNN-BLEURT and ML-Ent models is comparable.

Results for `zero-shot` En-Cs are not impressive (Table 1). One of the possible reasons for that is not using enough synthetic training data: while there are 300 000 examples for experiments with En-De and Et-En, we only use 50 000 for En-Cs.

The essential advantage of using CNN-BLEURT models is that they might be used for `zero-shot` settings when a training dataset is not available. However, the building and tuning of the neural network is not an easy task compared to ML-Ent models. The benefits of last ones are relatively fast training and fewer parameters that need to be tuned.

Table 2 presents results for the `Task 2`. We can see that for both languages, the results of ML-Ent and CNN-HTER models are pretty similar and

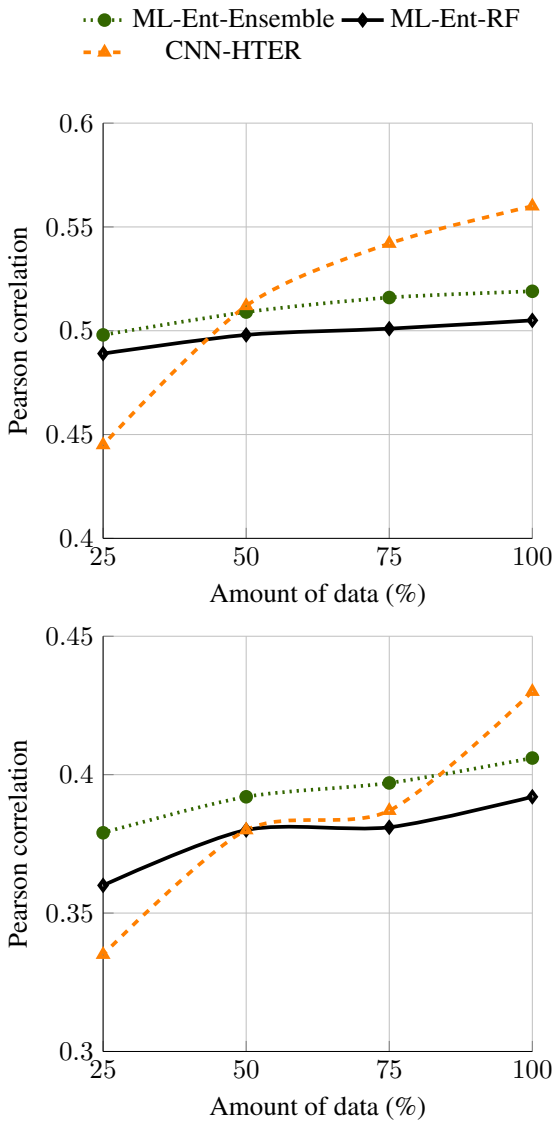


Figure 2: Pearson correlation coefficient between predicted values of WMT2020 test set and HTER scores for Et-En (top) and En-De (bottom) language pairs.

show a moderate correlation. As we mentioned in the previous chapter, the performance of CNN-HTERart is not as good as CNN-HTER, that is why we focus in this chapter only on CNN-HTER model.

Features of ML-Ent models are identical across DA- and HTER-models as well as CNN-DA and CNN-HTER share the same input features. Computed correlation coefficients of HTER- and DA-models are comparable for Et-En language pair. Nevertheless, we see a completely different picture for En-De: coefficients of DA-models are noticeably lower compared to HTER-models. As discussed in (Fomicheva et al., 2020a), the low results of DA models for En-De language pair might be caused by highly-skewed distribution of DA scores, as most translations have high quality scores.

Getting DA scores as well as HTER scores is a time-consuming and expensive task, so the less data you need, the better. To examine how labelled data we need to train models, we ran 10 tests for each examined amount of data (25%, 50%, 75%) and averaged the obtained correlation coefficients. According to our experiments, both discussed approaches, ML-Ent and CNN-HTER/DA, show comparable high performance even with a small amount of training/validation data. As Figure 2 shows, the performance of ML-Ent models for Et-En (top) and En-De (bottom) language pairs slightly worsens with decreased amounts of training data. The performance of CNN-HTER models decreases more noticeably, but still remains quite high. Especially in case of the En-Et language pair, all models demonstrate a moderate linear correlation with post-editing effort even with using 2000 training/validation examples (1750 for training and 250 for validation).

Raganato et al. (2018); Voita et al. (2019) showed that different layers play different roles in the attention mechanism. To examine it from the QE point of view, we compared CNN-HTER models with attention weights extracted from the first three layers, the last three layers and all six layers. According to Table 3, the performance of

	Et-En	En-De
all layers	0.580	0.43
first 3 layers	0.490	0.136
last 3 layers	0.536	0.43

Table 3: Pearson correlation coefficients between predicted values of WMT2020 test set and HTER scores. Results of three settings of CNN-HTER model are presented: with attention weights obtained (1) from all layers, (2) from the first three layers and (3) from the last three layers.

the models with last layers is comparable to the "all layers" models, whereas the difference between models with first layers and "all layers" models is more noticeable. While the performance gap between different models is not so noticeable for Et-En, then for En-De the difference is significant and even more, the lower layers do not provide any "useful" information to the model.

6 Conclusions

We presented sentence-level quality estimation models based on attention weights. The proposed models demonstrated a moderate linear correlation with human judgments as well as with required post-editing effort. The described models can be used as a cost-effective and light-weight QE approach in the machine translation pipeline. Results of empirical evaluation show a good performance even with a small amount of training data, as well as moderate performance in the absence of training data ("zero-shot" settings).

Acknowledgements

Lisa Yankovskaya and Mark Fishel were supported by funding from the Bergamot project (EU H2020 Grant No. 825303). The authors also thank the Unversity of Tartu's High-Performance Computing Center for providing the computing infrastructure (University of Tartu, 2018).

References

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.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. 2004. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18.

Marina Fomicheva, Shuo Sun, Frédéric Blain Lisa Yankovskaya, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020a. Unsupervised quality estimation for neural machine translation.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020b. Bergamot-latte submissions for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017.

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.

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.

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

Diederik P Kingma and Lei Ba. 2015. J. adam: a method for stochastic optimization. In *International Conference on Learning Representations*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Alessandro Raganato, Jörg Tiedemann, et al. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

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, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

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.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and

Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

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).

University of Tartu. 2018. Ut rocket cluster, https://doi.org/10.23673/ph6n-0144.

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.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Elizaveta Yankovskaya, Andre Tättar, and Mark Fishel. 2018. Quality estimation with force-decoded attention and cross-lingual embeddings. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 816–821.

IST-Unbabel 2021 Submission for the Quality Estimation Shared Task

Chrysoula Zerva[1,2,*] **Daan van Stigt**[3,*] **Ricardo Rei**[2,3,4,*] **Ana C. Farinha**[3]
Pedro G. Ramos[3] **José G. C. de Souza**[3] **Taisiya Glushkova**[1,2] **Miguel Vera**[3]
Fábio Kepler[3] **André F. T. Martins**[1,2,3]
[1]Instituto Superior Técnico [2]Instituto de Telecomunicações [3]Unbabel [4]INESC-ID
[2,3,4]Lisbon, Portugal
{chrysoula.zerva, ricardo.rei, taisiya.glushkova, andre.t.martins}@tecnico.ulisboa.pt
{daan.stigt, catarina.farinha, pedro.ramos, jose.souza, miguel.vera, fabio.kepler}@unbabel.com

Abstract

We present the joint contribution of IST and Unbabel to the WMT 2021 Shared Task on Quality Estimation. Our team participated on two tasks: Direct Assessment and Post-Editing Effort, encompassing a total of 35 submissions. For all submissions, our efforts focused on training multilingual models on top of `OpenKiwi` predictor-estimator architecture, using pre-trained multilingual encoders combined with adapters. We further experiment with and uncertainty-related objectives and features as well as training on out-of-domain direct assessment data.

1 Introduction

Quality estimation (QE) is the task of evaluating a translation system's quality without access to reference translations (Blatz et al., 2004; Specia et al., 2018). This paper describes the joint contribution of Instituto Superior Técnico (IST) and Unbabel to the WMT21 Quality Estimation shared task (Specia et al., 2021), where systems were submitted to two tasks: 1) sentence-level direct assessment; 2) word- and sentence-level post-editing effort.

This year's submission combines several ideas built on top of the `OpenKiwi` framework. Motivated by the mixture of *blind* and *seen* language pairs in the test sets, we experimented with extensions that would allow us to train multilingual models that maintain good generalization ability and are robust to the presence of epistemic and aleatoric uncertainty.

For both tasks we trained and submitted an ensemble of multilingual models. All submitted models follow the predictor-estimator architecture (Kim and Lee, 2016; Kim et al., 2017) and use pre-trained models for feature extraction. Also, we fine-tune all models on the provided QE data using stacked adapter layers (Pfeiffer et al., 2020). We show that we can thus achieve comparable performance across language pairs while minimising the number of trainable parameters (see Table 1). Furthermore, we experimented with different types of uncertainty-related information to leverage it's benefits, improving performance and robustness of the submitted systems (see §3.1.1). All related code extensions will be publicly available.

Our main contributions are:

- We build on our `OpenKiwi` architecture by exploring adapter layers (Houlsby et al., 2019; Pfeiffer et al., 2020) for quality estimation as these demonstrated to be less amenable to overfitting while presenting the same or superior quality performance than fine-tuning the whole base pre-trained model for different NLP tasks (He et al., 2021).
- We incorporate different types of uncertainty into our architectures. We make use of the glass-box features (Fomicheva et al., 2020) extracted from the NMT models, the *aleatoric* (data) uncertainty derived from the human annotations and the *epistemic* (model) uncertainty (Hora, 1996; Kiureghian and Ditlevsen, 2009; Huellermeier and Waegeman, 2021) that originates from the QE model.
- We show that training the QE models on additional out-of-domain direct assessment (DA) data gives considerable gains in performance for the new language pairs from the *blind* test sets.

2 Quality Estimation Tasks

In this year's shared task edition we submitted models for the first two tasks:

1. Task 1: sentence-level direct assessment
2. Task 2: word- and sentence level post-editing effort, comprising of two subtasks: a) predicting the HTER score of the translated sentence

∗ The first three authors have equal contribution.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 961–972
November 10–11, 2021.

(hypothesis); and b) predicting `OK/BAD` tags for the words and gaps (both in source and translation)

We note that this year, both tasks 1 and 2 provided additional *blind* test sets with language pairs that were not included in the data made available for training/development, providing an interesting challenge and motivating multilingual and generalisable approaches.

3 Implemented Systems

3.1 Task 1

For Task 1 our final submission consisted of an ensemble of two different multilingual models, that differ in the way they process the input source (original sentence) and hypothesis (machine translation). Both models are based on the predictor-estimator architecture, using different pre-trained models to extract features and different training approaches to optimise for the QE task.

The key idea explored with our first model (denoted by M1 variations in the experiments), revolved around pursuing highly generalisable multilingual models, robust to overfitting. To this end, we train a cross-lingual transformer (XLM-RoBERTa (Conneau et al., 2020)) on large, multilingual data with direct assessments and then use adapters (Houlsby et al., 2019; Pfeiffer et al., 2020) to adapt to the domain specific data of the QE task with minimal training effort. In line with our efforts for good generalisation, we use only task-specific adapters and refrain from using specific adapters for each language pair. For these experiments we build on the `OpenKiwi` architecture (Kepler et al., 2019), using a pre-trained `xlm-roberta-large` encoder as a feature predictor. The source and hypothesis sentences are jointly encoded with hypothesis first. Then, source and hypothesis features are generated using average pooling over the hypothesis embeddings and forwarded to the estimator module which corresponds to a feed-forward layer. Figure 1 provides the general architecture[1]

The model was first trained on the direct assessment data provided in the Metrics shared tasks (Mathur et al., 2020), as described in §3.1.2. Upon training, the XML-R encoder is frozen and the the model is fine-tuned on sentence regression with

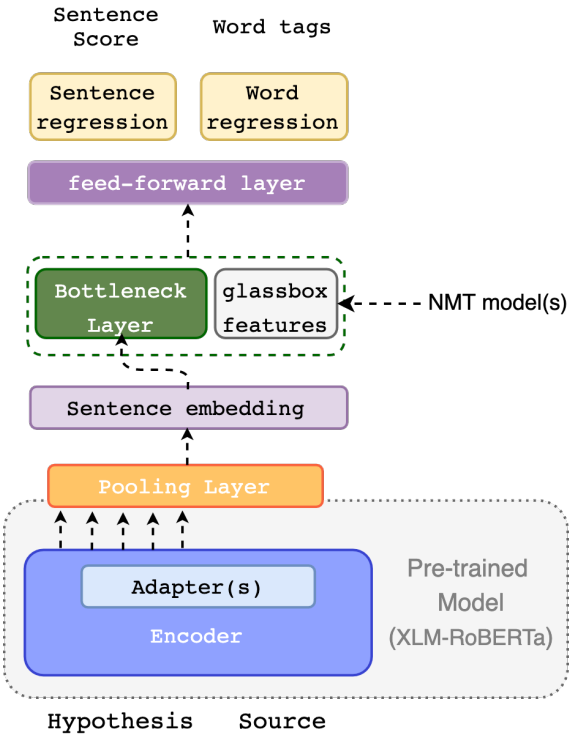


Figure 1: General architecture of M1 model variations. Word tag prediction is used only for Task 2.

the task-specific data, using stacked adapters. We hence manage to maintain a low number of trainable parameters during fine-tuning and minimize training time while learning to predict task-specific sentence scores.

For the second model (denoted by M2-KL-G-MCD) we aimed to explore the potential of a large pre-trained multilingual model (trained with MT objectives). We use the mBART (Liu et al., 2020) encoder-decoder architecture to encode the source and force-decode the hypothesis. We specifically use the mBART50 model (Tang et al., 2020) which is trained with multilingual finetuning on 50 languages, including all languages of interest for the QE 2021 task. We obtain the features by averaging the decoder embeddings and concatenating with the `<eos>` token of the sequence. The estimator part of the model consists of a *bottleneck* feed-forward layer that reduces the dimensionality of the decoder output, and is concatenated with a vector with additional glass-box features from the NMT models (see §3.1.1). The combined vector is then forwarded to a feed-forward estimator and the full model is fine-tuned on the task specific QE data. Apart from the glass-box features we experimented further with methods that allow the model to be

[1]Note that glass-box features are integrated but not used in this submission as they did not significantly improve performance.

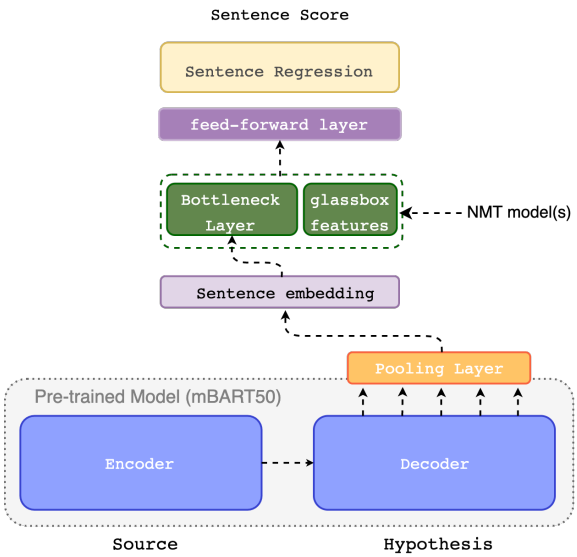


Figure 2: General architecture of M2 model variations.

more robust towards the underlying uncertainty of its predictions. We elaborate that in the next section. Figure 2 provides a general architecture of the M2 model variations.

3.1.1 Learning from uncertainty

Multiple neural models are involved in the process of obtaining and scoring machine translations, which naturally leads to several sources of uncertainty. These sources can be very informative and useful for MT evaluation. In this work we try to consider three types of uncertainty: (1) uncertainty of the NMT models used to obtain the *hypotheses*, (2) data (aleatoric) uncertainty for which we use the inter-annotator disagreement as a proxy, and (3) uncertainty of the MT evaluation model itself.

NMT model uncertainty The idea of extracting uncertainty-related features from the MT systems in order to estimate the quality of their predictions, was originally introduced by Fomicheva et al. (2020). This glass-box approach to QE is mostly focusing on capturing epistemic uncertainty, and the proposed features are extracted either using Monte Carlo (MC) dropout on the NMT or using the output probability distributions obtained from a standard deterministic MT system. In our last year's submission (Moura et al., 2020) the integration of such features proved to be effective, thus we decided to incorporate it into our new model as well. We list the extracted features below:

- `TP` sentence average of word translation probability
- `Softmax-Ent` sentence average of softmax output distribution entropy
- `Sent-Std` sentence standard deviation of word probabilities
- `D-TP` average TP across N(N = 30) stochastic forward-passes
- `D-Var` variance of TP across N stochastic forward-passes
- `D-Combo` combination of D-TP and D-Var defined by $1 - D - TP/D - Var$
- `D-Lex-Sim` lexical similarity - measured by METEOR score (Lavie and Denkowski, 2009) - of MT output generated in different stochastic passes.

Aleatoric uncertainty The noise and complexity of the training data is a source of predictive uncertainty in itself, referred to as data or aleatoric uncertainty (Kiureghian and Ditlevsen, 2009). This uncertainty is often reflected in the disagreement between human annotations for the same *source-hypothesis* segment (Cohn and Specia, 2013; Fornaciari et al., 2021). We hypothesize that the direct assessments can be better modelled as normally distributed scores rather than a single score, and that a model trained to predict this distribution (mean and standard deviation) could provide better quality estimates [2]. We formalise this as a KL divergence objective, using the closed form solution to estimate the KL divergence between the target distribution $p(x) = N(\mu_1, \sigma_1)$ and the predicted distribution $q(x) = N(\mu_2, \sigma_2)$, as shown in Eq. 1.

$$KL(p||q) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \quad (1)$$

where we take the mean and standard deviation (std) of the direct assessment z_scores as the target (ground truth proxy) values p. This way, we account for the annotator disagreement (reflected in the std value) during learning.

QE epistemic uncertainty We use MC dropout (Gal and Ghahramani, 2016) to account for the uncertainty of the QE model. Specifically, we enable dropout during inference and run multiple forward runs over each test instance. Thus we obtain a distribution of quality predictions for each instance

[2]Note that for this task's data we only had access to 3 scores per segment so the mean and std values are calculated over these numbers.

instead of a single point estimate. We use the estimated mean of the distribution as our predicted quality estimate. MC dropout has been shown to improve predictive accuracy and perform on par or even better compared to deep ensembles for MT evaluation tasks (Glushkova et al., 2021). It thus allows us to simulate ensembling in a cheap and effective way, without the need to train multiple checkpoints.

3.1.2 Out-of-domain direct assessment data

The QE data is relatively limited, making it harder to train multilingual models with a large number of parameters without over-fitting. Thus, as explained in §3.1 we aimed to investigate whether we could obtain models that generalise better and are more robust to noise and out-of-distribution data by training the XLM-RoBERTa model first on a larger –yet noisier and out-of-domain dataset. To that end we leverage the data provided for the past Metrics shared tasks, which covers the language pairs used in this year's QE task, including the blind tests for which we had no in-domain data available. Altogether, it encompasses 30 language pairs from the news domain (versus 7 in the QE dataset). We provide more detailed statistics for each language pair of the Metrics data in Appendix C. We refer to experiments using the model initially trained on the Metrics data as M1M-. We also show that using the trained XLM-RoBERTa encoder from the M1M model can prove beneficial for the predictions on post-edited data of Task 2 (see Table 3).

3.2 Task 2

For Task 2 we submitted an ensemble of two variations of the first model (M1-ADAPT and M1M-ADAPT) presented for Task 1 (see §3.1). In both cases, we use multi-task training and a feed-forward for each output types: hypothesis word tags, hypothesis gap tags, source word tags, and sentence regression (on HTER scores). Both variations use a pre-trained XLM-RoBERTa (`large`) encoder to extract features as described for Task 1, but differ in the training of the encoder. In the first case we use the pre-trained model [3] and fine-tune on the QE data using stacked adapters. In the second variation we swap the original pre-trained model with the XLM-RoBERTa model that has been trained on the Metrics data as described in §3.1.2. We note that the two variations favor different language pairs, hence we combine multiple checkpoints from each variation (ranging training steps). We use the `test-20` split of the data to optimise the hyper-parameters and following this approach we use the estimated top-3 checkpoints from each variation using the combined dataset [4] and the top checkpoint for the non-augmented model trained exclusively on the train set, resulting in total 7 checkpoints in our final ensemble.

4 Experimental Results

We present the performance of the implemented models on the `test-20` dataset.

4.1 Task 1

The results can be seen in Tables 1 and 2. In line with the shared task guidelines we treat Pearson r as the primary performance metric and select the submitted models accordingly. We can observe, that while on average the M1 model and its variations outperform the M2 model, their performance is comparable, and M2-KL-G-MCD can even outperform M1M-ADAPT for specific language pairs, hence it made sense to combine them in the final ensemble. We can also see that fine-tuning the M1 model on the Metrics data, results in performance gains for the majority of the language pairs. Specifically, even applying the M1M directly, without further fine-tuning on QE data, achieves competitive performance for most pairs, which further improves upon fine-tuning. It helps in increasing the performance on the *blind* sets (denoted as *zero-shot* in the Appendix B). The performance gains concern mostly the correlation performance indicators (Pearson and Spearman correlations), since especially for M1 the error-based indicators (MAE and RMSE) seem to favor the versions of the model that have not seen the Metrics data. One possible explanation for this discrepancy could lie in the differences between the range and distribution of DA scores for the two datasets. Indicatively, the range of scores on the `train-dev-test-20` concatenation of the QE data is $[-7.542, 3.178]$ and for the Metrics data $[-8.624, 4.332]$. The target DA scores in both datasets are calculated via standardizing (taking the z score) the direct assessments for each annotator and then averaging all standardized

[3] https://huggingface.co/transformers/model_doc/xlmroberta.html

[4] The combined dataset in this case refers to the concatenation of the train/dev/test20 annotated data splits provided for the shared task

		Pears↑	Spear↑	MAE↓	RMSE↓
EN-DE	M1 BASE	0.4534	0.4532	0.4482	0.6371
	M1-ADAPT	0.5092	0.4825	0.4868	0.6288
	M1M	0.5288	0.4872	0.4485	0.6327
	M1M-ADAPT	0.5695	0.5131	0.4127	0.6095
EN-ZH	M1 BASE	0.4429	0.4362	0.5364	0.6867
	M1-ADAPT	0.4723	0.4755	0.5228	0.6714
	M1M	0.4447	0.4400	0.4772	0.6110
	M1M-ADAPT	0.4815	0.4872	0.5502	0.7017
ET-EN	M1 BASE	0.7939	0.8076	0.5388	0.6928
	M1-ADAPT	0.7948	0.8061	0.4518	0.5810
	M1M	0.7580	0.7611	0.5820	0.7134
	M1M-ADAPT	0.7956	0.8110	0.5358	0.6921
NE-EN	M1 BASE	0.7805	0.7592	0.4278	0.5461
	M1-ADAPT	0.7609	0.7475	0.4075	0.5393
	M1M	0.7477	0.7324	0.4499	0.6161
	M1M-ADAPT	0.7888	0.7556	0.4192	0.5332
RO-EN	M1 BASE	0.8718	0.8360	0.3598	0.4878
	M1-ADAPT	0.8923	0.8533	0.3068	0.4201
	M1M	0.8345	0.8132	0.4585	0.5863
	M1M-ADAPT	0.8889	0.8488	0.3142	0.4437
RU-EN	M1 BASE	0.7587	0.6919	0.4885	0.6949
	M1-ADAPT	0.7736	0.7142	0.4138	0.6082
	M1M	0.6703	0.6535	0.5606	0.7583
	M1M-ADAPT	0.7425	0.7159	0.4989	0.7250
SI-EN	M1 BASE	0.6456	0.6112	0.5060	0.6481
	M1-ADAPT	0.6613	0.6172	0.4742	0.5939
	M1M	0.6308	0.6535	0.4742	0.5786
	M1M-ADAPT	0.6649	0.6225	0.4863	0.6064
ML	M1 BASE	0.6781	0.6565	0.4722	0.6276
	M1-ADAPT	0.6949	0.6709	0.4377	0.5775
	M1M	0.6593	0.5131	0.4127	0.6095
	M1M-ADAPT	**0.7045**	**0.6791**	**0.4596**	**0.6160**

Table 1: Results for Task 1 with the M1 predictor-estimator (XLM-RoBERTa) and different training/fine-tuning approaches. M1M is the M1 model trained on the Metrics dataset and M#-ADAPT signifies a model fine-tuned on the QE data with adapters. ML stands for MULTILINGUAL, showing the performance averaged over all language pairs. Underlined numbers indicate the best result for each language pair and evaluation metric. **Bold** systems were selected for the final ensemble.

		Pears↑	Spear↑	MAE↓	RMSE↓
EN-DE	M2 BASE	0.4889	0.4645	0.4608	0.6180
	M2-KL	0.4971	0.4769	0.4549	0.6191
	M2-KL-G	0.5110	0.4738	0.4396	0.6133
	M2-KL-G-MCD	0.5093	0.4754	0.4495	0.6128
EN-ZH	M2 BASE	0.4484	0.4355	0.4940	0.6374
	M2-KL	0.4574	0.4471	0.5042	0.6485
	M2-KL-G	0.4566	0.4543	0.5278	0.6751
	M2-KL-G-MCD	0.4628	0.4584	0.4973	0.6390
ET-EN	M2 BASE	0.7792	0.7842	0.4581	0.5624
	M2-KL	0.7833	0.7896	0.4684	0.5824
	M2-KL-G	0.7847	0.7962	0.4643	0.5924
	M2-KL-G-MCD	0.7868	0.7951	0.4539	0.5674
NE-EN	M2 BASE	0.7333	0.7154	0.4347	0.5531
	M2-KL	0.7638	0.7393	0.4040	0.5247
	M2-KL-G	0.7529	0.7228	0.4194	0.5353
	M2-KL-G-MCD	0.7596	0.7269	0.4125	0.5313
RO-EN	M2 BASE	0.8780	0.8407	0.3403	0.4514
	M2-KL	0.8826	0.8406	0.3199	0.4305
	M2-KL-G	0.8728	0.8397	0.3314	0.4635
	M2-KL-G-MCD	0.8777	0.8429	0.3209	0.4426
RU-EN	M2 BASE	0.7406	0.6874	0.4696	0.6381
	M2-KL	0.7532	0.7123	0.4558	0.6299
	M2-KL-G	0.7485	0.7191	0.4630	0.6612
	M2-KL-G-MCD	0.7509	0.7204	0.4492	0.6358
SI-EN	M2 BASE	0.6243	0.5899	0.4709	0.5939
	M2-KL	0.6373	0.6000	0.4572	0.5726
	M2-KL-G	0.6506	0.6168	0.4586	0.5796
	M2-KL-G-MCD	0.6545	0.6199	0.4495	0.5697
ML	M2 BASE	0.6704	0.6454	0.4469	0.5792
	M2-KL	0.6821	0.6580	0.4378	0.5725
	M2-KL-G	0.6825	0.6604	0.4434	0.5886
	M2-KL-G-MCD	**0.6859**	**0.6627**	**0.4333**	**0.5712**

Table 2: Results for Task 1 with the M2 predictor-estimator (mBART) and different uncertainty handling additions. "KL" signifies the incorporation of KL loss, "G"the incorporation of glass-box features and MCD the addition of MC dropout. ML stands for MULTILINGUAL, showing the performance averaged over all language pairs. Underlined numbers indicate the best result for each language pair and evaluation metric. **Bold** systems were selected for the final ensemble.

assessments for each segment. Thus, the difference in target score range and distribution could affect the magnitude of predicted scores and the distance to the ground truth values, which is reflected in the MAE and RMSE metrics. These findings, further supported by the results on Task 2, is a first step in exploring the underlying connection and bridging the gap between the Metrics and Quality Estimation shared tasks.

4.2 Task 2

The results can be seen in Table 3. Similarly to Task 1, the primary evaluation metric for the sentence level sub-task of Task 2 is the Pearson r coefficient, while the word level sub-task is evaluated using the Matthews correlation coefficient (MCC, (Matthews, 1975)) as the primary performance indicator.

We can see that while HTER scores do not always correlate highly with DAs (see Table 4), the use of the M1M model encoder that was trained on large data with direct assessments can still prove useful. Indeed, when fine-tuning on the Task2 data, the model using the M1M encoder (M1M-ADAPT in the table 3) provides a performance boost for the Pearson correlation in most language pairs, and competitive performance for the rest. Based on these results, we deem it worthwhile to include checkpoints trained with this configuration in the ensemble estimating that they will contribute in higher performance, especially on the blind test sets. This can be further confirmed when

		Pearson↑	SRC-MCC↑	TGT-MCC↑
EN-DE	M1 BASE	0.5256	0.3331	0.4092
	M1-ADAPT	0.5573	0.4211	0.36454
	M1M-ADAPT	0.5499	0.3647	0.4239
EN-ZH	M1 BASE	0.3786	0.3253	0.3589
	M1-ADAPT	0.3711	0.4346	0.3288
	M1M-ADAPT	0.3721	0.4255	0.3643
ET-EN	M1 BASE	0.7319	0.4537	0.5110
	M1-ADAPT	0.7360	0.5545	0.4978
	M1M-ADAPT	0.7498	0.4929	0.5513
NE-EN	M1 BASE	0.5898	0.5198	0.4386
	M1-ADAPT	0.5987	0.6884	0.5426
	M1M-ADAPT	0.6252	0.4244	0.4682
RO-EN	M1 BASE	0.8531	0.5727	0.6190
	M1-ADAPT	0.8282	0.5984	0.5653
	M1M-ADAPT	0.8280	0.5682	0.5813
RU-EN	M1 BASE	0.4899	0.2766	0.3213
	M1-ADAPT	0.4811	0.341	0.3071
	M1M-ADAPT	0.5060	0.2927	0.3421
SI-EN	M1 BASE	0.6659	0.4653	0.4776
	M1-ADAPT	0.6698	0.6776	0.5057
	M1M-ADAPT	0.6935	0.3872	0.4937
ML	M1 BASE	0.6050	0.4209	0.4479
	M1-ADAPT	**0.6061**	**0.5323**	**0.4445**
	M1M ADAPT	**0.6178**	**0.4222**	**0.4607**

Table 3: Results for Task 2 with the M1 predictor-estimator (XLM-RoBERTa) and different training/fine-tuning approaches. M1M is the M1 model trained on the Metrics dataset and M#-ADAPT signifies a model fine-tuned on the QE data with adapters. ML stands for MULTILINGUAL, showing the performance averaged over all language pairs. Underlined numbers indicate the best result for each language pair and evaluation metric. **Bold** systems were selected for the final ensemble.

inspecting the results for the blind sets (`en-cs`, `en-ja`, `km-en` and `ps-en`) in the official results on `test-21` as shown in Appendix B.

lp	TRAIN	DEV	TEST-20
EN-DE	-0.1654	-0.4032	-0.3850
EN-ZH	-0.2947	-0.1895	-0.1932
ET-EN	-0.5464	-0.5850	-0.5995
NE-EN	-0.4527	-0.5004	-0.4558
RO-EN	-0.5887	-0.7932	-0.7880
RU-EN	-0.5358	-0.5055	-0.5152
SI-EN	-0.3916	-0.4384	-0.4125

Table 4: Pearson correlation between the z_mean of the direct assessments for the QE Task 1 data and the HTER score for the post edits in QE Task 2 data.

5 Conclusions

We presented a joint contribution of IST and Unbabel to the WMT 2021 QE shared task. Our submissions are ensembles of multilingual checkpoints extending the `OpenKiwi` framework. We found adapter-tuning to be suitable for fine-tuning OpenKiwi on the QE tasks data and less prone to overfitting. We showed that pre-training on large, out-of-domain annotated data can prove beneficial both for the direct assessment and the post-editing QE tasks. We also demonstrated that handling uncertainty-related sources of information improves the performance when integrated into the QE system. For Task 2 we do multi-task training based on the models from the previous task and use multiple checkpoints to create the submitted ensemble.

Acknowledgements

We are grateful to Alon Lavie and Craig Stewart for their valuable feedback and discussions. This work was supported by the P2020 programs MAIA (contract 045909) and Unbabel4EU (contract 042671), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task Gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32–42, Sofia, Bulgaria. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-Aware Machine Translation Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Online. Association for Computational Linguistics.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online. Association for Computational Linguistics.

Stephen C. Hora. 1996. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2):217–223. Treatment of Aleatory and Epistemic Uncertainty.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Eyke Huellermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning : an introduction to concepts and methods. *Machine Learning*, 110(3):457–506.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim and Jong-Hyeok Lee. 2016. A recurrent neural networks approach for estimating the quality of machine translation output. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 494–498.

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*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112. Risk Acceptance and Risk Communication.

Alon Lavie and Michael Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23:105–115.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. IST-unbabel participation in the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

A Hyperparameters

A.1 M1

In Table 5 is an excerpt of the training configuration used for training OpenKiwi for our M1 models. Note that the configurations follow the configuration file format of OpenKiwi and any additional configurations are identical to the ones proposed in the sample configuration file of the `github` repository[5].

System	
batch_size	2
Encoder	
hidden_size	1024
Decoder	
bottleneck_size	1024
dropout	0.05
hidden_size	1024
Optimizer	
class_name	adam
encoder_learning_rate	0.0001
learning_rate_decay	1.0
learning_rate_decay_start	0
learning_rate	0.0001
Trainer	
training_steps	2180
early_stop_patience	10
validation_steps	0.5
gradient_accumulation_steps	4
gradient_max_norm	1.0

Table 5: Hyperparameters for M1 models

A.2 M2

In Table 6 is an excerpt of the training configuration used for training the M2 models using the mBART encoder-decoder:

B Evaluation on test set of WMT21

We present the performance of the submitted ensembles on the TEST-21 dataset as calculated in the official QE results [6] for each task and sub-task. We also provide the comparison with the organisers' baseline.

[5] https://github.com/Unbabel/OpenKiwi/blob/master/config/xlmroberta.yaml

[6] https://www.statmt.org/wmt21/quality-estimation-task_results.html

System	
bottleneck_size	256
dropout	0.1
hidden_size	2048
nr_frozen_epochs	0.333
Optimizer	
optimizer	adam
encoder_learning_rate	6.0e-06
learning_rate	1.0e-05
Trainer	
training_steps	5512
early_stopping_patience	2
save_top_k	3
batch_size	4
gradient_accumulation_steps	4

Table 6: Hyperparameters for M2 models

B.1 Task 1: Direct Assessments prediction at sentence-level

The results for Task1 on TEST-21 are presented in Table 7.

B.2 Task 2: HTER prediction at sentence-level

The results for Task2 on TEST-21TEST-21 are presented in Table 8, showing the performance for the sentence level, HTER score predictions.

B.3 Task 2: Word-level prediction

The results for Task2 on TEST-21 are presented in Table 9, showing the performance for the word tag predictions.

C Statistics on the Metrics data

We present below (Tables 10 and 11) the statistics on the Metrics data used to train the M1M model on direct assessments.

METHOD	PEARSON R↑	MAE↓	RMSE↓
MULTILINGUAL			
IST-UNBABEL	0.665	0.627	0.482
BASELINE	0.541	0.729	0.562
EN-DE			
IST-UNBABEL	0.579	0.567	0.393
BASELINE	0.403	0.629	0.433
EN-ZH			
IST-UNBABEL	0.586	0.631	0.499
BASELINE	0.525	0.683	0.534
RO-EN			
IST-UNBABEL	0.899	0.393	0.289
BASELINE	0.818	0.556	0.408
ET-EN			
IST-UNBABEL	0.796	0.519	0.404
BASELINE	0.660	0.700	0.543
NE-EN			
IST-UNBABEL	0.856	0.515	0.401
BASELINE	0.738	0.657	0.524
SI-EN			
IST-UNBABEL	0.605	0.742	0.583
BASELINE	0.513	0.797	0.626
RU-EN			
IST-UNBABEL	0.792	0.583	0.412
BASELINE	0.677	0.702	0.492
ZERO-SHOT LANGUAGE PAIRS			
EN-CZ			
IST-UNBABEL	0.577	0.751	0.583
BASELINE	0.352	0.845	0.686
EN-JA			
IST-UNBABEL	0.355	0.764	0.566
BASELINE	0.230	0.816	0.617
PS-EN			
IST-UNBABEL	0.628	0.780	0.658
BASELINE	0.476	0.852	0.711
KM-EN			
IST-UNBABEL	0.650	0.721	0.568
BASELINE	0.562	0.788	0.614

Table 7: Results for Task 1 on the held-out evaluation set of WMT 2021.

METHOD	PEARSON R↑	MAE↓	RMSE↓
MULTILINGUAL			
IST-UNBABEL	0.597	0.219	0.171
BASELINE	0.502	0.235	0.188
EN-DE			
IST-UNBABEL	0.617	0.172	0.116
BASELINE	0.529	0.183	0.129
EN-ZH			
IST-UNBABEL	0.290	0.266	0.220
BASELINE	0.282	0.287	0.246
RO-EN			
IST-UNBABEL	0.879	0.122	0.098
BASELINE	0.831	0.142	0.115
ET-EN			
IST-UNBABEL	0.811	0.153	0.112
BASELINE	0.714	0.195	0.149
NE-EN			
IST-UNBABEL	0.718	0.161	0.126
BASELINE	0.626	0.205	0.160
SI-EN			
IST-UNBABEL	0.710	0.178	0.136
BASELINE	0.607	0.204	0.159
RU-EN			
IST-UNBABEL	0.539	0.224	0.165
BASELINE	0.448	0.255	0.188
ZERO-SHOT LANGUAGE PAIRS			
EN-CZ			
IST-UNBABEL	0.529	0.271	0.200
BASELINE	0.306	0.262	0.206
EN-JA			
IST-UNBABEL	0.275	0.279	0.224
BASELINE	0.098	0.279	0.232
PS-EN			
IST-UNBABEL	0.555	0.328	0.284
BASELINE	0.503	0.333	0.290
KM-EN			
IST-UNBABEL	0.655	0.243	0.199
BASELINE	0.576	0.241	0.196

Table 8: Results for Task 2 sentence-level system on the held-out evaluation set of WMT 2021.

METHOD	SRC-MCC↑	TGT-MCC-WORDS↑	TGT-MCC-GAPS↑
		EN-DE	
IST-UNBABEL	0.404	0.466	0.183
BASELINE	0.322	0.370	0.116
		EN-ZH	
IST-UNBABEL	0.286	0.310	0.068
BASELINE	0.241	0.247	0.065
		RO-EN	
IST-UNBABEL	0.603	0.649	0.357
BASELINE	0.511	0.536	0.205
		ET-EN	
IST-UNBABEL	0.522	0.570	0.254
BASELINE	0.405	0.461	0.136
		NE-EN	
IST-UNBABEL	0.445	0.508	0.268
BASELINE	0.390	0.440	0.215
		SI-EN	
IST-UNBABEL	0.406	0.528	0.258
BASELINE	0.335	0.425	0.208
		RU-EN	
IST-UNBABEL	0.351	0.332	0.165
BASELINE	0.251	0.256	0.073
		ZERO-SHOT LANGUAGE PAIRS	
		EN-CZ	
IST-UNBABEL	0.294	0.376	0.125
BASELINE	0.224	0.273	0.039
		EN-JA	
IST-UNBABEL	0.175	0.169	0.025
BASELINE	0.175	0.131	0.036
		PS-EN	
IST-UNBABEL	0.294	0.370	0.177
BASELINE	0.249	0.313	0.134
		KM-EN	
IST-UNBABEL	0.345	0.448	0.259
BASELINE	0.279	0.351	0.175

Table 9: Results for Task 2 word-level system on the held-out evaluation set of WMT 2021.

	CS-EN LT-EN	DE-EN GU-EN	FI-EN KK-EN	RU-EN JA-EN	RO-EN KM-EN	TR-EN PL-EN	ZH-EN PS-EN	ET-EN TA-EN
Total tuples	28887 10315	91584 9063	47205 6789	61505 8917	560 4722	30746 11666	71941 4611	20496 7562
Avg. tokens (reference)	31.43 26.84	24.61 17.73	20.48 20.65	23.31 28.64	24.35 19.49	23.32 21.93	31.70 19.87	23.93 19.91
Avg. tokens (source)	25.65 20.61	22.93 15.13	14.49 16.47	19.77 3.27	24.99 29.91	19.01 18.55	6.05 21.87	18.61 15.31
Avg. tokens (MT)	29.99 25.44	24.19 17.15	19.95 20.00	23.51 27.41	24.42 19.59	22.97 21.64	30.60 19.37	24.06 20.14

Table 10: Statistics for the WMT 15 to 20 Direct Assessments corpus into-English language pairs.

	EN-RU EN-ET	EN-CS EN-LT	EN-DE EN-GU	EN-FI EN-KK	EN-LV EN-JA	EN-TR EN-PL	EN-ZH EN-TA
Total tuples	63771 13376	60905 8959	55352 6924	30924 8219	5810 9573	5171 10506	66830 7886
Avg. tokens (reference)	22.48 18.83	23.48 20.61	23.96 22.07	17.7 19.21	20.45 1.4	19.74 24.54	7.26 19.84
Avg. tokens (source)	24.5 24.23	25.82 24.09	24 24.3	23.21 24.13	24.99 25.2	24.2 25.33	28.81 25.15
Avg. tokens (MT)	22.14 18.96	23 20.62	23.84 22.39	17.81 19.71	21.18 2.29	19.24 23.19	7.53 19.18

Table 11: Statistics for the WMT 15 to 20 Direct Assessments corpus from-English language pairs.

The IICT-Yverdon System for the WMT 2021 Unsupervised MT and Very Low Resource Supervised MT Task

Àlex R. Atrio[1,2] and **Gabriel Luthier**[1] and **Axel Fahy**[1] and **Giorgos Vernikos**[1,2] and **Andrei Popescu-Belis**[1,2] and **Ljiljana Dolamic**[3]

[1]HEIG-VD / HES-SO
Yverdon-les-Bains
Switzerland

[2]EPFL
Lausanne
Switzerland

[3]Armasuisse, W+T
Thun
Switzerland

name.surname@heig-vd.ch

ljiljana.dolamic@armasuisse.ch

Abstract

In this paper, we present the systems submitted by our team from the Institute of ICT (HEIG-VD / HES-SO) to the Unsupervised MT and Very Low Resource Supervised MT task. We first study the improvements brought to a baseline system by techniques such as back-translation and initialization from a parent model. We find that both techniques are beneficial and suffice to reach performance that compares with more sophisticated systems from the 2020 task. We then present the application of this system to the 2021 task for low-resource supervised Upper Sorbian (HSB) to German translation, in both directions. Finally, we present a contrastive system for HSB-DE in both directions, and for unsupervised German to Lower Sorbian (DSB) translation, which uses multi-task training with various training schedules to improve over the baseline.

1 Introduction

In this paper, we present the systems submitted to the WMT 2021 task on Unsupervised MT and Very Low Resource Supervised MT. We first build a series of baseline systems, driven mostly by considerations of simplicity, trained on data from the 2020 edition of the task, for translation between Upper Sorbian (HSB) and German (DE). These systems, described in Section 3, enable us to quantify the merits of using additional back-translated data (Sennrich et al., 2016) and of initializing the system for a low-resource pair with parameters learned on a high-resource pair (same target language and related source language).

The systems described above serve as the basis for our 2021 baseline submitted to the shared task, for DE→HSB and HSB→DE, presented in Section 4, which improves upon our 2020 baseline with the addition of more parallel data, and achieves competitive performance with the use of back-translation and parent-initialization only. However, this approach does not lead to an effective baseline for unsupervised German to Lower Sorbian (DSB) translation (Section 5). In Section 6, we present experiments with a contrastive system that implements multi-task learning, with several schedules, in which denoising tasks together with translation are presented to the systems in increasing order of complexity, leading to more robust HSB↔DE systems, together with a strategy of diverse ensembling. We also use our DE→HSB system to initialize a multi-task DE→DSB system for the unsupervised task, although in this case the performance is not competitive.

2 Datasets

We use various Upper Sorbian datasets from the 2020 edition of the task, and additional WMT data, as presented in Table 1. The monolingual HSB data from 2020 comes from three sources: `sorbian_institute_monolingual` consists of a mix of high- and medium-quality HSB data provided by the Sorbian Institute; `witaj_monolingual` consists of high-quality HSB data from the Witaj Sprachzentrum; finally, `web_monolingual` consists of web-scraped noisier HSB data gathered by the Center for Information and Language Processing from LMU Munich (Fraser, 2020). We kept from all datasets only sentences that have strictly more than 2 and strictly fewer than 301 words.

3 Baseline HSB→DE System on 2020 Data

3.1 Subword Vocabulary

For the HSB→DE system, we use CS→DE initialization in several experiments, because Czech (CS) is a high-resource language and close neighbor to Upper Sorbian. Therefore, we create a tri-lingual shared subword vocabulary (CS, DE, HSB) using the Unigram LM model (Kudo,

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 973–981
November 10–11, 2021.

Dataset	Language	Before filtering		After filtering	
		sentences	words	sentences	words
Sorbian Institute Monolingual	HSB	339,822	5,044,079	339,822	5,044,079
Web Monolingual	HSB	121,003	1,661,898	115,632	1,651,154
Witaj Monolingual	HSB	222,027	2,672,255	215,370	2,660,805
Europarl v8	DE	2,234,583	48,430,884	2,186,477	48,347,698
JW300	DE	2,366,722	34,782,112	2,182,801	34,519,064
News Commentary v15	DE	422,009	8,942,517	409,955	8,939,335
Europarl v8 CS-DE	CS	568,589	11,571,876	562,716	11,561,049
Europarl v8 CS-DE	DE	=	13,098,638	=	13,086,320
JW300 CS-DE	CS	1,052,338	13,579,350	982,034	13,435,536
JW300 CS-DE	DE	=	15,133,882	=	14,992,424
News Commentary v13 CS-DE	CS	174,789	3,486,672	172,987	3,479,819
News Commentary v13 CS-DE	DE	=	3,751,102	=	3,746,708
WMT 2020 HSB-DE Train	DE	60,000	724,572	59,030	722,076
WMT 2020 HSB-DE Train	HSB	=	639,740	=	637,883
WMT 2021 HSB-DE Train	DE	87,521	1,251,339	87,502	1,251,287
WMT 2021 HSB-DE Train	HSB	=	1,094,421	=	1,094,375

Table 1: Monolingual and parallel corpora with their languages and numbers of lines (sentences) and words, before and after filtering by length (keeping sentences with more than 2 and fewer than 301 words).

2018) as implemented in SentencePiece.[1] We apply 32,000 merges and the other parameters of SentencePiece are kept to default values. We obtain 600k sentences of HSB data from `sorbian_institute_monolingual`, `witaj_monolingual` and `train.hsb-de`, the latter being the HSB side of the 2020 training data. We do not use `web_monolingual` as it appears to be noisy, due to the collection process. For CS and DE, 600k sentences are selected randomly from the monolingual corpora listed in Table 1. The vocabulary generated by SentencePiece is converted from log probabilities to frequencies using the `spm_to_vocab.py` tool from the OpenNMT-py toolkit. Using a common SentencePiece model for the three languages is not obligatory, but appeared to improve the performance by 2-3 BLEU points in most cases.

3.2 System Parameters and Results

We use OpenNMT-py (Klein et al., 2017) for our experiments.[2] We start with Transformer-Base (Vaswani et al., 2017) (78M parameters) but also experiment with Transformer-Big (245M parameters), with their main parameters described in Table 2. We apply the same regularization and optimization procedures to the two models. We accumulate gradients over 2 batches and train on 2 GPUs, with a `batch_size` of 1k for Base and and 2k for Big. We use the "noam" learning rate schedule (Vaswani et al., 2017) with its values at each step multiplied by two, and 8k warmup steps. We evaluate and save checkpoints every 5k steps. Final translations are generated with a beam width of 5, ensembling the last two checkpoints in these experiments. We report BLEU scores (Papineni et al., 2002) obtained with SacreBLEU (Post, 2018) on detokenized text.

	N	h	d_{model}	d_{ff}	P_{drop}	steps
Base	6	8	512	2048	0.1	60k
Big	6	16	1024	4096	0.3	100k

Table 2: Parameters of the two Transformer models used in our experiments. Other parameters are set to the default values of the OpenNMT-py toolkit.

3.3 Use of Back-translated Data

The first HSB→DE system we trained, for comparison purposes, used only the HSB/DE parallel data provided for the WMT 2020 Low-Resource task. Its BLEU scores are 47.98 on the 'dev' set (`devel.hsb-de`) and 41.22 on the 'devtest' set (`devel_test.hsb-de`) after 60k steps of training (first line of Table 3). The already high BLEU scores that are reached, compared to scores generally observed on high-resource language pairs,

[1] https://github.com/google/sentencepiece (v. 0.1.95)
[2] https://github.com/OpenNMT/OpenNMT-py (v. 2.0.1)

indicate that the 'dev' and 'devtest' sets are probably quite similar to the training data.

We obtain additional training data through back-translation (Sennrich et al., 2016) of widely available monolingual German data. To this end, we train a DE→HSB model on the same parallel corpus as above, which reaches BLEU scores of 45.23 / 40.62 respectively on 'dev' and 'devtest'. Using this model, we translate News Commentary V15 from German into Upper Sorbian. The resulting pseudo-parallel data (noisy on the HSB side) is used in addition to the initial data for training a new HSB→DE model, which reaches a score of 52.91 / 44.39 (second line of Table 3). The improvement of this single enrichment with imperfect data of the initial low-resource system thus exceeds 4 BLEU points.

3.4 Initialization with Parameters from a High-Resource Pair

The second technique we use for improvement is transfer from a high-resource pair (Zoph et al., 2016; Kocmi and Bojar, 2018), i.e. initialization with parameters from an MT system trained on such a pair. As Upper Sorbian has many similarities with Czech, which is a high-resource language, we initialize the HSB-DE model with the parameters of a model trained for CS→DE, then train it with the same data as in the previous subsection. Firstly, the CS→DE model is trained using Europarl and News Commentary, and reaches a BLEU score of 27.13 on a sample test set extracted from these two corpora.

The resulting HSB→DE system reaches BLEU scores of 55.99 / 47.53, a further increase of about 3 BLEU points (third line of Table 3). The use of an even larger dataset further improves performance: the addition of the JW300 corpus (Agić and Vulić, 2019) to the CS→DE training data increases BLEU by half a point (56.5 on 'dev'). The rather small increase could be attributed to the large difference in domains between JW300 and the HSB/DE data.

Since back-translation can provide very large amounts of data, we also trained a Transformer Big (with the parameters shown in Table 2) with the addition of the monolingual German corpora of Europarl and JW300 backtranslated into Upper Sorbian. This model reaches 58.08 / 49.99 BLEU points respectively on 'dev' and 'devtest', improving performance by more than 1.5 BLEU points. This is currently our best baseline model for HSB→DE, obtained with two simple augmentation techniques only.

We can compare this score with three of the highest-scoring systems on the 2020 HSB→DE 'devtest' set, noting some of the differences between them and our baseline. Scherrer et al. (2020) achieved a BLEU score of 56.9 using back-translation and bilingual pre-training with CS→DE, but also scheduled multitask with several monolingual and multilingual tasks. Knowles et al. (2020) achieved a BLEU score of 58.9 using iterative back-translation, multiplication of the HSB data for BPE training, and character- and word-level lexical modifications of Czech to make it more similar to Upper Sorbian. Libovický et al. (2020) achieved a score of 56.0 with much larger corpora for back-translation and CS→DE pre-training (14M lines) and the use of an unsupervised CS→HSB system to translate the CS side of the DE/CS parallel data into HSB.

3.5 Initialization with Parameters from Other High-Resource Pairs

We studied the role of the closeness between Upper Sorbian and the high-resource source language used for initialization, by reproducing the above initialization experiments (CS→DE) with Polish and French instead of Czech. Polish is a West Slavic language just as Czech and Upper Sorbian, although geographically more remote, whereas French is a Romance language: we thus expected the former to outperform the latter. To keep training time more manageable, we used a Transformer-Base, and trained the parent model on Europarl and JW300, because News Commentary is not available for Polish. For each experiment we build a different tri-lingual SentencePiece model trained with 600k sentences per language.

The use of the PL→DE model (with a 22.33 BLEU score on its respective test set) for initialization leads to a HSB→DE performance of 56.07 / 47.94, which is very similar to the system initialized with CS→DE parameters (55.99 / 47.53). The use of the FR→DE model (with a 19.25 BLEU score) for initialization leads to a HSB→DE system reaching 54.92 / 46.30. This is about 1.3 BLEU points lower than with Polish or Czech, although the difference is smaller than expected given the linguistic distance between French and Upper Sorbian. These results are in line with the findings of Aji et al. (2020) who argue that no parent is clearly better than other for transfer learning in MT.

System	HSB→DE		DE→HSB	
	dev	devtest	dev	devtest
1. Transformer-Base, 2020 parallel data	47.98	41.22	45.23	40.62
2. Add back-translated data to #1	52.91 *(+4.93)*	44.39 *(+3.17)*	51.00 *(+5.77)*	43.23 *(+2.61)*
3. Initialize #2 with high-resource pair	55.99 *(+3.08)*	47.53 *(+3.14)*	–	–
4. Transformer-Big with #3	58.08 *(+2.09)*	49.99 *(+2.46)*	–	–
5. Add 2021 parallel data to #4	59.29 *(+1.21)*	51.86 *(+1.87)*	57.22 *(+6.22)*	49.95 *(+6.72)*

Table 3: Scores of our 2020 (1–4) and 2021 (5) baseline systems, with absolute improvements brought by each additional technique or data set.

3.6 Two Rounds of Back-Translation

Multiple rounds of back-translation can be done on each side, but this computational effort is not always compensated by a significant increase of the BLEU score. Using the best HSB→DE system above, we translate monolingual HSB data and use it to train an improved DE→HSB model, which reaches 51.00 on the 'dev' data (+5.77 with respect to the initial DE→HSB system) and 43.23 on the 'devtest' data (+2.61). We then use this improved model to translate the monolingual German data again and use the resulting pseudo-parallel data to train a new HSB→DE model. The model without CS initialization reaches BLEU scores of 53.62 on 'dev' (+0.62) and 44.95 on 'devtest' (+0.43). If CS initialization is used, the models reaches respectively 58.44 (+0.36) and 50.03 (+0.04) on 'dev' and 'devtest'. The improvement brought by the additional rounds of back-translation is quite marginal, therefore we do not pursue this approach, and focus on a system which is initialized from a parent high-resource pair and trained with original and back-translated data, where the latter comes from a reverse system trained only with the original parallel HSB-DE data provided by the shared task.

4 Baseline HSB↔DE Low Resource Systems for 2021

Given the results of the previous section, we choose the Transformer-Big for our 2021 baseline. We change the dropout level from 0.3 to 0.1 since our experiments revealed an increase in performance with the latter value. Furthermore, we add the 87,502 sentences of additional parallel HSB-DE training data provided in 2021 to the datasets used in our 2020 baseline. We use the same SentencePiece model with DE, HSB, and CS data that we used for our 2020 baseline system, with approximately 700k lines for each language. At translation time, after observing a number of out-of-vocabulary tokens, we replace the unknown tokens with the source token that has the highest attention weight. We do not make any further changes regarding our 2020 Transformer-Big model.

The scores of our baseline systems on 2020 and 2021 data are synthesized in Table 3 for the various techniques we experimented with. Our baseline HSB→DE model with combined 2021 and 2020 data is system #5 in Table 3: it reaches BLEU scores of 59.29 on the 'dev' set and 51.86 on the 'devtest' set after training for 150,000 steps and by ensembling the best 4 saved checkpoints. For our DE→HSB model, we obtain 57.22 on the 'dev' set and 49.95 on the 'devtest' set after training for 85,000 steps and by ensembling the best 4 saved checkpoints.

After the submission to the 2021 shared task, we continued training the above HSB→DE model up to 300,000 steps- Ensembling the *last* 4 saved checkpoints, BLEU scores were close to the ones shown in the last line of Table 3, reaching 59.42 on the 'dev' set and 51.37 on the 'devtest' set. However, several checkpoints gained almost 2 BLEU points on 'dev', pointing to the potential benefits of training for a longer time.

5 Baseline for Unsupervised DE→DSB Translation

Moreover, we studied the same techniques for translating Lower Sorbian (DSB), for which no parallel resources are provided. We translated the monolingual DSB data provided by the organizers with our HSB→DE model, hypothesizing that the differences between DSB and HSB are small enough to obtain an acceptable DSB-DE pseudo-parallel corpus, with high-quality text on the DSB side, following insights from our experience with Swiss-German dialects (Honnet et al., 2018).

We use the parameters from our best DE→HSB model to initialize a DE→DSB model that we train

for 120k steps with the DSB-DE pseudo-parallel data. When ensembling the best 4 checkpoints, we reach BLEU scores of 8.25 / 8.22 without observing any significant increase of the scores during training. In fact, the initial score, which is the performance of a DE→HSB model on the DE-DSB 'devtest' data, is even slightly higher. An even lower BLEU score was reached when using our CS→DE model to translate monolingual DSB data into DE to obtain a pseudo-parallel corpus, thus confirming the finding that this approach does not lead to pseudo-parallel corpora of sufficient quality. Therefore, we did not submit these translations to the 2021 shared task.

6 Contrastive HSB↔DE and DE→DSB Systems using Multi-Task Learning

In contrast to the baseline systems presented above, we study an innovative approach, in which we train multitask systems with denoising auxiliary tasks that are presented in order of increasing complexity. This insight is drawn from curriculum learning (Bengio et al., 2009). We thus test whether increasing the complexity of the tasks makes it easier for an NMT model to learn the simple tasks first, and the harder ones later in training.

As Raffel et al. (2020) showed, source–to–source pre-training and multitasking improves translation, but not enough to compete with state-of-the-art setups. Therefore, instead, we perform target–to–target and source+target–to–target denoising. Considering their findings, we decide not to introduce special tokens into our vocabulary, such as mask tokens (instead just deleting the tokens with wish to mask), or sentence and language separators. Finally, due to computational constraints, we use the Transformer-Base as our architecture.

6.1 Data and Auxiliary Tasks

For our contrastive system we consider two new monolingual corpora in Czech and in German: the document-separated news crawls from WMT20 (Barrault et al., 2020), consisting of text extracted from online newspapers. They contain 17M lines and 43M lines respectively in each language. To keep training time within acceptable limits, we sample 1.4M lines from these corpora (including empty lines that serve as document-separators), we apply the same length-based filtering criterion ($2 < L < 301$) as for our baseline data, and we also delete all sentences that are made of more than 15% non-alphabetic characters. The resulting Czech corpus is 1.3M lines and 131,644 documents long, and the German corpus is 1.2M lines and 130,891 documents long.

For our document-level denoising tasks, we first divide into "chunks" a tokenized document-separated corpus so that each chunk is no more than 500 subwords in length, made up of consecutive lines in the same document; we only select documents made of at least 3 sentences. In Table 4 we list all corpora that we use to create our auxiliary data, including monolingual corpora back-translated with our baseline systems. The DE→DSB back-translated data was obtained with a baseline DE→HSB model.

We make use of the four following auxiliary denoising tasks (the main task being of course standard sentence-level translation, with all parallel and back-translated data), with the first two inspired by Devlin et al. (2019); Raffel et al. (2020) and Conneau and Lample (2019):

1. **Masking** (`MASK`): randomly delete 15% of words of a line on the source side, but keep the full original sequence on the target side.
2. **Translation Language Modeling** (`TLM`): concatenate the source and target sentences from a parallel corpus, and apply separately the `MASK` algorithm to each one. The target is the original target sentence.
3. **Mask Document First Words** (`MF`): for each chunk, leave the first sentence untouched, and for the remaining ones delete the first word of each sentence, with the target being the full original sequence in the same language.
4. **Next Sentence Generation** (`NSG`): for each chunk, leave all the sentences untouched except the last one, of which delete all but the two longest words; the model has to output the full original sequence. Keeping the two longest words (in characters) is based on the assumption that they are the most informative ones in the sentence.

The denoising tasks are listed above by increasing complexity. Indeed, `MASK`, as a monolingual sentence-level task, is the simplest denoising task we present, with `TLM` following, as it includes a context in a different language which needs to be identified. The two document-level tasks are more

complex, as they require a larger context. In particular, NSG is harder than MF, since it consists of reconstructing a whole sentence with just two words from the original sequence, forcing the model to look for a more abundant context to estimate the correct answer. Furthermore, predicting the first word requires to take into account exclusively inter-sentential context, whereas masking a single random word allows also for the use of intra-sentential context, with the latter providing more direct context than the former.

Corpus	Lines	Words	Aux. tasks
CS-DE	1.5M	25M / 28M	TLM
HSB-DE	144k	2M / 2M	TLM
CS	1.3M	41M	MF, NSG
DE	1.2M	44M	MF, NSG
HSB	640k	9M	MASK
DSB	128k	2M	MASK
HSB→DE	4.5M	94M / 104M	
DE→HSB	637k	10M / 9M	
DE→DSB	124k	2M / 2M	

Table 4: Parallel (2), monolingual (4), and back-translated corpora (3) used for our contrastive system trained with multi-tasking. Each corpus is assembled from the raw datasets presented in Table 1 with the filtering setup described in Subsection 6.1. For bilingual corpora, we indicate the number of words in each language.

6.2 Training Schedules

All our models translate to one target language only, therefore the target side of our datasets is always the same language, be it for the monolingual denoising tasks or for TLM. Since all datasets correspond to sequence-to-sequence tasks, we are in essence simply removing and introducing datasets during training. The specific splits of the tasks in each training schedule have been manually set, guided by the reasons given below, without any attempt for fine-tuning.

All the hyperparameters of the models are those presented in Section 3, with the only exception of the parameters of CS↔DE models for initialization, which were trained on 4 GPUs to reduce training time. When we introduce new tasks during the training of a model, we continue training from the last checkpoint of the previous task.

Training CS↔DE models. Both directions are trained according to the same schedule, shown in Table 5, with simply the source and target languages switched. First, we train for 30k steps with a TLM task, then we train for another 30k steps with a mixture of the MF auxiliary task (50% of the samples) and the main translation task (50%). Then we continue for another 30k steps, changing MF to NSG. Finally, we finish with 30k steps on translation only. In total, the model is being trained for 30k steps (25%) with TLM, 15k steps (12.5%) with MF, 15k steps (12.5%) with NSG, and 60k steps (50%) with the main task, i.e. sentence-level translation.

	Steps ×1000			
Task	0-30	30-60	60-90	90-120
TLM	100%			
MF		50%		
NSG			50%	
Translation		50%	50%	100%

Table 5: Training schedule of the parent models in CS↔DE. For each direction, the model is only trained to output target language, so corpora differ depending on the direction (see 6.1). Both models are trained for 120k steps with three auxiliary denoising tasks and the main sentence-level translation task.

HSB→DE. The schedules of the child models are shown in Table 6 for the (DE, HSB) pair. For HSB→DE, we continue training from the best scoring checkpoint of the last 60k steps of the parent CS→DE model, and start with a TLM task for 60k steps. Then, we introduce back-translated data only for 60k steps. We continue with 60k steps with true parallel data only.

Additionally, we train two more models by continuing to train another 60k steps from the best scoring checkpoint (which is also the last one saved), with one of the models having its learning rate schedule reset. Although at first performance worsens due to a more aggressive learning rate during the warmup steps, the model ends up converging to a score similar to the one we obtain if we continue to train without resetting the learning rate schedule. The goal is to emulate a multiple-run seeding strategy for ensembling, by achieving a different weight distribution among the two models. We additionally train a randomly-initialized model with parallel data only, for 60k steps, also for ensembling. We generate our translations of the test data with an ensembling of 16 models: the best 4 checkpoints from the parallel-only randomly-initialized

model, the best 4 of our main setup during the first 60k steps of parallel-only training, and the 4 checkpoints each for the two runs that continued to train with, and respectively without, resetting the learning rate schedule.

DE→HSB. We continue training from the best-scoring checkpoint of the last 60k steps of DE→CS, and provide it with a MASK task for 60k steps, since the model has not seen the target language at all during pre-training, for this direction. Then, we provide the model with a TLM task for 60k steps. Since in this direction we have much less back-translated data than in the opposite, we decide to train for 60k more steps with 50% of the samples being from the back-translated data, and the other 50% from the true parallel corpora. Finally, we continue training two more models in the same manner as explained for the HSB→DE direction. We additionally train a randomly-initialized parallel data only model for 60k steps for ensembling. We translate with the same ensembling setup as described for the HSB→DE direction.

	Steps ×1000		
Task	0-60	60-120	120-180
	HSB→DE		
TLM	100%		
Trans-BT		100%	
Trans-Parallel			100%
	DE→HSB		
MASK	100%		
TLM		100%	
Trans-BT			50%
Trans-Parallel			50%

Table 6: Training schedule of the child models for the HSB→DE and DE→HSB models presented in 6.2.

DE→DSB. We start training with a MASK task for 60k steps from the highest-scoring checkpoint DE→HSB. We continue training for 60k steps with just the back-translated data, although we notice that the quality of the translation affects negatively the scores. To address this issue, for another 60k steps we give it the back-translated corpus for 50% of the samples and the MASK task for the other 50%, starting training from the previous highest-scoring checkpoint. Finally, for another 60k steps we give it a parallel-only DE-HSB task for 50% of the samples, MASK for 30%, and back-translated data for 20%. After testing, using just the highest-scoring checkpoint for the back-translation only, back-translation + MASK, and DE-HSB + back-translation + MASK appeared to work better on the development data than using the highest four ones.

	Steps ×1000			
Task	0--60	60--120	120--180	180--240
MASK	100%		50%	30%
Trans-BT		100%	50%	20%
DE-HSB				50%

Table 7: Training schedule of the child DE→DSB models presented in 6.2

6.3 Results

The scores of the parent DE→CS and CS→DE models obtained with multi-task training are shown in Table 8. Compared to the CS→DE models from Sections 3 and 4, the present models have markedly lower scores. This difference can be due to the use of Transformer-Base vs. Big, or to differences in training data, apart from the multi-task training procedure itself. Still, we decided to use these models as parents for initializing the DE→HSB and HSB→DE models respectively, so that both parents and children are trained with multi-tasking. Although changes in the parameters of a parent model that result in better translations may not necessarily also result in better child initialization, it would be interesting to also test here the parent models from Section 4.

System	**DE→CS**	**CS→DE**
1. MF + translation	14.05	15.46
2. NSG + translation	15.30	16.17
3. Translation	18.19	19.80

Table 8: BLEU scores of parent models after each stage of the training schedule described in 6.2, on the 'devtest' set from 4.

Our child DE↔HSB models show that the scheduled training improves results over the baseline. The HSB→DE model with a training schedule (system 2 in Table 9), trained with a lighter architecture (Base vs. Big) and lower quality parent model (19.8 vs. 24.5), achieves a higher BLEU score than the system in Section 4, as shown in Table 3: 52.2 vs. 51.86. Additionally, the diversity of the ensembling of the models appears to improve the overall quality of the translation.

System	DE→HSB	HSB→DE
1. Parallel data	50.37	48.50
2. Multi-task	52.10	52.21
3. #2 cont. train	53.42	52.37
4. #2 cont. train with l. r. reset	53.05	52.12
Ensemble	54.58	53.21

Table 9: BLEU scores of child DE↔HSB models for various training schedules on the 2021 'devtest' set.

The scores of our DE→DSB model (Table 10) show that the quality of the back-translated data with our HSB→DE model improved slightly with the addition of the `MASK` monolingual task, but not with the addition of a DE→HSB translation task. However, when including in the ensemble the models trained on a DE→HSB task, scores improved from 8.7 to 9.6 on the 'devtest' set. This was the version submitted to the shared task on unsupervised MT (DE→DSB).

System	DE→DSB
1. Back-translation only	8.23
2. BT + `MASK`	8.57
3. BT + `MASK` + DE→HSB	7.14
Ensemble	9.62

Table 10: BLEU scores of child DE↔DSB models for various training schedules on the 2021 'devtest' set.

Finally, as we can see in Table 11, even with our possibly suboptimally trained parent models and lighter architecture, the strategy of diverse ensembles and scheduled multi-task training improved over our best performing baselines given in Section 4 for all directions of the low-resource MT task.

HSB→DE		DE→HSB		DE→DSB	
dev	devtest	dev	devtest	dev	devtest
62.74	53.21	62.49	54.58	9.22	9.62
(+3.45)	*(+1.35)*	*(+5.27)*	*(+4.63)*	*(+0.97)*	*(+1.40)*

Table 11: BLEU scores of our primary system's final configurations, on the development data, with the improvements over our highest baselines from Section 4.

7 Conclusion

In this work, we showed that non-iterative back-translation and parent-model transfer learning provide improvements for translation in a low-resource setting. Furthermore, multi-task scheduled training with monolingual or cross-lingual tasks also resulted in better models. In particular, child models starting with Translation Language Modeling tasks and Masking tasks improved over the baseline in all translation directions. Finally, our strategy of ensembling diverse models also produced higher scores than a mere checkpoint ensemble strategy.

Acknowledgments

We are grateful for their support to Armasuisse through the FamilyMT project, and to the Swiss National Science Foundation through grant n. 175693 for the DOMAT project: "On-demand Knowledge for Document-level Machine Translation".

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Alexander Fraser. 2020. Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Pierre-Edouard Honnet, Andrei Popescu-Belis, Claudiu Musat, and Michael Baeriswyl. 2018. Machine translation of low-resource spoken dialects: Strategies for normalizing Swiss German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

Rebecca Knowles, Samuel Larkin, Darlene Stewart, and Patrick Littell. 2020. NRC systems for low resource German-Upper Sorbian machine translation 2020: Transfer learning with lexical modifications. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1112–1122, Online. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Jindřich Libovický, Viktor Hangya, Helmut Schmid, and Alexander Fraser. 2020. The LMU Munich system for the WMT20 very low resource supervised MT task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1104–1111, Online. 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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. 2020. The University of Helsinki and Aalto university submissions to the WMT 2020 news and low-resource translation tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1129–1138, Online. 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.

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*, volume 30.

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.

Unsupervised Translation of German–Lower Sorbian: Exploring Training and Novel Transfer Methods on a Low-Resource Language

Lukas Edman **Ahmet Üstün** **Antonio Toral** **Gertjan van Noord**

Center for Language and Cognition
University of Groningen

{j.l.edman, a.ustun, a.toral.ruiz, g.j.m.van.noord}@rug.nl

Abstract

This paper describes the methods behind the systems submitted by the University of Groningen for the WMT 2021 Unsupervised Machine Translation task for German–Lower Sorbian (DE–DSB): a high-resource language to a low-resource one. Our system uses a transformer encoder-decoder architecture in which we make three changes to the standard training procedure. First, our training focuses on two languages at a time, contrasting with a wealth of research on multilingual systems. Second, we introduce a novel method for initializing the vocabulary of an unseen language, achieving improvements of 3.2 BLEU for DE→DSB and 4.0 BLEU for DSB→DE. Lastly, we experiment with the order in which offline and online back-translation are used to train an unsupervised system, finding that using online back-translation first works better for DE→DSB by 2.76 BLEU. Our submissions ranked first (tied with another team) for DSB→DE and third for DE→DSB.

1 Introduction

Unsupervised Neural Machine Translation (UNMT) has become increasingly useful in the field of MT, given that monolingual data is easier to gather compared to bilingual (or parallel) data. Such is especially the case for low-resource languages, which constitute the majority of languages in the world.

The WMT 2021 Unsupervised MT Task focuses on one such low-resource language: Lower Sorbian (DSB). The task is to translate between German (DE), a high-resource language, and Lower Sorbian, which is a very low resource language with roughly 150 thousand sentences of monolingual data available for the task at hand. The unsupervised task from prior years of WMT focused on German–Czech and German–Upper Sorbian translation. Unique to this year however is the relatively small amount of monolingual data available for DSB, compared to last year in which roughly 750 thousand sentences of Upper Sorbian were available. This makes it increasingly difficult to rely on the ubiquitous state-of-the-art UNMT methods (Lample and Conneau, 2019; Song et al., 2019; Liu et al., 2020), as they typically rely on a large amount of monolingual data available for both languages.

To alleviate the difficulty that comes with the lack of monolingual data for DSB, this year's WMT task allows for the use of monolingual and parallel data outside of DE–DSB. Specifically, all Upper Sorbian (HSB) data from WMT20 and all parallel data for German (DE) from WMT and OPUS (Tiedemann and Nygaard, 2004) are made available to use. Additionally, as auxiliary languages related to DSB, monolingual data for Czech (CS) and Polish (PL) is also provided.

Given the success of language transfer via multilingual models such as mBART (Liu et al., 2020), this fundamentally changes this year's unsupervised task from a bilingual NMT task to a multilingual task. However, pretrained multilingual models like mBART cannot be used as they do not fit the limitations on the training data that one is allowed to use for this shared task.

As the problem is unique to date due to the limited available data as well as the limitation on pre-existing pretrained models, we aim to establish a standard for training systems under these restrictions. Specifically, we ask three research questions (RQs):

1. Is it better to pretrain and fine-tune a multilingual model or to focus the training on a few languages at a time when data and time are limited?

2. How can we obtain a good initialization for the vocabulary of an unseen language for

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 982–988
November 10–11, 2021.

which there is very limited training data?

3. Given there are two methods for doing back-translation, online and offline, what is the best way to combine them?

Concerning RQ1, there is a wealth of research into multilingual models, however these typically require a large amount of monolingual data for each language in addition to a wealth of computational resources and time. Therefore, we develop the hypothesis that under limited constraints (time and computational resources) training only on two languages at a time will result in better performance due to there being fewer training objectives. Specific to this task, we propose training on DE coupled with CS, then HSB, then DSB itself, following the order of least to most linguistically similar to the low resource language that we target: DSB.

As for RQ2, due to the scarcity of monolingual data and complete absence of parallel data for DSB, our ability to train the model on this language is limited, thus the model's initialization for the language plays an increasingly important role in its resulting performance. Relying on the similarity of the two Sorbian languages, we aim to improve this initialization by transferring the model's knowledge of the HSB vocabulary to the DSB vocabulary.

Finally, with respect to RQ3, Garcia et al. (2020) established a method for incorporating offline and online back-translation (BT) into their MT system by first using offline BT then online BT, both following a multilingual pretraining. However, the reverse order (i.e. online BT followed by offline BT) has not been tested to the best of our knowledge, and theoretically doing online BT should improve the quality of the synthetic data that would be used for offline BT. Therefore, we test Garcia et al.'s method as well as the reverse order to establish the best practice for this task specifically.

The remaining of the paper is organized as follows. We first outline the data we chose to use and our preprocessing steps in Section 2. We then specify our architecture and training methods in Section 3. Our results are in Section 4, followed by our conclusions in Section 5.

2 Data

Apart from our main language pair of DE and DSB, we opted to use data from two languages that are related to the latter, namely CS and HSB.[1]

For HSB and DSB, we use all of the data provided for training by WMT. For DE and CS, we use data from WMT NewsCrawl, years 2010-11 and 2018-20. We chose these years as they are the most frequent years occurring in the HSB data, following Edman et al. (2020). For DE, we take the first 1 million sentences each from 2018-20 and 0.5 million from 2010-11, totalling 4 million. For CS, we take 0.5 million and 0.25 million from the respective years, totalling 2 million.

In terms of parallel CS–DE data, we use MultiParaCrawl, Europarl v8, WMT News v2019, and News Commentary v16, all available from the OPUS project (Tiedemann and Nygaard, 2004). The datasets are shown in Table 1.

Language(s)	Dataset Name	Sentences
CS	NewsCrawl 2010-11, 18-20	2,000,000
DE	NewsCrawl 2010-11, 18-20	4,000,000
DSB	WMT 2021	145,198
HSB	WMT 2020	696,271
CS–DE	Europarl v8	568,589
	MultiParaCrawl v7.1	5,680,308
	NewsCommentary v16	204,311
	WMT News v2019	20,567
HSB–DE	WMT 2020-21	147,521

Table 1: Training data used in our models.

While MultiParaCrawl (MPC) is the largest portion of Czech–German data, it is constructed using English as a pivot language, so we anticipate the data to be lower quality in general. As such, we run 2 models, 1 including MPC and 1 without MPC.

For development and testing, we use the DE–HSB and DE–DSB `devel` and `devel_test` datasets provided by WMT. For CS–DE, we make use of the WMT News Translation Task dev set, using `newstest2012` for development and `newstest2013` for testing.

All data is tokenized using the Moses toolkit (Koehn et al., 2007). We then apply BPE (Sennrich et al., 2016), for all languages jointly, using FastBPE.[2] The segmentation is applied on the same number of randomly-selected sentences for each language, roughly 145 thousand, matching the number of sentences in the DSB training data. We experiment with the number of joins used (trying {20, 40, 50, 60, 80} thousand), finding 50 thousand

[1] We initially also used Polish data but did not see any improvements so we ultimately left it out.

[2] https://github.com/glample/fastBPE

to perform the best, according to BLEU scores after step 4.

3 Method

3.1 Architecture

We used the MASS (Song et al., 2019) model, which is a 12-layer encoder-decoder (6 layers each) Transformer model identical to the XLM (Lample and Conneau, 2019) architecture. The difference comes in the training, using the MASS sequence masking (MA) objective allows both the encoder and decoder to be trained in the language model pretraining phase. This can be contrasted with XLM, which only pretrains the encoder.

3.2 Training

The training objectives we use are:

- MASS sequence masking (MA): Reconstructing a sentence fragment (a token sequence) given the remaining part of the sentence.
- Machine translation (MT): The standard translation objective with a cross-entropy loss.
- Denoising auto-encoding (AE): Reconstructing the original text from a noisy version corrupted by a set of functions.
- Back-translation (BT): Both online (on-the-fly) and offline back-translation, where synthetic data created from the forward direction is used to train the backward direction, and vice versa.
- Cross-lingual back-translation (XBT) (Li et al., 2020): Using an intermediate reference language for forward and backward translation during online BT.

We tested various training schemes, ultimately deciding upon a 6-step process, shown in Figure 1.

In Figure 1, UNMT (steps 4 and 6) refers to the combination of AE, online BT, and XBT. For the AE, we use word shuffling, masking, and removal as noise functions, following XLM (Lample and Conneau, 2019). Online BT is done on DE–DSB as well as on HSB–DSB.[3] XBT is done using our DE–HSB parallel data for the directions HSB→DSB→DE and DE→DSB→HSB.[4]

[3]We found in initial testing that including HSB–DSB gave a slight improvement to BLEU scores.

[4]We also tried using our DE–CS data for XBT, but this performed worse.

3.3 Vocabulary Transfer

To facilitate a better alignment of DSB to DE, we make use of the linguistic similarity of DSB to HSB, coupled with the fact that HSB is expected to be reasonably well-aligned to DE after training with the HSB–DE parallel data (step 2). As the model is language agnostic, apart from the language embeddings as well as word embeddings that occur exclusively in one language, we initially align these parts on the DSB side to the HSB side (step 3), prior to the first UNMT training (step 4).

We align the language embeddings by copying the HSB language embeddings to the DSB language embeddings. To align the vocabulary, we first train two word embeddings on the DSB and HSB data using fastText (Bojanowski et al., 2017). Next, we align these two embeddings using VecMap (Artetxe et al., 2018), treating identical words in HSB and DSB as the same.

From the aligned embeddings, we construct a bilingual dictionary. This is done by first getting the top 10 nearest HSB neighbors, according to the cosine similarities of the aligned embeddings, for each DSB word. From these 10 candidates, we choose the closest HSB word, determined by the lowest Levenshtein distance between the DSB word and the respective HSB candidate.[5] We also filter out DSB words that occur frequently in the DE training data, removing all those which occur more than 0.001% of the time.

We use this filtered bilingual dictionary to copy the embeddings within the encoder and decoder of the MASS model. Specifically, all of the embeddings for the words on the DSB side of the bilingual dictionary are copied from the embedding of their corresponding HSB word pair.

As DSB has not yet been seen in the training, a large number of the embeddings for DSB words are essentially not learned at this stage. However, since the Sorbian languages are closely related and often the differences between words are merely in spelling, we expect this approach to help with initializing the model's DSB vocabulary.

3.4 Experimental Setup

Training is done on an Nvidia V100 32GB GPU. Each training step of the model is limited to 24 hours, with the exception of our model using MPC data, in which the first step is trained for 2 days

[5]We strip accents before calculating the Levenshtein distance.

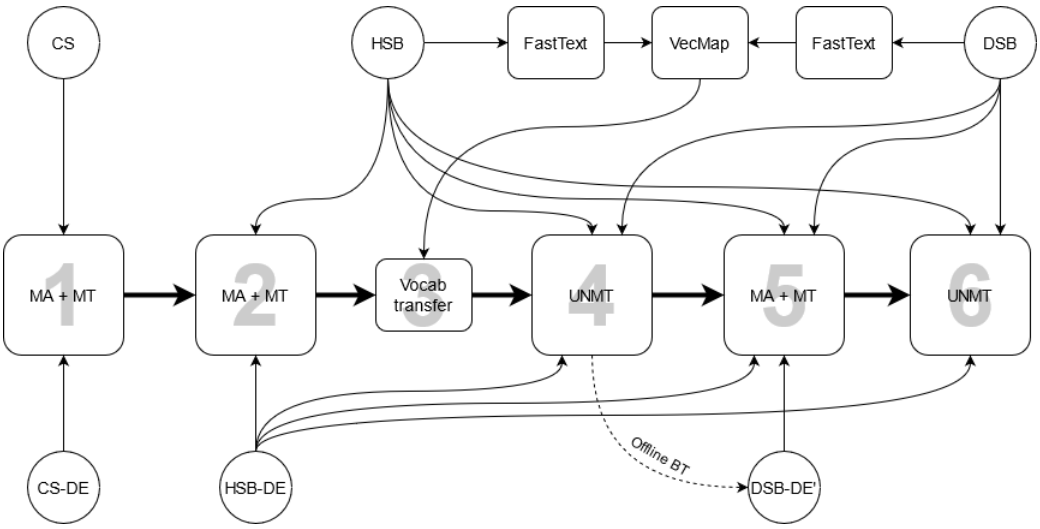


Figure 1: Diagram of the training steps. The circular nodes are datasets, the larger boxes are training steps of the MASS model, and the smaller boxes represent the steps for vocabulary transfer. German monolingual data is not shown as it is used in every step.

Data	Training Step	DE→DSB	DSB→DE
Without MPC	4	22.46	30.04
	5	**24.92**	28.48
	6	23.22	31.34
With MPC	4	23.03	30.70
	5	23.62	24.92
	6	22.95	**32.06**

Table 2: BLEU scores for our submitted models at various training steps. Best scores are in bold, and underlined scores are the models submitted to OCELoT, the submission system used for the shared task.[6] We compare performances of models trained with or without MultiParaCrawl (MPC) data.

due to the large amount of data. The training of fastText and VecMap and application of vocabulary transfer take less than an hour, and the offline BT for DSB→DE takes around 1 hour.

We use an additional stopping criterion of no improvement on the validation set in 1 million iterations. The metric we use for measuring improvement is the into-DE BLEU score, with the exception of step 6, in which we use the into-DSB BLEU score.

The hyperparameters used follow those used in Song et al. (2019), except for the epoch size being set at 100 thousand steps, rather than 200 thousand steps [7]. We shorten this as it saves systems more often and early stopping is applied more quickly. Our code is made freely available.[8]

Model	DE->DSB	DSB->DE
Multilingual	19.70	25.29
Multilingual + VT	20.30	28.14
Ours	19.25	26.00
Ours + VT	**22.46**	**30.04**

Table 3: BLEU scores comparing the two-language versus multilingual training schemes, with and without vocabulary transfer (VT).

4 Results

Table 2 shows our BLEU scores for our submitted models DE–DSB, starting from step 4. As we can see, the 5th step of MT using the synthetic data obtained from offline BT scores the best for DE→DSB, as such we used our models at this step for our submissions for this direction. For DSB→DE, we see the BLEU score actually drops for step 5, but the second phase of UNMT training improves the BLEU by roughly 1 point over step 4.

4.1 Pretraining Two Languages at a Time

In our first research question, we asked if it is best to train on two languages at a time. To answer this, we compare our model (without the MPC data) to another model trained on all 4 of our languages from the start. Specifically, we do MA for all languages, and MT for those with parallel data. We train this for 2 days, so that it is trained for the same length as steps 1 and 2. We then train using

[6] https://ocelot-west-europe.azurewebsites.net/

[7] The implementation we use, which is based on the MASS implementation (https://github.com/microsoft/MASS), defines epochs in steps.

[8] https://github.com/Leukas/WMT21

Model	DE→CS	CS→DE	DE→HSB	HSB→DE
Multilingual	11.56	14.36	49.22	49.14
Ours (step 1)	**15.05**	**16.16**		
Ours (step 2)			**52.14**	**51.57**

Table 4: BLEU scores of the auxiliary directions during pretraining.

Transfer Method	DE→DSB	DSB→DE
None	19.25	26.00
Simple	**22.64**	29.98
Levenshtein	22.46	**30.04**

Table 5: BLEU scores comparing no vocabulary transfer to our 2 methods.

Transfer Method	DSB→DE
None	3.15
Simple	19.00
Levenshtein	**21.27**

Table 6: Comparison of our model from step 2 with and without vocabulary transfer.

the UNMT objectives. The results are shown in Table 3.

Despite the model being exposed to DSB for longer, the performance is equal compared to our model without vocabulary transfer, and worse compared to the model with vocabulary transfer.

Without vocabulary transfer, the results being equal shows that exposing the model to DSB early within the training is not important to the final performance. We expect this might be due to the limited training data of DSB, as Wu and Dredze (2020) similarly found that the multilingual model mBERT performed more poorly on languages lacking in monolingual data.

With vocabulary transfer, we see that the model that has not seen DSB at all benefits more from the initialization than the model which has. We believe this shows the underlying advantage of training a model on few languages at a time, as it develops a better internal representation for the auxiliary languages, which enables better transfer. Table 4 shows that the performance of our model on the auxiliary languages is better when it can focus on learning one language pair at a time.

Moreover, while the multilingual model could learn to mimic its internal representation of HSB when encoding DSB, its representation according to Table 4 is poorer, and our model with vocabulary transfer explicitly copies the only language-dependent information the model receives, forcing an internal representation of DSB based on that of HSB.

4.2 Vocabulary Transfer Analysis

We also conduct an ablation of our novel method of vocabulary transfer. Table 5 shows the results of step 4, with and without vocabulary transfer. We also show results for a simpler transfer method: rather than taking the top 10 most similar candidates and choosing based on Levenshtein distance, we simply select the most similar candidate.

The addition of vocabulary transfer adds over 3 BLEU to the performance. We also see both transfer methods are competitive with each other in terms of improvement to performance. We expect the simple method to work better for language pairs with less similar spelling than the Sorbian languages. However our following analysis leads us to believe the Levenshtein version may perform better for similar languages.

We also perform a form of "zero-shot" transfer, where we use the model from Step 2 and test its ability to translate DSB→DE, despite the neural model never being trained on DSB at this stage. We contrast that with applying our 2 transfer methods. The results are in Table 6.

Without vocabulary transfer, the model expectedly has trouble with translation as it has not yet seen any DSB, so its vocabulary is not properly initialized. However with vocabulary transfer, we see an improvement of 16-18 BLEU, with the Levenshtein version performing best. This shows the degree to which a good initialization of the word embeddings can play a role in the overall performance of the model on an unseen language. Although UNMT (step 3) helps narrow the gap in performance, the difference of 3 BLEU also shows that unsupervised training can also stand to benefit from a better vocabulary initialization.

4.3 Back-translation

Our final research question concerns the order of back-translation. Garcia et al. (2020) found that, using a multilingual model trained on a hub language (e.g. German), one can achieve noticeable improvement by first zero-shot translating into the hub language (e.g. DSB→DE as done in step 5), then using this synthetic data for MT training. This can be followed by UNMT training (which includes online BT) for further improvement. We compare this method with the reverse, where we do UNMT training before offline BT, with the assumption that

Back-translation	DE→DSB	DSB→DE
Offline	21.74	22.74
Offline ⇒ Online	21.88	**30.14**
Online	22.46	30.04
Online ⇒ Offline	**24.64**	27.89

Table 7: BLEU scores for the different back-translation methods.

the better-quality translation after first training with online BT will result in better MT training. We show the results in Table 7.

As we can see, the performance of the model using only offline BT produces lower quality translations compared to using only online BT. While following offline BT with online BT makes up the difference in performance into DE, it still performs much worse into DSB. This supports our assumption that the better quality synthetic data leads to better MT training, as the main goal of creating synthetic DE data is to improve training with DSB on the target side.

5 Conclusion

The translation of Lower Sorbian to and from German presents a unique challenge in the field of unsupervised MT, due to the absence of parallel data and the scarcity of monolingual data for training. Therefore, the task necessitates an initial pretraining with similar, higher-resource languages. With this assumption, we experimented with various methods of pretraining, positing that training on 2 languages at a time is competitive with training with all languages at once, while allowing for a better initialization of DSB.

We also showcase a new method for transferring knowledge to the word embeddings of a transformer, provided a similar language is used in pretraining. We intend to experiment with this method further to gauge its applicability for more distantly-related languages. Finally, the use of both online and offline back-translation can improve the performance of a model, and if not done in an iterative fashion, the order in which they are performed can greatly affect the results.

Acknowledgments

We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297*.

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.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2020. Data selection for unsupervised translation of german–upper sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1099–1103.

Xavier Garcia, Aditya Siddhant, Orhan Firat, and Ankur P Parikh. 2020. Harnessing multilinguality in unsupervised machine translation for rare languages. *arXiv preprint arXiv:2009.11201*.

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.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Zuchao Li, Hai Zhao, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020. Reference language based unsupervised neural machine translation. *arXiv preprint arXiv:2004.02127*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Jörg Tiedemann and Lars Nygaard. 2004. The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC*. Citeseer.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? *arXiv preprint arXiv:2005.09093*.

The LMU Munich Systems for the WMT21 Unsupervised and Very Low-Resource Translation Task

Jindřich Libovický and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
{libovicky,fraser}@cis.lmu.de

Abstract

We present our submissions to the WMT21 shared task in Unsupervised and Very Low-Resource machine translation between German and Upper Sorbian, German and Lower Sorbian, and Russian and Chuvash. Our low-resource systems (German↔Upper Sorbian, Russian↔Chuvash) are pre-trained on high-resource pairs of related languages. We fine-tune those systems using the available authentic parallel data and improve by iterated back-translation. The unsupervised German↔Lower Sorbian system is initialized by the best Upper Sorbian system and improved by iterated back-translation using monolingual data only.

1 Introduction

In this paper, we describe systems for translation between German (*de*) and Upper Sorbian (*hsb*), German (*de*) and Lower Sorbian (*dsb*), and Russian (*ru*) and Chuvash (*cv*) developed at LMU Munich for the WMT21 shared task on unsupervised and very low resource machine translation (MT).

Upper Sorbian is a minority language spoken by around 30,000 people in today's German federal state of Saxony, Lower Sorbian has around 7,000 speakers and is spoken in the German federal state of Brandenburg. With such a small number of speakers, machine translation and automatic processing of Sorbian language is an inherently low-resource problem without any chance that the resources available for Sorbian would ever approach the size of resources for languages spoken by millions of people. On the other hand, being Western Slavic languages related to Czech and Polish, it is possible to take advantage of relatively rich resources collected for these two languages.

Unlike our last year's submission for Upper Sorbian (Libovický et al., 2020), we decided not to use synthetic data from unsupervised translation between Czech and Upper Sorbian and only did iterative back-translation. Despite having more authentic parallel data than last year, our system reaches approximately the same translation quality. Our Upper Sorbian systems ranked third out of six systems in the official ranking.

We leverage the relatedness between the Sorbian languages and use the Upper Sorbian system as a starting point for iterative back-translation using monolingual data only. Our Lower Sorbian Systems ranked second (de→dsb) and third (dbs→de) out of four teams in the official ranking.

Chuvash is a minority language spoken in the Volga region in the southwest of Russia. Although it uses the Cyrillic script, it is not related to eastern Slavic languages, but it is a Turkic language, relatively isolated in the Turkic language family. As a language with the highest number of speakers in this shared task, it also has the highest amount of available parallel data. We adopt a similar approach as for German-Upper Sorbian translation and pre-train our models on the related Kazakh language. In addition, we experiment with character-level models in the hope that they will be particularly effective for agglutinative morphology.

2 Experimental Setup

Most of our experimental setup is shared across all the language pairs. All our models use the Transformer architecture (Vaswani et al., 2017) as implemented in FairSeq (Ott et al., 2019).

All data is segmented using BPE (Sennrich et al., 2016b) with 16k merge operations as implemented in YouTokenToMe[1] without previous explicit tokenization. The merges are computed using a concatenation of all training data: German, Czech, Upper and Lower Sorbian in the first set of experiments, Russian, Kazakh, and Chuvash in the second set of experiments.

For the supervised task, we first pre-train mod-

[1] https://github.com/VKCOM/YouTokenToMe

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 989–994

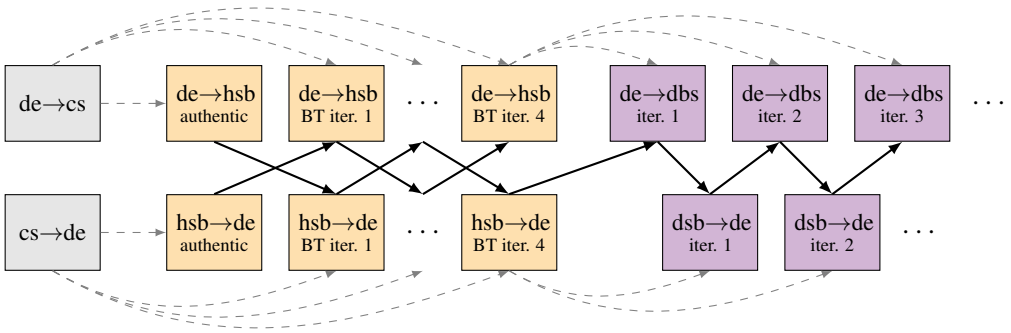


Figure 1: A diagram of the training procedure of the German↔Upper/Lower Sorbian systems. Gray dashed arrows (- - ▸) denote model initialization, solid black arrows (——▸) denote syntetic data generation by back-translation.

els on high-resource related languages: Russian-Kazakh for Chuvash and German-Czech for Upper Sorbian. We first train Transformer Base models on authentic data. These systems are used to generate back-translation (Sennrich et al., 2016a) of monolingual data. Using tagged back-translation (Caswell et al., 2019), we trained Transformer Big models for German↔Czech and Russian↔Kazakh translation. All back-translation steps use sampling and simple length-based filtering as proposed by Edunov et al. (2018)[2]. We upsample the authentic parallel data to match the size of the synthetic data.

We keep most default hyperparameters from the predefined architectures in FairSeq (`transformer` for the Base model, `transformer_wmt_en_de_big_t2t` for the Big model. The batch size is 6k tokens for the Base models, 2k tokens for Big models on a single GPU, Because we always start with high-resource training, we keep the dropout on the standard value of 0.1.

We use these models to initialize the weights (Nguyen and Chiang, 2017; Kocmi and Bojar, 2018) of the supervised low-resource models without restarting the optimizer. Because the learning rate is already low at that stage of training, we do not need to change the dropout to prevent overfitting. First, we train the supervised models using the authentic parallel data only, then we continue with iterated back-translation. The best Upper Sorbian-to-German model is used to translate Lower Sorbian monolingual data into German. In the next steps, we continue with a standard iterative back-translation procedure for unsupervised neural machine translation (Artetxe et al., 2018; Lample et al., 2018).

[2]We re-used the published code `https://github.com/pytorch/fairseq/tree/master/examples/backtranslation`.

Our final submission is an ensemble (with the vote strategy) of the best-scoring systems in the process of iterated back-translation. Language-pair-specific descriptions and results are discussed in the following sections.

We evaluate our systems using the BLEU Score (Papineni et al., 2002), chrF score (Popović, 2015) as implemented in SacreBLEU (Post, 2018).[3] Further, we evaluate the models using BERTScore (Zhang et al., 2020)[4] with XLM-RoBERTa Large (Conneau et al., 2020) as an underlying model for German and Russian and mBERT (Devlin et al., 2019) for Chuvash. Similar to the official task evaluation, we also report for each system the number of significantly worse systems in each metric at the significance level 0.95 with bootstrap resampling (Koehn, 2004) with 1k samples. For each metric, each system receives one point for each system it significantly outperforms in the metric at the significance level of 0.95.

3 German ↔ Upper Sorbian

Pre-training. For training the German↔Czech systems, we followed the same setup as in our last year's submission (Libovický et al., 2020). We used all parallel datasets from the Opus project (Tiedemann, 2012), which was 15.4M sentences after filtering by length and language identity. We trained a Transformer Base model on this data and used this model to generate back-translation. We used 20M Czech and 20M German sentences from the WMT News Crawl. We mix the back-translated and authentic parallel data one-to-one and train Transformer Big models on it.

[3]BLEU score signature `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0` chrF score signature `nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.0.0`

[4]`https://github.com/Tiiiger/bert_score`

	hsb → de				de → hsb		
	BLEU	chrF	BERTScore	Points	BLEU	chrF	Points
Authentic data only	53.4 [0]	.763 [0]	.933 [0]	[0]	54.9 [0]	.769 [0]	[0]
BT iter 1	55.2 [0]	.773 [0]	.936 [1]	[1]	56.4 [0]	.778 [0]	[0]
BT iter 2	55.8 [1]	.777 [1]	.937 [2]	[4]	56.5 [0]	.778 [0]	[0]
BT iter 3	55.8 [1]	.777 [1]	.937 [3]	[5]	56.2 [0]	.778 [0]	[0]
BT iter 4	56.1 [1]	.779 [1]	.938 [5]	[7]	56.0 [0]	.776 [0]	[0]
Ensemble	56.2 [1]	.779 [1]	.938 [4]	[6]	56.4 [0]	.779 [0]	[0]

Table 1: Quantitative results of the German↔Upper Sorbian translation systems on the development test data.

Sorbian data. We used all Upper Sorbian data provided for the shared task, i.e., 148k parallel sentence pairs (this is 88k sentence pairs more than last year), we did not apply any filtering on the parallel dataset. The development validation and the development test set of 2k sentences were the same as the last year.

Back-translation. We used 15M German sentences from the WMT News Crawl and all available monolingual Upper Sorbian data, 696k sentences, for back-translation. We applied the same rule-based statistical fixing of hyphenation-related OCR errors as the last year (Libovický et al., 2020, § 3.1). To better leverage the limited amount of monolingual data, we sample the Upper Sorbian translations 5×. We iterated the back-translation 4 times, always initializing the model with the Czech-German models (see Figure 1).

Results. The results are presented in Table 1. In the translation direction into German, the translation quality gradually increased between the back-translation steps. In the opposite direction, the translation quality oscillated. We attribute this to a larger amount of authentic German sentences. Ensembling only has a negligible effect. Note also that for translation into Sorbian, no differences between the models are statistically significant. In the opposite direction, the BLEU and the chrF score only separate the systems into two clusters, whereas the differences among BERTScores are always significant in the bootstrap testing, even though the absolute score differences are smaller. The best system for translation into German is a single from the last iteration of back-translation despite scoring slightly worse in the BLEU score.

4 German ↔ Lower Sorbian

Data. Because this is a purely unsupervised task, we did not use any Lower Sorbian parallel data.

		BLEU	chrF	BERTScore
dsb→de	Single	33.7	.606	.873
	Ensemble	33.8	.602	.874
de→dsb	Single	30.1	.587	—
	Ensemble	30.1	.588	—

Table 2: Automatic scores for the best German↔Lower Sorbian Systems.

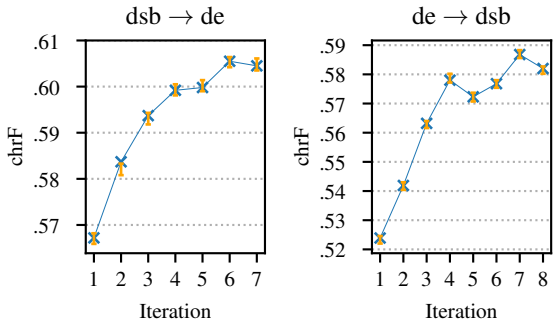


Figure 2: chrF scores during iterative back-translation for unsupervised German↔Lower Sorbian translation. The orange vertical lines denote 95%-confidence intervals using bootstrap resampling.

We used the same German monolingual data as we used for back-translation for Upper Sorbian. We use all the Lower Sorbian monolingual data, 145k sentences, provided by the organizers.

Iterative back-translation. Similarly to Upper Sorbian, we sample the back-translation of Lower Sorbian 10× for higher diversity in the training data.

Results. The final results are tabulated in Table 2. Figure 2 shows the translation quality in terms of chrF score during back-translation iterations. Similar to Upper Sorbian, the direction into German that uses larger monolingual data tends to improve more smoothly than the opposite direction. Also, the ensembling of the three best-scoring systems only has a negligible effect. The single system and

	cv → ru				ru → cv			
	BLEU	chrF	BERTScore	Points	BLEU	chrF	BERTScore	Points
Authentic data only	20.5 [2]	.451 [2]	.847 [3]	[7]	18.4 [0]	.486 [2]	.854 [3]	[5]
BT iteration 1	19.1 [0]	.443 [2]	.846 [2]	[4]	18.6 [0]	.487 [2]	.854 [4]	[6]
BT iteration 2	20.3 [2]	.450 [2]	.848 [4]	[8]	18.5 [0]	.487 [2]	.854 [2]	[4]
Ensemble of the two above	20.0 [2]	.450 [2]	.848 [4]	[8]	18.8 [1]	.489 [2]	.855 [5]	[8]
BT iteration 1 to char	18.0 [0]	.423 [0]	.843 [1]	[1]	16.9 [0]	.457 [0]	.850 [0]	[0]
BT iteration 2 to char	17.4 [0]	.420 [0]	.841 [0]	[0]	17.1 [0]	.463 [0]	.851 [1]	[1]
Ensemble of the two above	20.0 [2]	.450 [2]	.848 [4]	[8]	18.9 [1]	.490 [2]	.855 [5]	[8]

Table 3: Quantitative results of the Russian↔Chuvash translation systems on the development test data.

the ensemble do not significantly differ in any of the metrics.

5 Russian ↔ Chuvash

Pre-training. Similar to Upper Sorbian systems, we pre-train the systems on high-resource related language pair, Kazakh-Russian. We used the crawled Kazakh-Russian corpus of 5M sentence pairs published for WMT19 (Barrault et al., 2019) to train a Transformer Base model. We used these models to back-translation 3M Kazakh and 3M Russian sentences from the WMT News Crawl from the most recent years.

Chuvash data. We used all parallel data provided by the organizers, 717k sentence pairs, without any filtering. For back-translation, we used all 2.8M monolingual Chuvash sentences provided for the competition. For Russian, we used 18M monolingual sentences from the WMT News Crawl.

Back-translation. We ran two iterations of back-translation. We sample from the model during back-translation. We sampled 4 different translations for each Chuvash sentence to increase the training data diversity. We mix the authentic and synthetic parallel training data in the one-to-one ratio. All models are initialized by the Russian↔Kazakh models.

Character models. We further experiment with finetuning the system to the character level. Libovický and Fraser (2020) managed to train a character-level system for another Turkic language, English-to-Turkish translation. Here, we test if this is a property of Turkic languages or an artifact of the dataset English-Turkish dataset. We follow Libovický and Fraser (2020) and finetune the subword model to the character level.

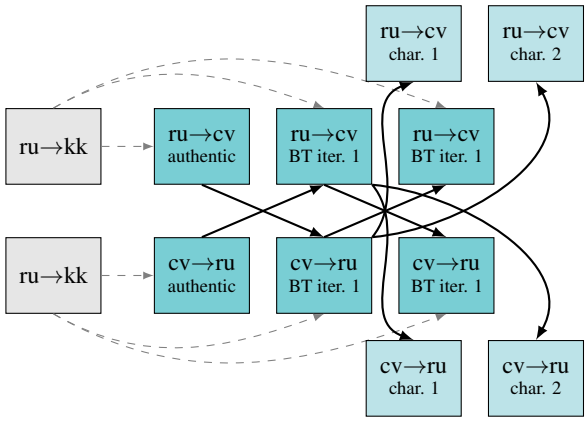


Figure 3: A diagram of the training procedure of the Russian↔Chuvash. Gray dashed arrows (- - ▸) denote model initialization, solid black arrows (⟶) denote syntetic data generation by back-translation.

Results. The results are presented in Table 3. Compared to other language pairs, back-translation had a surprisingly small effect on the translation quality. We suspect this result might be due to errors in data processing or signalize a need for a better data filtering technique. Model ensembling has no effect here. The character-level systems are on average 2 BLEU points worse than their subword counterparts, which is consistent with the results of character-level models on high-resource languages (Libovický and Fraser, 2020). Surprisingly, the character-level models seem to have much larger gains from model ensembling than the subword-based models. In fact, the ensemble of the character-level models is statistically indistinguishable from the best subword-based models.

6 Conclusions

We presented our systems for low-resourced translation between German and Upper Sorbian, unsu-

pervised translation between German and Lower Sorbian, and translation between Chuvash and Russian.

Our systems used standard state-of-the-art techniques for low-resource and unsupervised machine translation but did not exhaust all available methods. Better results could be achieved using more monolingual data and by more careful filtering of the synthetic parallel data.

Acknowledgments

This work was also supported by the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement #640550) and by the DFG (grant FR 2829/4-1).

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jindřich Libovický and Alexander Fraser. 2020. Towards reasonably-sized character-level transformer NMT by finetuning subword systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2572–2579, Online. Association for Computational Linguistics.

Jindřich Libovický, Viktor Hangya, Helmut Schmid, and Alexander Fraser. 2020. The LMU Munich system for the WMT20 very low resource supervised MT task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1104–1111, Online. Association for Computational Linguistics.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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, Philadelphia, Pennsylvania, 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, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Training hyper-parameters

We use the following command line options for `fairseq-train` command in all experiments. For the Transformer Base models, we use the pre-defined `transformer` architecture, for Transformer Big, we use `transformer_wmt_en_de_big_t2t`. The batch size is 6000 tokens for the Base models and 2000 tokens for the Big models.

```
fairseq-train \
  $DATA \
  --arch $ARCHITECTURE \
  --share-all-embeddings \
  --label-smoothing 0.1 \
  --criterion \
     label_smoothed_cross_entropy \
  --optimizer adam \
  --adam-betas '(0.9, 0.998)' \
  --clip-norm 5.0 \
  --lr 5e-4 \
  --lr-scheduler inverse_sqrt \
  --warmup-updates 16000 \
  --max-tokens $TOKENS
```

Language model pre-training and transfer learning for very low resource languages

Jyotsana Khatri[†], **Rudra Murthy V**[‡], **Pushpak Bhattacharyya**[†]
[†] Center for Indian Language Technology (CFILT)
Department of Computer Science and Engineering
IIT Bombay, India.
[‡]IBM Research, Bangalore, India.
{jyotsanak,pb}@cse.iitb.ac.in, rmurthyv@in.ibm.com

Abstract

This paper describes our submission for the shared task on Unsupervised MT and Very Low Resource Supervised MT at WMT 2021. We submitted systems for two language pairs: German ↔ Upper Sorbian (de ↔ hsb) and German ↔ Lower Sorbian (de ↔ dsb). For de ↔ hsb, we pretrain our system using MASS (Masked Sequence to Sequence) objective and then finetune using iterative back-translation. We perform final finetuning using the provided parallel data for translation objective. For de ↔ dsb, no parallel data is provided in the task, we use final de ↔ hsb model as initialization of the de ↔ dsb model and train it further using iterative back-translation, using the same vocabulary as used in the de ↔ hsb model.

1 Introduction

Transformer based architecture (Vaswani et al., 2017) has become the de-facto approach for training NMT models. These models have achieved good performance for resource rich languages. NMT models are usually data hungry and require lot of parallel data to get trained. However, many low-resource languages have very little or no parallel data to train a NMT model. For low resource language pairs, unsupervised MT (Artetxe et al., 2018; Lample et al., 2018; Lample and Conneau, 2019; Song et al., 2019), and transfer learning (Zoph et al., 2016a) have proven to be helpful in improving the translation performance. Unsupervised MT has gained a lot of attention in the past 3 years as it utilizes only monolingual data to train a NMT system.

In this paper, we present our system for shared task on Unsupervised MT and Very Low Resource Supervised MT at WMT2021. The task covers three languages pairs German (de) ↔ Lower Sorbian (dsb), German (de) ↔ Upper Sorbian (hsb), and Russian (ru) ↔ Chuvash (ch). We submitted systems for de ↔ hsb and de ↔ dsb. For de ↔ dsb there is no parallel data provided but for de ↔ hsb, there is small parallel data.

Summary of our submitted systems:

- We use language model pretraining using MASS (Song et al., 2019) objective to pretrain a model for de ↔ hsb using shared encoder, shared decoder, and shared vocabulary, which is followed by finetuning using iterative back-translation. The final model is finetuned using parallel data with translation objective.
- For de ↔ dsb, our model is trained using provided monolingual dsb and de data using iterative back-translation. The model is initialized using the final model of de ↔ hsb.

2 Related Work

Supervised NMT using transformer based architectures (Vaswani et al., 2017) has achieved high translation accuracy for high resource languages like English-French and English-German. Supervised NMT requires lots of parallel data to get trained. For low resource languages (which does not have large amount of parallel data) the performance of NMT systems is usually poor. We briefly describe some literature on Unsupervised MT and transfer learning.

2.1 Unsupervised NMT

Unsupervised MT gained quite a lot of attention of researchers because of its ability to train MT system without using any parallel data. The research in Unsupervised MT started with techniques which are based on statistical decipherment (Ravi and Knight, 2011; Dou and Knight, 2012, 2013; Dou et al., 2014, 2015). The approaches proposed in Artetxe et al. (2018); Lample et al. (2017) are majorly based on unsupervised cross-lingual embeddings, denoising auto-encoders, and iterative

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 995–998

back-translation. Later, some approaches of Unsupervised SMT have been proposed where a phrase table is constructed using bilingual embeddings and training is performed using language model and distortion model (Artetxe et al., 2019; Lample et al., 2018).

State of the art approaches of Unsupervised NMT are based on cross-lingual language model pretraining followed by iterative back-translation (Lample and Conneau, 2019; Song et al., 2019; Lewis et al., 2019). All these models differ with respect to pretraining objective. Lample and Conneau (2019) pretrain encoder and decoder separately using masked language modeling objective, while Song et al. (2019) pretrains encoder and decoder together using MASS (masked sequence to sequence) objective. Lewis et al. (2019) pretrains encoder and decoder using an objective similar to MASS but here the decoder is supposed to predict the whole sentence rather than only predicting the masked span of tokens.

2.2 Transfer Learning

Transfer learning have proven to be helpful for low resource languages (Zoph et al., 2016b; Dabre et al., 2017; Nguyen and Chiang, 2017). In Zoph et al. (2016b), authors use a model trained on one language pair as the initialization of the model for another language pair, they do not consider or do anything with the vocabulary. Gheini and May (2019) proposed to create a universal vocabulary before starting the training of the parent model. The transfer learning works best if the language pairs are related (Dabre et al., 2017). Aji et al. (2020) shows that the internal layers are most important in transfer learning.

3 System Overview

In this section, we describe the details of the submitted systems to shared task on Unsupervised MT and Very Low Resource Supervised MT at WMT 2021. We report results for our 2 types of models:

- **Language model pretraining using MASS objective**: For de ↔ hsb, we pretrain our model using MASS objective and then finetune it using iterative back-translation. Final finetuning is performed using parallel data of de ↔ hsb provided in the task.
- **Transfer learning**: For de ↔ dsb, we use the final model of de ↔ hsb to initialize the model of de ↔ dsb and train it further using iterative back-translation using monolingual data of de and dsb.

To train our models, we use shared encoder-decoder transformer architecture. We also use shared vocabulary of both source and target languages. For de ↔ dsb, we use the same vocabulary as used in de ↔ hsb model without considering the vocabulary mismatch.

4 Experiments

In this section, we describe the experimental setup and the hyper-parameters used.

4.1 Data and Preprocessing

For de ↔ hsb, we use monolingual data of hsb provided in the task and we use a subset (equal to the size of the hsb monolingual data) of news-crawl-2020 dataset downloaded from WMT[1] provided in the WMT news translation task for de monolingual data, and also use the parallel data provided in the task. For de ↔ dsb, we use monolingual data of dsb provided in the task together with a subset (equal to the size of the dsb data) of news-crawl-2020 dataset provided in WMT news translation task for de monolingual data.

We tokenize using Moses tokenizer (Koehn et al., 2007). We use fastBPE[2] to learn BPE (Byte pair encoding) (Bojanowski et al., 2017) with 32k BPE codes over the combined tokenized data of both languages. For de ↔ dsb, we use the same vocabulary and codes learnt for de ↔ hsb.

4.2 Experimental Setup

We use 6 layers in the encoder and decoder with 8 attention heads and 1024 embedding dimension. We use Adam (Kingma and Ba, 2015) optimizer. We use, a warm-up phase of 4000 steps with initial learning rate starting from $1e^{-7}$ to $1e^{-4}$, in the warm-up phase learning rate is increased linearly and then starts to decrease with inverse square root learning rate schedule. We use mini-batches of size 2000 tokens and set the dropout to 0.1 (Gal and Ghahramani, 2016). Maximum sentence length is set to 100 after applying BPE. At the time of decoding, we set beam size to 1. For experiments, we are using MASS[3] codebase.

[1] http://statmt.org/wmt21/translation-task.html

[2] https://github.com/glample/fastBPE

[3] https://github.com/microsoft/MASS

The pretraining is performed for 100 epochs for both de ↔ hsb. de ↔ hsb model is further finetuned using iterative back-translation for 60 epochs and then trained using parallel data for 60 epochs. de ↔ dsb model is further finetuned for iterative back-translation using the final de ↔ hsb model for 60 epochs. Epoch size is set to .2M sentences.

4.3 Results and Discussion

Lang Pair	Train	Valid	Test
de-hsb	147521	2000	2000
de-dsb	0	601	602

Table 1: Parallel data (Number of sentences)

Language	Train
hsb	695721
dsb	145198

Table 2: Monolingual data (Number of sentences) (We use equal amount of german data from news-crawl2020 as dsb and hsb to train their respective models)

Lang Pair	Our system	Best system
de-hsb	60.2	66.3
hsb-de	60.1	67.7
de-dsb	6.4	29.9
dsb-de	5.9	33.5

Table 3: Results: BLEU scores for our system and highest scoring system in the task

All the results are shown in 3. We achieve BLEU score of 60.2 and 60.1 for de → hsb and hsb → de respectively. Using the final model of de ↔ hsb as initialization of the model for de ↔ dsb, we achieve BLEU score of 6.4 and 5.9 for de → dsb and dsb → de respectively even with using the same vocabulary of de ↔ hsb. The percentage of vocabulary overlap (the percentage of de ↔ dsb vocabulary that is present in de ↔ hsb vocabulary) is 68.21 after applying BPE which makes the transfer learning work. After MASS pretraining and iterative back-translation (without using any parallel data), the BLEU scores are 4.74 and 4.92 for de → hsb and hsb → de respectively. We are able to achieve above BLEU scores without using any parallel data because of the similarity between de and hsb. The percentage of vocabulary overlap between de and hsb (the percentage of vocabulary of de present in hsb) is 60.73, which makes them highly similar languages. Similarly, the percentage of vocabulary overlap between de and dsb (the percentage of vocabulary of de present in dsb) is 54.26. The vocabulary here refers to the number of unique tokens after applying BPE.

5 Conclusion

In this paper, we study the impact of language model pretraining together with iterative back-translation for very low resource language pair i.e. de ↔ hsb. We also study the impact of transfer learning from de ↔ hsb to de ↔ dsb. In future, we plan to filter bad back-translated data while training for de ↔ dsb using iterative back-translation and also different transfer learning techniques to improve the performance for de ↔ dsb.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. *ArXiv*, abs/1710.11041.

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.

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.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages

266–275, Jeju Island, Korea. Association for Computational Linguistics.

Qing Dou and Kevin Knight. 2013. Dependency-based decipherment for resource-limited machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1668–1676, Seattle, Washington, USA. Association for Computational Linguistics.

Qing Dou, Ashish Vaswani, and Kevin Knight. 2014. Beyond parallel data: Joint word alignment and decipherment improves machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 557–565.

Qing Dou, Ashish Vaswani, Kevin Knight, and Chris Dyer. 2015. Unifying Bayesian inference and vector space models for improved decipherment. 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 836–845, Beijing, China. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS, pages 1027–1035, Red Hook, NY, USA. Curran Associates Inc.

Mozhdeh Gheini and Jonathan May. 2019. A universal parent model for low-resource neural machine translation transfer. *arXiv preprint arXiv:1909.06516*.

Diederik P. Kingma and Jimmy Ba. 2015. 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 association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. 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. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

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.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

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. 2016a. Transfer learning for low-resource neural machine translation. *ArXiv*, abs/1604.02201.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016b. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

NRC-CNRC Systems for Upper Sorbian–German and Lower Sorbian–German Machine Translation 2021

Rebecca Knowles* and **Samuel Larkin***
National Research Council Canada
{Rebecca.Knowles, Samuel.Larkin}@nrc-cnrc.gc.ca

Abstract

We describe our neural machine translation systems for the 2021 shared task on Unsupervised and Very Low Resource Supervised MT, translating between Upper Sorbian and German (low-resource) and between Lower Sorbian and German (unsupervised). The systems incorporated data filtering, backtranslation, BPE-dropout, ensembling, and transfer learning from high(er)-resource languages. As measured by automatic metrics, our systems showed strong performance, consistently placing first or tied for first across most metrics and translation directions.

1 Introduction

This work describes our machine translation (MT) systems for translating between Upper Sorbian–German and Lower Sorbian–German (all translation directions). We focused primarily on the supervised task of Upper Sorbian–German, and then applied those systems to the task of building simple Lower Sorbian–German systems.[1]

Upper Sorbian and Lower Sorbian are Slavic minority languages spoken in eastern Germany, alongside German. The shared task data was provided to the organizers through collaborations with the Sorbian Institute[2] and the Witaj Language Centre,[3] as described in Fraser (2020), to which we direct interested readers for additional information on the languages and data. Following the 2020 shared task, the Witaj Language Centre released a publicly-available Sorbian–German MT system `sotra` (Witaj Language Centre, 2021) based on Moses (Koehn et al., 2007) and OpenNMT (Klein et al., 2017).[4]

We provide an overview of the data, preprocessing, and model architectures in Sections 2, 3, and 4. We then discuss baselines, systems, experiments in monolingual filtering, and backtranslation (all focused on Upper Sorbian–German) in Sections 5, 6, 7, and 8. In Section 9, we discuss how we applied and finetuned our existing Upper Sorbian MT systems for the task of translating Lower Sorbian. Section 10 discusses additional experiments with negative results. Finally, Sections 11 and 12 summarize the final systems and our conclusions.

2 Data

We used all provided parallel German–Upper Sorbian data and all monolingual Upper Sorbian data (after filtering), along with German–Czech parallel data from Open Subtitles (Lison and Tiedemann, 2016),[5] DGT (Tiedemann, 2012; Steinberger et al., 2012), JW300 (Agić and Vulić, 2019), Europarl v10 (Koehn, 2005), News-Commentary v15, and WMT-News.[6] We also used the monolingual Upper Sorbian Web, Witaj and Sorbian Institute datasets as well as the Lower Sorbian monolingual data (the latter for Lower Sorbian tasks only).[7] We used the provided `devel` sets for development, and the `devel_test` systems for measuring progress and choosing which systems to submit.

3 Preprocessing and Postprocessing

As preprocessing, we first clean all of the available training data (but not development or test data) using `clean-utf8-text.pl` with the `-no-phrase-sep` flag from `PortageTextProcessing`.[8] For parallel training data, we use `clean-corpus-n.perl`

*Both authors contributed equally to this work.

[1] We abbreviate language names as follows: *cs* (Czech), *de* (German), *dsb* (Lower Sorbian), and *hsb* (Upper Sorbian).

[2] https://www.serbski-institut.de/en/Institute/

[3] https://www.witaj-sprachzentrum.de/

[4] https://sotra.app

[5] http://www.opensubtitles.com

[6] http://www.statmt.org/wmt20/translation-task.html

[7] http://www.statmt.org/wmt21/unsup_and_very_low_res.html

[8] https://github.com/nrc-cnrc/PortageTextProcessing

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 999–1008
November 10–11, 2021.

Data	Lines	BPE	Voc.	CS-DE Parent	Multi.	Multi. ML-0
train.hsb-de.de	60,000	Y	Y		21 ×	36 ×
train.hsb-de.hsb	60,000	Y × 2	Y		21 ×	36 ×
train2021.hsb-de.de	87,521	Y	Y		21 ×	36 ×
train2021.hsb-de.hsb	87,521	Y × 2	Y		21 ×	36 ×
sorbian_institute_monolingual.hsb	337,730	Y × 2	Y		6 × <BT>	
web_monolingual.hsb	105,484				6 × <BT>	
witaj_monolingual.hsb	219,177	Y × 2	Y		6 × <BT>	
OpenSubtitles.cs-de.{de,cs}	11,073,440			Y +10× BPE-dr		
DGT.cs-de.{de,cs}	3,653,397			Y +10× BPE-dr		
JW300.{de,cs}	1,037,533			Y +10× BPE-dr		
Europarl.cs-de.{de,cs}	558,693	Y	Y	Y +10× BPE-dr	3 × <CS>	3 × <CS>
News-Commentary.cs-de.{de,cs}	180,053	Y	Y	Y +10× BPE-dr	3 × <CS>	3 × <CS>
WMT-News.cs-de.{de,cs}	19,892	Y	Y	Y +10× BPE-dr	3 × <CS>	3 × <CS>
news.2019.de.shuffled.deduped.de	31,650,966					
news-commentary-v15.dedup.de	226,820	Y	Y			
news.2019.de.shuffled.deduped.ml_t00	5,071,268					1 x <BT>

Table 1: Data and how it was used, whether for BPE training and vocabulary extraction, parent model training, or child model training. All numbers of lines reflect data after initial cleaning and filtering by known characters. Special tags (for language or backtranslation) are shown where they are used, upsampling is shown with ×, and BPE-dropout is shown.

from Moses (Koehn et al., 2007) with ratio 15, and for monolingual data we remove empty lines. We normalize punctuation with Moses's `normalize-punctuation.perl` and remove the non-breaking space `\xa0`. We perform additional sentence splitting to improve tokenization,[9] then tokenize with Moses's `tokenizer.perl -a -l $LNG` (where $LNG is cs, de, or hsb), then re-merge the sentences that were split into single lines. For all German-Czech parallel data and all monolingual German or Czech data, we removed any lines that contained characters that had not been observed in DE-HSB training data, WMT-News, or Europarl. This helps clean data of unusual encoding issues, as well as removing text that is clearly in other languages (i.e., written in other scripts).

We build BPE vocabularies of size 10k, 15k, 20k, and 25k merges using `subword-nmt`[10] (Sennrich et al., 2016). We also add all Moses and Sockeye special tags (ampersand, <unk>, etc.) and a number of additional reserved tags (for backtranslation, languages, etc.) to a glossary file used for applying BPE, which prevents them from being segmented. For building the BPE models, we used all HSB-DE data, the Sorbian Institute and Witaj monolingual HSB data, CS-DE data, and news-commentary (DE) data; the HSB data was upscaled twice (see Table 1 for full details). The same datasets were used for extracting the joint vocabulary, which was then used for source and target.

In standard postprocessing, we de-BPE and detokenize (using the Moses `detokenizer.perl -a -l $LNG`).

4 Models

We built Transformer models (Vaswani et al., 2017) using Sockeye (Hieber et al., 2018) version 2.3.14 and cuda-10.1. We used the default value of 6 encoder/decoder layers, 8 attention heads, the Adam (Kingma and Ba, 2015) optimizer, label smoothing of 0.1, a cross-entropy-without-softmax-output loss, and a model size of 512 units with a FFN size of 2048. We performed early stopping after 32 checkpoints without improvement. We chose custom checkpoint intervals of 4000 updates when the train corpus was deemed big enough and 500 updates when the train corpus was small. We optimized for BLEU (Papineni et al., 2002)[11] and used the whole validation set during validation. The batch size was set to 8192 tokens, and the maximum sequence length for both source and target was set to 200 tokens. We used weight tying and vocabulary sharing, but we set gradient clipping to absolute and kept the initial learning rate of 0.0002. We used a beam size of 5 in all submit-

[9] Using `utokenize.pl -p -ss -notok -paraline -lang=en` from `PortageTextProcessing`.

[10] https://github.com/rsennrich/subword-nmt

[11] All BLEU scores were computed using `sacreBLEU` (Post, 2018) with the signature `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14`. The chrF scores (Popović, 2015) were generated in the submission interface.

ted systems.[12] When systems deviate from these (i.e., different learning rates or label smoothing), we make note of it in our descriptions.

5 Baselines

We build several types of baselines against which to measure our improvements. They are shown in Table 2 and discussed in the following sections.

System	DE-HSB	HSB-DE
Translation Memory	16.3	15.5
2020 Bitext Baseline	47.0 (10k)	45.7 (10k)
Bitext Baseline	53.2 (10k)	51.9 (15k)
2020 Final Submission	59.4 (20k)	58.9 (10k)

Table 2: BLEU scores for baseline systems measured on devel_test data (vocabulary size in parentheses).

5.1 Translation Memory

We build translation memory baselines, following Simard and Fujita (2012). For each source sentence in devel_test, we find the most similar (as measured by sentence-level BLEU) source sentence in the full training set and return its translation as our hypothesis. The relatively high scores obtained demonstrate the high levels of similarity between the devel_test and train domains (though we note that very few sentences are *exact* matches for ones in the training data).

5.2 Bitext Baselines

We build baselines using the available DE-HSB bitext, first with only the bitext available for the 2020 iteration of the task, and then with the 2020 and 2021 training data combined. As we see in the middle rows of Figure 2, the increase in data from 60,000 to 147,521 lines resulted BLEU score increases of +6.2 in both translation directions.

5.3 2020 Final Systems

As a last "baseline", we consider our final submissions to the 2020 shared task. In the DE-HSB direction, this was a four system ensemble, all of which were child systems incorporating backtranslation and built on top of parent systems trained on either DE-CS or DE-"pseudo-HSB", with a mix of types of BPE-dropout. The HSB-DE direction was an ensemble of 5 child systems using backtranslation on top of a similar set of parent systems. These are described in detail in Knowles et al. (2020).

[12] In paraphrasing experiments where we generated 10-best lists, we used a beam size of 10, but these did not contribute to our final systems.

6 Systems

Here we describe the general types of systems that we have built, including parent DE-CS systems, multilingual systems, and the child (and grandchild) systems we built on top of those.

6.1 Parent Systems

We first built DE-CS and CS-DE parent systems, using OpenSubtitles, DGT, JW300, Europarl, News-Commentary, and WMT-News as training data. The training data is used once in its original form, and concatenated with 10 different versions each generated by an iteration of BPE-dropout (both source and target)[13] with a dropout rate of 0.1. We use newstest2019-csde as the development set. This results in DE-CS and CS-DE systems with BLEU scores between 22 and 25 on the newstest2019-csde development set. We use these parent systems for transfer learning.

6.2 Multilingual Systems

When we build CS-DE and DE-CS parent systems and then use them for transfer learning by finetuning on HSB-DE or DE-HSB data, they undergo "catastrophic forgetting" (Thompson et al., 2019; Gu and Feng, 2020) and lose the ability to translate Czech while gaining the ability to translate Upper Sorbian, as measured on their respective development sets. While we don't necessarily *need* to maintain the ability to translate Czech, we explored whether multilingual systems might improve performance on our task of interest. To this end, we build multilingual systems, which incorporate CS-DE data, upsampled HSB-DE data, and backtranslated data (DE in the case of HSB-DE systems, HSB in the case of DE-HSB systems). In these systems we performed upsampling with the aim of having approximately 1 part CS-DE to 4 parts HSB-DE data (reflective of our priority to translate HSB-DE). We did not experiment with additional ratios; we leave this to future work.

For DE-HSB multilingual systems, we used monolingual HSB data (backtranslated and upsampled 6 times) tagged with <BT> tags, DE-CS data (Europarl, News-Commentary, WMT-News; upsampled 3 times) tagged with <CS>, and the parallel DE-HSB training data (upsampled 21 times and untagged). For HSB-DE multilingual systems,

[13] Note that we only apply BPE-dropout to training data, never to development or test data.

we used a sample[14] of the news 2019 DE data backtranslated and tagged with <BT>, the same three CS-DE corpora tagged with <CS> and upsampled, and the HSB-DE training data upsampled (the upsampling of these latter corpora depended on the size of the backtranslated data).

6.3 Child Systems

Child systems are initialized with the parameters of some given system and are then finetuned on a new set of data with continued training. This is how we perform transfer learning, taking a parent system trained on CS-DE (or DE-CS) data and converting it to an HSB-DE (or DE-HSB) system by starting from the parent system parameters and training on the appropriate language data, as in Kocmi and Bojar (2018). In some cases, we repeat this process multiple times with different sets of data, building "grandchild" systems on top of child systems.

7 Monolingual Data Filtering

Given the tight coupling between the domain of the development/development-test data and the training data, the large quantity of monolingual data available for backtranslation from German, and inspired by the filtering used in the high-performing 2020 submission by Scherrer et al. (2020) we examined whether we should subsample data for backtranslation.[15] We used the 2020 final systems described in Knowles et al. (2020) to backtranslate all available News 2019 DE. This enabled us to train child systems with a random sample of 1.5 million lines of text, the full available backtranslated data, and several approaches to sampling the data.

We first describe HSB-DE experiments with the fixed data size of 1.5 million lines of backtranslated DE monolingual data. We use a *random* sample as a baseline. We compare to it two approaches to using pretrained Sentence-Transformer embeddings[16] (Reimers and Gurevych, 2019) and cosine similarity for domain filtering: ranking sentences in the monolingual data based on their similarity to the *average* embedding of the full DE side of the DE-HSB training data and selecting the 1.5 million most similar,[17] and selecting the 1.5 million sentences most similar to any *individual* sentence in the DE side of the DE-HSB training data.[18] We also apply *Moore-Lewis* filtering (Moore and Lewis, 2010), again treating the DE side of the DE-HSB training data as the "in-domain" data. Moore-Lewis (M-L) uses language models[19] to compare out-of-domain data to in-domain data on the basis of cross-entropy, enabling the sampling of in-domain-like text from the out-of-domain set.

Filter	Both	Src.	None
Random	57.5	57.1	*57.3*
Average	57.5	*57.4*	57.1
Individual	**57.7**	56.9	57.1
Moore-Lewis	**57.7**	57.2	*57.3*
M-L thresh.: 0	**57.7**	**58.1**	**57.9**

Table 3: BLEU scores of HSB-DE child systems trained on authentic HSB-DE parallel text (upsampled) and 1.5 million lines of backtranslated (iteration 1) News 2019 data, sampled using different approaches. Results shown are for 15k vocabulary. Columns indicate type of BPE-dropout (both source and target, source only, and neither). The last line shows thresholded Moore-Lewis, with 5,071,268 lines selected.

We find that the Moore-Lewis approach outperforms or matches the random baseline across three variations of BPE-dropout. Table 3 shows results for 15k BPE, but we found the same across 10k, 15k, 20k, and 25k vocabularies. With both source and target BPE-dropout, the Moore-Lewis sample was always best or tied for best, with source side or no dropout it was always best or second best.

We also built systems with no BPE dropout, using full backtranslated News 2019 data; for larger BPE sizes, the Moore-Lewis samples outperformed the full data (despite being much smaller and thus more efficient), while for the smaller BPE sizes, Moore-Lewis came in second behind the full data.

With 1.5 million as a relatively arbitrary size, we proceeded with using a threshold for Moore-Lewis filtering. A threshold of 0 resulted in 5,071,268 lines sampled from News 2019. With upsampling

[14] As described in Section 7, Moore-Lewis filter.

[15] Data subsampling or filtering or of one sort or another was also used by several other submissions in 2020, including: Dutta et al. (2020), Edman et al. (2020), and Knowles et al. (2020).

[16] From https://github.com/UKPLab/sentence-transformers, with model *paraphrase-xlm-r-multilingual-v1*, a multilingual version of *paraphrase-distilroberta-base-v1*, trained on parallel data for 50+ languages (Reimers and Gurevych, 2020).

[17] Similar to the domain-cosine approach in Aharoni and Goldberg (2020).

[18] We note that this uses external pretrained models, and we have done this only for the purpose of experimenting with backtranslation; none of our final submissions are built using these approaches, so they remain constrained.

[19] 4-gram language models built with MITLM (https://github.com/mitlm/mitlm).

HSB-DE data to match it in size, we found that the Moore-Lewis threshold child models outperformed the 1.5 million size and also outperformed or matched the full data size systems.

We also tested Moore-Lewis filtering on the Upper Sorbian monolingual data, but found it to be less useful in that case, likely due to the much smaller size of available data, and potentially to closer matches (due to the shared origins of the training data and some of the monolingual data).

8 Backtranslation

8.1 BT1

For our first iteration of backtranslation, BT1, we use our final submitted systems from last year's task, as described in Section 5.3.

8.2 BT2

Our second iteration of backtranslation was performed using ensembles of (at the time) best-performing systems at the midpoint of the shared task. Keeping in mind that ensembles typically outperform single systems, and that we found that diverse ensembles seemed to outperform less diverse ensembles, we chose our best-performing systems and then two variants of each to ensemble for the second round of backtranslation. For DE-HSB, the child systems ensembled were the `both`, `src`, and `none` BPE-dropout variants trained on the true DE-HSB data (upsampled 3 times) and BT1 backtranslated HSB data, with a 25k vocabulary. For HSB-DE, the child systems ensembled were also `both`, `src`, and `none` BPE-dropout variants trained on the the true DE-HSB data (upsampled 35 times) and BT1 News 2019 DE data filtered using Moore-Lewis and a threshold of 0, with a 15k vocabulary.

8.3 Analysis of Backtranslation

We compared BT1 and BT2 outputs and found them to be quite similar, sometimes even identical. This brought us to a closer examination of the backtranslation systems and the training data itself. As part of our analysis and experiments, we performed backtranslation of the full DE-HSB training data.

Doing so, we observed that significant portions of the training data had been memorized by many of our systems, and where differences existed, they tended to be quite small. As evidence of this, for both BT1 and BT2, in both translation directions, the BLEU scores for backtranslated training data were 98.2 or higher. Nevertheless, the high automatic metric scores on held-out data suggest that these systems are still able to generalize (that is, they have not *only* memorized data), though it does raise questions about *how* general the models are: would they perform nearly as well on out-of-domain data?

9 Lower Sorbian

The data provided for Lower Sorbian consists of 145,196 lines of monolingual data and the small (approx. 600 line) parallel devel and devel_test sets. In order to build systems, we relied on the relatedness of Lower Sorbian and Upper Sorbian. Since we primarily focused on Upper Sorbian, our BPE vocabularies were *not* learned using Lower Sorbian; we leave an exploration of that to future work. Here we describe our process of building Lower Sorbian systems from Upper Sorbian systems.

9.1 Initial Round

Without any parallel data, we first tried simply translating with our existing HSB-DE and DE-HSB systems and ensembles. In the DE-DSB direction, the resulting devel_test DSB scores were between 7.7 and 8.1 BLEU, while in the DSB-DE direction, the scores were naturally a bit higher (since the system *has* trained on the output language of German), between 17 and 19 BLEU.

From there, we translated the full DSB monolingual data using one of our best HSB-DE single systems: 25k vocabulary, standard parent CS-DE (BPE-dropout both), finetuned child system using BT2 M-L threshold 0 news data and the original HSB-DE training data with BPE-dropout (both) and label smoothing of 0.15. The relatively high BLEU scores that we observed when translating DSB devel and devel_test data with HSB-DE systems allowed us to assume that the output might be more than just noise, and ideally at least good enough for use as the source side.

For backtranslation of DE into DSB, we used an ensemble of two 25k vocabulary DE-DSB systems. The first started from a default parent DE-CS system with BPE-dropout (both) and was then finetuned as a multilingual system using BT2 backtranslated HSB monolingual data and DE-CS data (as described in Section 6.2) with BPE-dropout (both). Then it was finetuned with the initial round backtranslated DSB monolingual data just described, again with BPE-dropout (both). The

second also started from the default DE-CS BPE-dropout (both) system and was finetuned with BT2 backtranslated HSB monolingual data with BPE-dropout (both) and learning rate of 0.0001. This was then also finetuned with initial round backtranslated DSB monolingual data, with BPE-dropout (both) and a learning rate of 0.0001.

9.2 Next Round

We also built another DSB-DE system for performing next round backtranslation. It began with a 20k vocabulary standard parent CS-DE (BPE-dropout both), as with previous systems, finetuned child system using BT2 M-L threshold 0 news data and the original HSB-DE training data with BPE-dropout (both) and label smoothing of 0.15. We then finetuned this with the DE side of the full HSB-DE training data, backtranslated to DSB using the initial round DE-DSB system.

10 Inconclusive and Negative Results

We now discuss negative results, i.e., experiments that we performed that were unsuccessful. All of these were performed on Upper Sorbian-German.

10.1 Fuzzy Matching

We performed brief and ultimately unsuccessful experiments with using similar translations from the training data to guide translation, as in Xu et al. (2020). In this approach, for each source sentence (train, development, or test), we first extract its best "fuzzy match" from a translation memory (we use the parallel HSB-DE data for this, and select the best *non-exact* fuzzy match) if any is available. The system input then consists of the source sentence, followed by a special token, followed by the target language sentence corresponding to the closest source fuzzy match from the translation memory (called **FM**$^{\#}$ in Xu et al. (2020)). We also tried an approach like their **FM*** approach, where target language tokens are masked with a special token if they do not align[20] to a source language word that is contained in the source sentence to be translated. In either case, if no fuzzy match is returned, a special null token replaces the target language text in the input. Both approaches performed almost identically to the baseline, so we did not proceed with additional experiments (including those approaches that used factors). We experimented with a range of thresholds for fuzzy matches (0.0, 0.35, 0.5), all using `FuzzyMatch-cli`[21] but all performed comparably. We believe this remains an open area for exploration: did the systems fail to outperform the baseline because the baseline had already attained high quality? Did the small size of the translation memory hurt performance?

10.2 Backtranslation as Paraphrasers

Inspired by work like Khayrallah et al. (2020), we also experimented with whether we could treat our high-quality backtranslation systems as paraphrasers to generate more diverse data by translating the HSB-DE parallel data (in each direction) with sampling rather than using one-best output of beam search. We tried building children with this data (both with only authentic target side data and with full combinations of sampled datasets), but did not find that it improved over comparable systems. One issue is that the HSB-DE training data is nearly memorized, as discussed in Section 8.3, so even in the sampled data, many of the differences between translations are quite small.

10.3 Backtranslation-Only Systems

Following Abdulmumin et al. (2021), we experimented with finetuning our parent systems using *only* backtranslated data, followed by then finetuning on the authentic parallel data. We had mixed results with this approach – one of them was high-performing enough to include in our HSB-DE final ensemble, but there was not enough evidence for this language pair to conclude that the approach is broadly useful (beyond providing additional diversity to ensembles).

11 Final Systems

According to preliminary automatic metric results from the shared task organizers, our systems performed quite well. The metrics considered were BLEU, chrF, and – in the case of translation into German – BERT Score. Each translation direction saw five systems submitted, with the exception of DSB-DE, which only had four. Our HSB-DE had the best BERT score (0.981), the second-best BLEU score (67.3, 0.4 BLEU behind NoahNMT), and the best chrF score; it was significantly better than all four other systems in terms of BERT score, while in terms of BLEU and chrF it was better than three other systems (tying with NoahNMT).

[20] We used `fast_align` (Dyer et al., 2013).

[21] https://github.com/SYSTRAN/fuzzy-match

	Test		devel_test	
System	**BLEU**	**chrF**	**BLEU**	**BLEU (sing.)**
HSB-DE	67.3 (-0.4)	0.836 (+0.002)	60.0	58.5
DE-HSB	66.3 (+0.4)	0.837 (+0.004)	59.9	58.1
DSB-DE	33.5 (+0.2)	0.638 (+0.016)	34.9	34.5
DE-DSB	29.9 (+2.4)	0.599 (+0.020)	31.0	30.1

Table 4: Final submission scores on test sets. In parentheses, we show the difference between our system and the best performing system by another task participant (positive indicates our system scored highest, negative indicates that the other team's score was higher). The last two columns show scores on devel_test, first for the ensemble and then for the single best component systems in the ensemble (sing.).

Our DSB-DE system had the best BLEU and chrF scores and was tied for the best BERT score (0.953) with CL_RUG; in terms of BLEU and chrF it was significantly better than one other system (alongside CL_RUG and LMU) and in terms of BERT score it was significantly better than two other systems (alongside CL_RUG). Both of our systems translating out of German had the highest BLEU and chrF scores. Our DE-HSB system was, alongside NoahNMT, significantly better than three other systems in both automatic metrics. Our DE-DSB system was, alongside LMU, significantly better than three other systems in both automatic metrics.

11.1 HSB-DE

Our Upper Sorbian-German submission is an ensemble of eight systems with 25k vocabulary, which scored 67.3 BLEU (0.836 chrF) on the test set. The first six systems in the ensemble are children and grandchildren of a CS-DE system (with both source and target BPE-dropout). The final two were multilingual systems trained on a mix of CS-DE and HSB-DE data. Their details are as follows:

1. HSB-DE data, BT2 news (M-L threshold 0), BPE-dropout (both), and label smoothing set to 0.15 (best single system)
2. HSB-DE data, BT2 news (M-L threshold 0), BPE-dropout (both), and transformer dropout at 0.20
3. Child of system 1, finetuned on only HSB-DE (with BPE-dropout, both)
4. Multilingual (mix of CS-DE language-tagged, BT1 news M-L top 1.5M tagged as BT, and HSB-DE upscaled)
5. HSB-DE data, BT1 and BT2 (M-L threshold 0, each), BPE-dropout (both)
6. Child of a backtranslation-only, BT1 and BT2 (M-L threshold 0, each), BPE-dropout (both)
7. Multilingual (not a child) mix of CS-DE and HSB-DE data, BT2 (M-L threshold 0), BPE-dropout (both)
8. Multilingual (not a child) mix of CS-DE and HSB-DE data, BT1 (M-L top 1.5M), BPE-dropout (both)

11.2 DE-HSB

Our German-Upper Sorbian system is an ensemble of seven systems with 25k vocabulary, of which the first five are children or grandchildren of a DE-CS parent system (with both source and target dropout). The final two are multilingual systems. In all cases, any backtranslation listed (BT1 or BT2) is backtranslation of the monolingual HSB data. For this language direction, our primary submission does *not* use additional postprocessing. While the additional postprocessing improved all other language directions/pairs, it decreased BLEU by 0.1 in this pair (chrF remained unchanged). The system scored 66.3 BLEU (0.837 chrF) on test.

1. Multilingual system with BT2 and BPE-dropout (both)
2. Child of system 1, finetuned on DE-HSB with BPE-dropout (both)
3. DE-HSB data, BT2, BPE-dropout (both), label smoothing set to 0.15
4. Same as system 3, with transformer dropout set to 0.15
5. DE-HSB data, BT1, BPE-dropout, both
6. Multilingual (not child), BT2, BPE-dr. (both)
7. Multilingual (not child), BT1, BPE-dr. (both)

11.3 DSB-DE

Our Lower Sorbian-German system is an ensemble of two systems with 20k vocabulary, scoring 33.5 BLEU (0.6388 chrF) on test. Both systems are

children of the 20k vocabulary equivalent of the first component of the HSB-DE ensemble: taking a CS-DE parent, then training on HSB-DE data and BT2 news (M-L threshold 0), with BPE-dropout (both), and label smoothing set to 0.15.

They train on authentic DE data that is paired with DSB backtranslations, generated by the initial round DE-DSB ensemble (described in Section 9.1).

1. Child trained on backtranslation paired with DE side of the DE-HSB training data, BPE-dropout (both), learning rate 0.0001.
2. Same as the first system, with the addition of backtranslated news 2019 (M-L threshold 0).

11.4 DE-DSB

Our German-Lower Sorbian system is an ensemble of four 20k BPE systems, and scored 29.9 BLEU (0.599 chrF) on test. All four systems are based on a default DE-CS parent with BPE-dropout (both). The first three are then finetuned with BT2 backtranslated HSB data and BPE-dropout (both). The last was finetuned with a multilingual system (again with BT2 backtranslated HSB and BPE-dropout both). We now describe how those systems were finetuned to the DE-DSB task (all used BPE-dropout both):

1. Finetuned with DSB monolingual data (initial round backtranslated).
2. Finetuned with DSB monolingual data (next round backtranslated).
3. Same as system 2 with learning rate 0.0001.
4. Same as system 3 (different parent).

12 Conclusions

As with last year's task, we found that our best systems consisted of ensembles, with more diverse ensembles performing better than less diverse ones. The very high automatic metric scores along with our experiments in backtranslation led us to examine the state of memorization of the training data, which we found to be quite high. We also found that the close relationship between Upper Sorbian and Lower Sorbian enabled us to bootstrap seemingly strong Lower Sorbian systems through iterative backtranslation. We believe that the true test of these systems will be through human evaluation, as well as an analysis of how well they perform in a real-life setting (i.e., with more out-of-domain test data), as the current set seems potentially quite constrained in domain.

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. We also thank our colleagues Michel Simard, Cyril Goutte, Marc Tessier, Darlene Stewart, Chi-kiu Lo, and Patrick Littell for discussion, comments, feedback, and technical assistance.

References

Idris Abdulmumin, Bashir Shehu Galadanci, and Aliyu Garba. 2021. Tag-less back-translation.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Sourav Dutta, Jesujoba Alabi, Saptarashmi Bandyopadhyay, Dana Ruiter, and Josef van Genabith. 2020. UdS-DFKI@WMT20: Unsupervised MT and very low resource supervised MT for German-Upper Sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1092–1098, Online. 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.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2020. Data selection for unsupervised translation of German–Upper Sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1099–1103, Online. Association for Computational Linguistics.

Alexander Fraser. 2020. Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Shuhao Gu and Yang Feng. 2020. Investigating catastrophic forgetting during continual training for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4315–4326, Barcelona, Spain (Online).

International Committee on 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 *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. Simulated multiple reference training improves low-resource machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–89, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. 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, Vancouver, Canada. Association for Computational Linguistics.

Rebecca Knowles, Samuel Larkin, Darlene Stewart, and Patrick Littell. 2020. NRC systems for low resource German-Upper Sorbian machine translation 2020: Transfer learning with lexical modifications. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1112–1122, Online. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit)*, pages 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

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.

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.

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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. 2020. The University of Helsinki and aalto university submissions to the WMT 2020 news and low-resource translation tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1129–1138, Online. 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.

Michel Simard and Atsushi Fujita. 2012. A poor man's translation memory using machine translation evaluation metrics. In *Proceedings of the 10th Biennial*

Conference of the Association for Machine Translation in the Americas. Association for Machine Translation in the Americas.

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 Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 454–459, Istanbul, Turkey. European Language Resources Association (ELRA).

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota. 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'12)*, pages 2214–2218, 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Witaj Language Centre. 2021. Sorbisches Übersetzungsprogramm „sotra" ist online.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.

NoahNMT at WMT 2021: Dual Transfer for Very Low Resource Supervised Machine Translation

Meng Zhang[1], **Minghao Wu**[2], **Pengfei Li**[1], **Liangyou Li**[1], **Qun Liu**[1]
[1] Huawei Noah's Ark Lab
[1] {zhangmeng92, lipengfei111, liliangyou, qun.liu}@huawei.com
[2] Monash University
[2] minghao.wu@monash.edu

Abstract

This paper describes the NoahNMT system submitted to the WMT 2021 shared task of Very Low Resource Supervised Machine Translation. The system is a standard Transformer model equipped with our recent technique of dual transfer. It also employs widely used techniques that are known to be helpful for neural machine translation, including iterative back-translation, selected finetuning, and ensemble. The final submission achieves the top BLEU for three translation directions.

1 Introduction

In this paper, we describe the NoahNMT system submitted to one of the WMT 2021 shared tasks. The shared task features both unsupervised machine translation and very low resource supervised machine translation. As our core technique is mainly suitable for low resource supervised machine translation, we participated in four translation directions between Chuvash-Russian (chv-ru) and Upper Sorbian-German (hsb-de).

Our core technique is called dual transfer (Zhang et al., 2021), which belongs to the family of transfer learning. It transfers from both high resource neural machine translation model and pretrained language model to improve the quality of low resource machine translation. During the preparation for the shared task, we conducted additional experiments that supplement the original paper, including the choice of parent language, the validation of Transformer big model, and the usage of dual transfer along with iterative back-translation.

In addition, we also applied proven techniques to strengthen the quality of our system, including selected finetuning and ensemble. Our final submission achieves the top BLEU on the blind test sets for three translation directions: chv→ru, ru→chv, and hsb→de.

2 Approach

In this section, we describe the techniques used in our system. Interested readers are encouraged to check out the original papers for further details.

2.1 Dual Transfer

We reproduced the illustration of dual transfer from the original paper (Zhang et al., 2021), as shown in Figure 1. This illustration shows the case of general transfer, where the high resource translation direction is A→B, and the low resource translation direction is P→Q. As discussed in the original paper, in many cases, it is possible to use shared target transfer (B=Q) or shared source transfer (A=P). Taking chv→ru as an example, we can choose en→ru as the high resource translation direction, resulting in an instance of shared target transfer. In this shared task, when training the high resource translation model, we always initialize the shared language side with the pretrained language model BERT (Devlin et al., 2019).

2.2 Iterative Back-Translation

Iterative back-translation (Hoang et al., 2018) is an extension of back-translation (Sennrich et al., 2016a). It can exploit both sides of monolingual data of a language pair, and produces translation models for both directions, which is suitable for this shared task.

The initial models for generating synthetic parallel data are produced by using dual transfer with low resource authentic parallel data. In each iteration of iterative back-translation, we use the latest model to greedily decode a disjoint subset of 4m monolingual sentences[1] to generate synthetic parallel data. Then a new model is trained on a mixture of authentic and synthetic parallel data. With the use of dual transfer, model training can start from

[1] For chv and hsb, all monolingual sentences are used in each iteration.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1009–1013

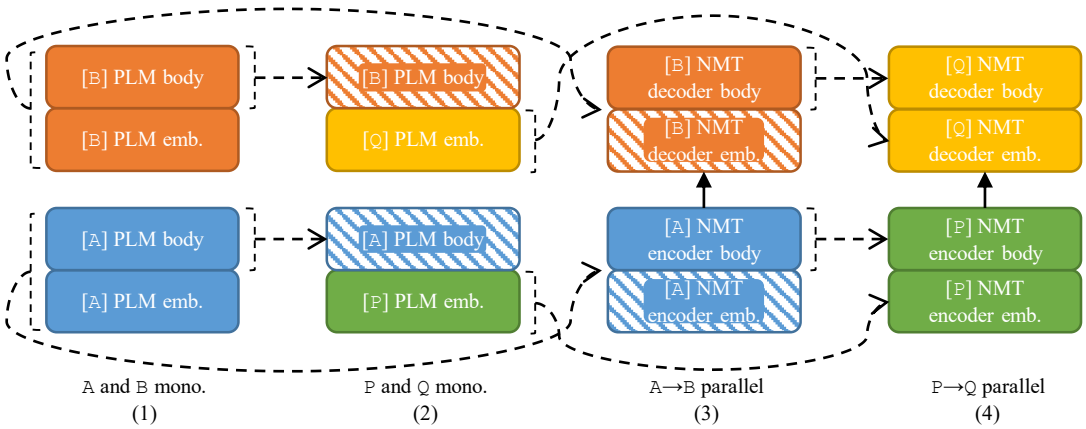


Figure 1: Dual transfer from pretrained language model and high resource A→B neural machine translation to low resource P→Q neural machine translation. Dashed lines represent initialization. Parameters in striped blocks are frozen in the corresponding step, while other parameters are trainable. Different colors represent different languages. Data used in each step is also listed.

language code	# sentence (pair)
cs-de	15m
hsb-de	0.1m
kk-ru	3.9m
en-ru	17m
chv-ru	0.7m
cs	90m
de	100m
hsb	0.8m
kk	17m
en	54m
ru	110m
chv	3m

Table 1: Training data statistics.

parent language	chv→ru BLEU
kk	18.47
en	18.61

Table 2: Test set BLEU for chv→ru, when the parent language is either kk or en (i.e. the parent translation direction is either kk→ru or en→ru). The translation model is Transformer base.

the initial parameters as shown in Step (4) of Figure 1. This has the additional benefit of reducing training time, because convergence is faster than training from random initialization.

2.3 Selected Finetuning

Selected finetuning aims to deal with the domain difference that may exist between the test set and the training set. Given the source side of the test set, we try to select similar source sentences from the training set, and then finetune the translation model on the selected subset of training sentence pairs.

We use BM25 (Robertson and Zaragoza, 2009) to calculate the similarity between two sentences for retrieval. The BM25 score between a query sentence Q and a sentence D in the corpus for retrieval $\mathcal{C}$ is given by

$$s(D,Q) = \sum_{i=1}^{L_Q} \frac{\text{IDF}(q_i) \cdot (k+1) \cdot \text{TF}(q_i, D)}{k \cdot \left(1 - b + b \cdot \frac{L_D}{L_{\text{avg}}}\right) + \text{TF}(q_i, D)},$$

where the query sentence Q is a sequence of L_Q subwords $\{q_i\}_{i=1}^{L_Q}$, $\text{IDF}(q_i)$ is the Inverse Document Frequency for q_i in the corpus $\mathcal{C}$, $\text{TF}(q_i, D)$ is the Term Frequency for q_i in the sentence D, L_D is the length of the sentence D, L_{avg} is the average length of the corpus $\mathcal{C}$, k and b are hyperparameters, which are set as 1.5 and 0.75, respectively.

Based on the BM25 score, we calculate the similarity between a source test sentence (as the query sentence) and the source sentences in the training set to obtain the top 500 sentences. After performing the selection for all the source test sentences, we merge them and remove duplicates to obtain the set for finetuning.

model	chv→ru	ru→chv	hsb→de	de→hsb
Transformer base	18.61	16.18*	55.60	55.98
Transformer big	19.24	17.12	56.10	57.12

Table 3: Test set BLEU for the four translation directions, using either Transformer base or Transformer big for dual transfer. *: The parent translation direction is ru→kk, and we did not train a Transformer base with ru→en as the parent, though the resulting ru→chv BLEU scores should be close based on the experiment in Section 4.1.

	runtime (hours)
BERT_{en}	143
BERT_{chv}	54
$\text{NMT}_{\text{en}\rightarrow\text{ru}}$	52
$\text{NMT}_{\text{chv}\rightarrow\text{ru}}$	14

Table 4: Runtime of each step in dual transfer for $\text{NMT}_{\text{chv}\rightarrow\text{ru}}$ with Transformer big.

3 Experimental Setup

3.1 Data

We collected allowed data for the involved languages and followed the same preprocessing pipeline of punctuation normalization and tokenization, using scripts from Moses[2]. The English monolingual data came from the English original side of ru-en back-translated news[3], but its automatic translation to Russian was discarded. The provided Chuvash-Russian dictionary was not used. Each language was encoded with byte pair encoding (BPE) (Sennrich et al., 2016b). The BPE codes and vocabularies were learned on each language's monolingual data, and then used to segment parallel data. We used 32k merge operations for all languages. After BPE segmentation, we discarded sentences with more than 128 subwords, and cleaned parallel data with length ratio 1.5. Training data statistics is provided in Table 1. Note that we experimented with Kazakh (kk) data (Section 4.1), but did not use it for our final submission. Evaluation on test sets is given by SacreBLEU[4] (Post, 2018), after BPE removal and detokenization.

3.2 Hyperparameters

We use Transformer (Vaswani et al., 2017) as our translation model, but with slight modifications that follow the implementation of BERT[5]. The absolute position embeddings are also learned as in BERT. The encoder and decoder embeddings are independent because each language manages its own vocabulary, but we tie the decoder input and output embeddings (Press and Wolf, 2017). We apply dropout with probability 0.1. We use LazyAdam as the optimizer. Learning rate warms up for 16,000 steps and then follows inverse square root decay. The peak learning rate is 5×10^{-4} for parent translation models, and 1×10^{-4} for child translation models. Early stopping occurs when the validation BLEU does not improve for 10 checkpoints. We set checkpoint frequency to 2,000 updates for parent translation models and 1,000 updates for child translation models. The batch size is 6,144 tokens per GPU and 8 NVIDIA V100 GPUs are used.

Hyperparameters for BERT are the same as in the original paper (Zhang et al., 2021).

For selected finetuning, we use stochastic gradient descent as the optimizer, and the learning rate is 1×10^{-5}. We finetune for 10,000 updates, and save a checkpoint every 100 updates. The checkpoint with the highest validation BLEU is kept.

4 Results

4.1 The Choice of Parent Language

In our preliminary experiments, we found it beneficial to use a closely related language as the parent language. It is clear that there are several factors that should be taken into account, such as the degree of closeness, and the amount of resource for training the parent model. For Upper Sorbian, Czech (cs) is closely related to it, and Czech-German has a good amount of parallel data, so we directly choose Czech as the parent language.

Chuvash, however, is a rather isolated language in the Turkic family. The closest language with usable data is Kazakh (kk), but the amount of parallel data for Kazakh-Russian is relatively small, and we found it to be quite noisy. Therefore, we considered

[2] https://github.com/moses-smt/mosesdecoder

[3] http://data.statmt.org/wmt20/translation-task/back-translation

[4] SacreBLEU signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.12.

[5] https://github.com/google-research/bert

iteration	chv→ru	ru→chv	hsb→de	de→hsb
0	19.24	17.12	56.10	57.12
1	19.73	17.45	57.23	56.81
2	**20.42**	17.69	57.12	56.79
3	19.85	**17.81**	**57.72**	**57.47**
4	19.57	17.78	57.40	57.33
5	19.60	17.48	57.66	57.07

Table 5: Test set BLEU for the four translation directions with iterative back-translation. Iteration 0 is the Transformer big model in Table 3. Best BLEU scores are in bold.

method	chv→ru	ru→chv
before selected finetuning	20.42	17.69
after selected finetuning	20.55	18.03

Table 6: Test set BLEU to show the effect of selected finetuning.

model	hsb→de	de→hsb
best single	57.72	57.47
ensemble	58.54	58.28

Table 7: Test set BLEU to show the effect of ensemble.

using English (en) as the parent language of Chuvash. Even though English is unrelated to Chuvash and they use different scripts, English-Russian has more parallel data that can guarantee the quality of the parent model.

We conducted an experiment with Transformer base. Results in Table 2 indicate that English can serve as an eligible parent for Chuvash. Considering that we plan to use Transformer big for which data amount is likely to play a more important role, we decided to use English as the parent language for Chuvash.

4.2 The Effect of Transformer Big

The original paper (Zhang et al., 2021) evaluated dual transfer only with Transformer base. In this shared task, we scale up to Transformer big. We also face a more realistic setting where the monolingual data for the low resource languages (chv and hsb) are quite scarce. Therefore it is worth testing the effect of scaling up. Results in Table 3 show that Transformer big brings consistent improvements. We also report the runtime of each step in dual transfer for $\text{NMT}_{\text{chv}\rightarrow\text{ru}}$ with Transformer big in Table 4 for reference, but the numbers can vary depending on implementation and data size. In the following experiments and our final submission, we use Transformer big models.

4.3 Iterative Back-Translation

We ran five iterations of iterative back-translation. Results are shown in Table 5. The best BLEU scores are attained with two or three iterations. Another observation is that iterative back-translation brings larger improvements for chv→ru and hsb→de than ru→chv and de→hsb. This is probably because the monolingual data for chv and hsb are small in quantity.

4.4 Selected Finetuning

We only use selected finetuning for the chv-ru pair because parallel data for hsb-de is scarce. In order to test the effect of selected finetuning, we start from the models of Iteration 2 in Table 5. Results in Table 6 indicate that selected finetuning gives modest improvements.

4.5 Ensemble

We validate the effectiveness of ensemble on hsb→de and de→hsb, by performing ensemble decoding from the five models from iterative back-translation. Results in Table 7 demonstrate that ensemble gives BLEU improvements of about 0.8.

4.6 Final Submission

For chv→ru and ru→chv, we perform selected finetuning starting from the best models from iterative back-translation (Iteration 2 for chv→ru, Iteration 3 for ru→chv). Note that the selected training subsets are different from those in Section 4.4 because the selection is based on the source side of the blind test sets. We finetune five times

with different random seeds for model ensemble. For hsb→de and de→hsb, we ensemble the five models from iterative back-translation.

5 Conclusion

In this paper, we describe a series of experiments that contribute to our submission to the WMT 2021 shared task of Very Low Resource Supervised Machine Translation. These experiments, as well as the good results of the final submission, show that dual transfer can work in synergy with several widely used techniques in realistic scenarios.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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, Valencia, Spain. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

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.

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*, volume 30, pages 5998–6008.

Meng Zhang, Liangyou Li, and Qun Liu. 2021. Two Parents, One Child: Dual Transfer for Low-Resource Neural Machine Translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2726–2738.

cushLEPOR: customising hLEPOR metric using Optuna for higher agreement with human judgments or pre-trained language model LaBSE

Lifeng Han[1], **Irina Sorokina**[2], **Gleb Erofeev**[2], and **Serge Gladkoff**[2]
[1] ADAPT Research Centre, DCU, Ireland
[2] Logrus Global, Translation & Localization
lifeng.han@adaptcentre.ie
gleberof, irina.sorokina, serge.gladkoff@logrusglobal.com

Abstract

Human evaluation has always been expensive while researchers struggle to trust the automatic metrics. To address this, we propose to customise traditional metrics by taking advantages of the pre-trained language models (PLMs) and the limited available human labelled scores. We first re-introduce the hLEPOR metric factors, followed by the Python version we developed (ported) which achieved the automatic tuning of the weighting parameters in hLEPOR metric. Then we present the customised hLEPOR (cushLEPOR) which uses Optuna hyper-parameter optimisation framework to fine-tune hLEPOR weighting parameters towards better agreement to pre-trained language models (using LaBSE) regarding the exact MT language pairs that cushLEPOR is deployed to. We also optimise cushLEPOR towards professional human evaluation data based on MQM and pSQM framework on English-German and Chinese-English language pairs. The experimental investigations show cushLEPOR boosts hLEPOR performances towards better agreements to PLMs like LaBSE with much lower cost, and better agreements to human evaluations including MQM and pSQM scores, and yields much better performances than BLEU (data available at https://github.com/poethan/cushLEPOR). Official results show that our submissions win three language pairs including **English-German** and **Chinese-English** on *News* domain via cushLEPOR(LM) and **English-Russian** on *TED* domain via hLEPOR.

1 Introduction

Machine Translation (MT) is a rapidly developing research field that plays an important role in NLP area. MT started from 1950s as one of the earliest artificial intelligence (AI) research topics and gained a large improvement in the output quality in large resourced language pairs after the introduction of Neural MT (NMT) in recent years (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Bahdanau et al., 2014). However, the challenge still remains in achieving human parity of MT output (Han et al., 2021a). Thus MT evaluation (MTE) continues to play an important role in aiding MT development from the aspects of timely and high quality evaluations, as well as reflecting the translation errors that MT systems can take advantages of for further improvement (Han et al., 2021b). On one hand, human evaluations have long been criticised as expensive and unrepeatable. Furthermore, the inter- and intra-agreement levels from Human raters may struggle to achieve a consistent and reliable score, unless done in rigour with highly trained and skilled evaluators (Alekseeva et al., 2021). On the other hand, even though researchers have claimed that the automatic evaluation metrics have reached much better performances in the category of system level evaluations of MT outputs, with high correlation to human judgements, the segment level performance is still a large gap from human experts' expectation (Freitag et al., 2021; Barrault et al., 2019, 2020; Han et al., 2013a; Macháček and Bojar, 2013).

In the meantime, many pre-trained language models have been proposed and developed in very recent years and showing big advantages in different NLP tasks, for instance, BERT (Devlin et al., 2019) and its further developed variants (Feng et al., 2020). In this work, we take the advantages of both high performing automatic metric and pre-trained language model, aiming at one step further towards higher quality performing automatic MT evaluation metric from both system level and segment level perspectives.

Among the evaluation metrics developed recent years, hLEPOR (Han et al., 2013b,a) is an augmented metric that include many evaluation factors with tunable weights assigned including precision, recall, word order (via position difference factor),

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1014–1023
November 10–11, 2021.

and sentence length. It has also been applied by researchers from different NLP fields including natural language generation (NLG) (Novikova et al., 2017; Gehrmann et al., 2021), natural language understanding (NLU) (Ruder et al., 2021), automatic text summarization (ATS) (Bhandari et al., 2020), and searching (Liu et al., 2021), in addition to MT evaluation (Marzouk, 2021).

However, original hLEPOR has disadvantage of the manual tuning of its parameter weights which take a lot of human efforts. We choose hLEPOR (Han et al., 2013b,a) as our baseline model, and use the very recent language model LaBSE (Feng et al., 2020) to achieve automatic tuning of its parameters thus aiming at reducing the evaluation cost and further boosting the performance. This system description paper is based on our earlier work, especially the training models (Erofeev et al., 2021).

The rest of the paper is organised as below: Section 2 revisits hLEPOR metric, its factors, advantages and disadvantages, Section 3 introduces our Python ported version of hLEPOR and the further customised hLEPOR (cushLEPOR) using language models, Section 4 presents our experimental development and evaluation that we carried out on cushLEPOR metric using WMT historical data, Section 5 reserves space for our submission to this year WMT21 metrics task, and Section 6 finishes this paper with discussions of our findings and possible future work.

2 Revisiting hLEPOR

hLEPOR is a further developed variant of LEPOR (Han et al., 2012) metric which was firstly proposed in 2013 including all evaluation factors from LEPOR but using harmonic mean for grouping factors to produce final calculation score (Han et al., 2013b). Its submission to WMT2013 metrics task achieved system level highest average correlating scores to human judgement on English-to-other (French, Spanish, Russian, German, Czech) language pairs by Pearson correlation coefficient (0.854) (Han, 2014; Macháček and Bojar, 2013). Other MT researchers also analysed LEPOR metric variant as one of the best performing segment level metric that was not significantly outperformed by other metrics using WMT shared task data (Graham et al., 2015). hLEPOR is calculated by:

$$\textit{h}\text{LEPOR} = Harmonic(w_{LP}LP, w_{NPosPenal}NPosPenal, w_{HPR}HPR)$$

where ***LP*** is a sentence length penalty factor which was extended from brevity penalty utilised in BLEU metric, ***NPosPenal*** is for n-gram position difference penalty which captures the word order information, as bellow, where $MatchN_{hyp}$ and $MatchN_{ref}$ indicate the position number of matched words in hypothesis and reference sentences:

$$\text{LP} = \begin{cases} e^{1-\frac{Length_{ref}}{Length_{hyp}}} \; if \; Length_{hyp} < Length_{ref} \\ 1 \; if \; Length_{hyp} = Length_{ref} \\ e^{1-\frac{Length_{hyp}}{Length_{ref}}} \; if \; Length_{hyp} > Length_{ref} \end{cases}$$

$$NPosPenal = e^{-NPD}$$

$$NPD = \frac{1}{Length_{hyp}} \sum_{i=1}^{Length_{hyp}} |PD_i|$$

$$|PD_i| = |MatchN_{hyp} - MatchN_{ref}|$$

The factor ***HPR*** is the harmonic mean of Precision and Recall values.

$$HPR = \frac{(\alpha+\beta)PrecisionxRecall}{\alpha Precision + \beta Recall}$$

$$Precision = \frac{Aligned_{num}}{Length_{hypothesis}}$$

$$Recall = \frac{Aligned_{num}}{Length_{reference}}$$

We refer the work (Han, 2014; Han et al., 2013a, 2012) for detailed factor calculation with examples there.

The basic version of hLEPOR carries out similarity calculation between MT system outputs and reference translations, in the same language setting, based on the *word surface level* tokens. The hybrid hLEPOR metric also carries out similarity calculation based on POS sequences from system-output and reference text. To do this, POS tagging is needed as the first step, then hLEPOR(POS) calculation uses the same algorithms used for the word level similarity score hLEPOR(word). Finally, hybrid hLEPOR is a combination of both word level and POS level score. In this system submission

work, with the time limitations, to make an easier to use customised hLEPOR, we take the basic version of hLEPOR, i.e. the word level similarity calculation and leave the hybrid hLEPOR into the future work.

The weighting parameters for the three main factors in original hLEPOR metric, i.e the (w_{LP}, $w_{NPosPenal}$, w_{HPR}) set, in addition to the other parameters inside each factor, were tuned by manual work based on development data (see *Appendix* for detailed parameter value sets on each language pair for word-surface level evaluation, en $\Leftrightarrow$ cs/fr/de/es/ru). This is very time consuming, tedious, and costly. In this work, we will introduce an automated tuning model for hLEPOR to customise it regarding deployed language pairs, which we name as cushLEPOR.

3 Proposed Model

3.1 Python port of hLEPOR

Original hLEPOR was published as Perl code [1], in a non-portable format, which is not very suitable for modern AI/NLP applications, since they are using almost exclusively Python. Python is a programming language of choice for AI and machine learning (ML) tasks, thanks to its amazing ecosystem of open source or simply free libraries available to researchers and developers. However, hLEPOR was not available in NLTK (Bird et al., 2009) or any other public Python libraries. We therefore took original published Perl code and ported it to Python, carefully comparing the logic of original paper and the Perl implementation. During this work we run both Perl code to reproduce the results of original code, and the new Python implementation. This work helped us to spot and fix at least three minor errors which did not significantly affected the score, but nevertheless we fixed the bugs of the Perl code.

While doing the porting we did also notice that hLEPOR parameter values were taken empirically and never explained in detail except for the suggested parameter setting table in the paper (Han et al., 2013b,a) for eight language pairs that were tested for the WMT2013 shared task, including EN-CZ/DE/FR/ES and the opposite direction. They were:

- **alpha**: the tunable weight for recall
- **beta**: the tunable weight for precision
- **n**: words count before and after matched word in npd calculation
- **weight_elp**: tunable weight of enhanced length penalty
- **weight_pos**: tunable weight of n-gram position difference penalty
- **weight_pr**: tunable weight of harmonic mean of precision and recall

The parameter values for hLEPOR as published in the publicly available Perl code were mannually tuned for *English-to-Czech/Russian (EN=>CS/RU) language pair* setting (Han et al., 2013b,a) as below:

- **alpha** = 9.0,
- **beta** = 1.0,
- **n** = 2,
- **weight_elp** = 2.0,
- **weight_pos** = 1.0,
- **weight_pr** = 7.0.

We refer to our **Appendix** for the manually tuned parameters for other language pairs available in the paper by (Han et al., 2013a,b) including English=>French/German (EN=>FR/DE) and Czech/Spanish/French/German=>English (CS/ES/FR/DE=>EN) and Russian=>English (RU=>EN) which was set up using CS=>EN without extra manual tuning. We came to the conclusion that we need to check whether these parameters are optimal, and find out whether better set of values exist to improve agreement with human judgement.

Because the different characteristics of each language, and language families, the evaluation of MT outputs would emphasis on different factors. For instance, word order factor reflected by n-gram position different penalty in hLEPOR (NPosPenal), can be with higher or lower weight for strict order languages and loose/flexible word order languages. Thus, we assumed that hLEPOR optimisation towards different languages will generate corresponding different set of parameter values. We call this step language-specific optimisation, and it will save much cost and time to achieve an automatic tuning process. The Python ported hLEPOR is available at Pypi https://pypi.org/project/hLepor/.

[1] https://github.com/poethan/LEPOR

3.2 Customised hLEPOR: cushLEPOR

With the recent development of pre-trained neural language models and their effective applications in different NLP tasks, including question answering, language inference and MT, it becomes a natural question that why do not we apply them in MT evaluation as well.

Very recent work from Google team verified that MQM (Multi-dimension quality metric) (Lommel et al., 2014) and SQM (Scalar Quality Metrics) (Freitag et al., 2021) have good agreement with each other when they were carried out both by professional translators. However, this does not correlate to Mechanical Turk based crowd-sourced human evaluation that was carried out by general researchers or untrained online workers with low professional linguistic skills. It also reflected that crowd-sourced evaluation tends to favour very literal translations instead of better translations with more diverse meaning equivalent lexical choices.

To customise hLEPOR (cushLEPOR) towards optimised parameter setting for deployed language pairs, we choose Optuna open source hyper-parameter optimisation framework (Akiba et al., 2019) to automate hyper-parameter search for best agreement between cushLEPOR and human experts evaluation wherever such data-set is available.

SQM (Freitag et al., 2021) borrows WMT shared task settings to collect segment-level scalar rating, but set the score scale from 0 to 6 instead of 0 to 100. Professional translator labelled scores using SQM is named as pSQM.

We aim at optimising cushLEPOR parameters to obtain best agreement with pSQM scores. However, in practical situation, human evaluations are not often feasible to obtain due to the constrains from both time and financial aspects.

We therefore propose to carry out an alternative optimisation model, i.e. customising cushLEPOR parameters towards pre-trained large scale language models, e.g. LaBSE (Language Agnostic BERT Sentence Embedding) model similarity score.

LaBSE model is built on BERT (Bidirectional Encoder Representations from Transformers) architecture and trained on monolingual (for dictionaries) and bilingual training data. LaBSE training data is filtered and processed. The resulting sentence embeddings achieve excellent performance on measures of sentence embedding quality such as the semantic textual similarity (STS) benchmark and sentence embedding based transfer learning (Feng et al., 2020).

LaBSE linguistic similarity score finds matching translations very well. The disadvantages, however, are high demand for computational resources (with GPUs), intensive application coding with requirement for ML skills, and slow performance.

The design of using optimised hLEPOR (cushLEPOR) in lieu of LaBSE similarity aims at developing a simple, high-performing, easy to run and not computationally demanding script to achieve results similar to high-end LaBSE similarity score, and hopefully towards human judgement. The cushLEPOR parameters can be optimised for agreement with any type of scores, such as pSQM, MQM, and LaBSE, etc.

Regarding the optimisation stage using Optuna, the task is to find the extremum values of continuous (not discrete) surface in a 6-dimensional space of six cushLEPOR parameters. The values of parameter set change continuously which means there's an infinite number of parameter values; however it is not a differentiable situation mathematically, and there are gaps. Generally we cannot presume that it is a smooth surface. Before Optuna, computational tools used to deploy a discrete mesh in such cases by using discretization method, which was less computationally intense than full scale continuous search on all possible values of parameter sets.

Optuna framework is currently one of the best Tree-structured Parzen Estimator (TPE) model implementations, which kind of estimators converges to optimal solution in 200-300 epochs, and the method can work with continuous (real) parameters (Bergstra et al., 2011).

4 Experimental Evaluations

The training and development data we used regarding MQM scores and pSQM labels is from the recent work by Google Research team on investigating into human evaluations based on WMT2020 shared task (Freitag et al., 2021) (data available at `https://github.com/google/wmt-mqm-human-evaluation`).

We first focus on English-to-German language (EN-DE) pair, which includes MQM and pSQM labels, acquired from 10 submission of WMT 2020, then take ZH-EN data-set. We refer to the paper (Freitag et al., 2021) for detailed MT system names and offering institutions.

Firstly, a multi-parameter optimisation against LaBSE for EN-DE language pair gave the following values for cushLEPOR parameters:

- **alpha** = 2.97,
- **beta** = 1.97,
- **n** = 4,
- **weight_elp** = 1.0,
- **weight_pos** = 14.97,
- **weight_pr** = 2.2.

This set of values reflected very different weighting systems in comparison to the original hLEPOR metric. For instance, 1) cushLEPOR assigned recall and precision much closer weight (2.97 vs 1.97) in comparison to hLEPOR (9.0 vs 1.0), 2) cushLEPOR chose 4-gram in chunk matching instead of bi-gram used in hLEPOR, 3) cushLEPOR assigned NPosPenal (n-gram position difference penalty) factor a very heavy weight against other two factors LP (length penalty) and HPR (harmonic mean of precision and recall) by (**14.97** vs 1.0 and 2.2) in comparison to hLEPOR which emphasised the weight on HPR (1.0 vs 2.0 and **7.0**). From these points of view, cushLEPOR trained on EN-DE language pair indicates the importance of the larger window context consideration during word matching, as well as the word order information reflected by n-gram (n value) and novel factor NPosPenal introduced by hLEPOR respectively.

This also reflected that LaBSE similarity is indeed a feasible goal for cushLEPOR optimisation. The correlations of hLEPOR and cushLEPOR to LaBSE are shown in Fig. 1 and 2.

However, we found out that we were not able to decrease much on RMSE (Root Mean Square Error) score for cushLEPOR towards pSQM , in comparison to original hLEPOR, (0.28 vs 0.29)which does indicate that original hLEPOR empirically shows very good fit for pSQM type human evaluation, using the suggested parameter settings for EN-DE (Han et al., 2013a,b) as bellow.

- **alpha** = 9.0,
- **beta** = 1.0,
- **n** = 2,
- **weight_elp** = 3.0,

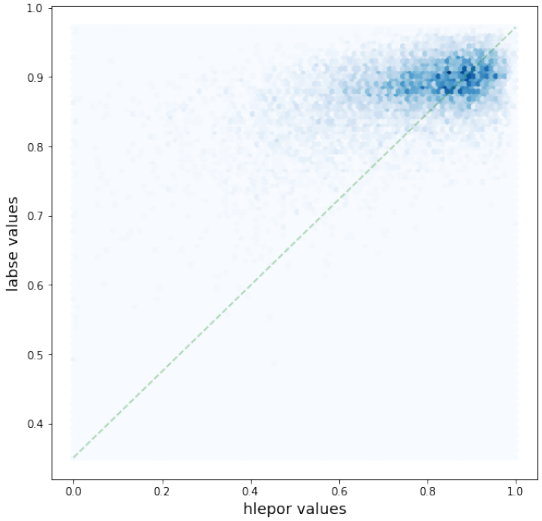


Figure 1: Agreement with LaBSE: hLEPOR

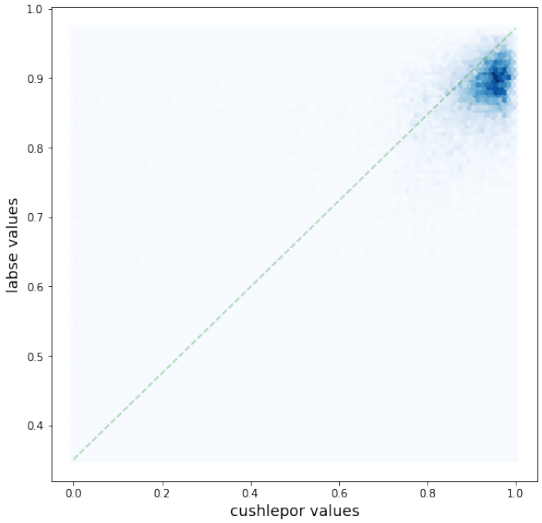


Figure 2: Agreement with LaBSE: cushLEPOR

- **weight_pos** = 7.0,
- **weight_pr** = 1.0.

The RMSE value between pSQM and hLEPOR, vs pSQM and cushLEPOR is shown in Fig. 3. However, it indeed shows much better performance then BLEU metric, as in Fig. 4 (0.28 vs 0.46).

Optuna did optimise cushLEPOR against LaBSE very well, halving the RMSE distance between LaBSE and cushLEPOR as compared to original hLEPOR, shown in Fig. 5.

The performances of tuning on LaBSE and pSQM are shown in Fig. 6 and 7 respectively. The horizontal axis is the score value (0, 1) and the vertical axis is the sentence number that falls into the corresponding score intervals.

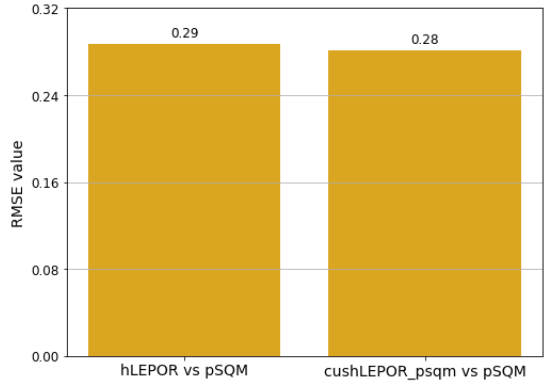


Figure 3: RMSE: hLEPOR vs cushLEPOR to pSQM (lower score is better)

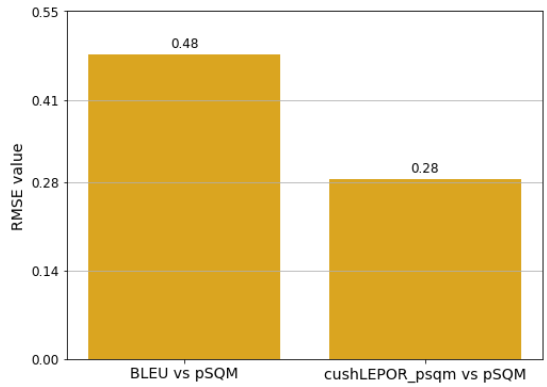


Figure 4: RMSE: BLEU vs cushLEPOR to pSQM (lower score is better)

Figure 5: RMSE: hLEPOR vs cushLEPOR to LaBSE (lower score is better)

From the score distribution visualisation, it reflects the tuning on pSQM has a larger covered error types while LaBSE is less sensitive to some errors that human experts would spot out. As shown on these charts, pSQM human rating shows much wider "tail" of "low score ratings", while LaBSE rating is much more focused. The reason is that LaBSE similarity model underestimates the severity of errors and error types, while humans analyse the meaning and assign proper error penalties in more diverse setting. As an example, the sentence "The comet did not struck the Earth this time." and "The comet did struck the Earth this time." has very close lexical similarity, but the meaning is very different, in this case “opposite”. LaBSE similarity score would not assign significant penalty to such difference, while human will treat it as a major error. This difference plays a crucial role for reliable translation quality evaluation.

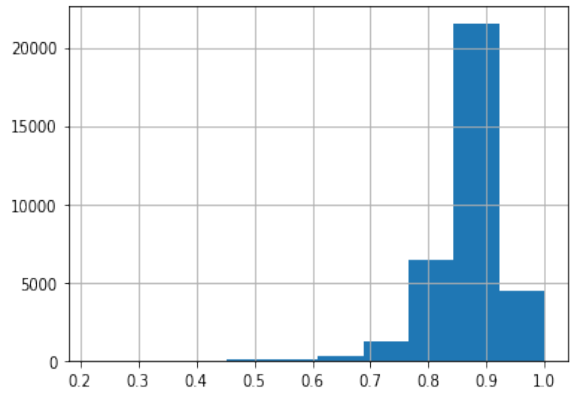


Figure 6: Score Distribution: tune on LaBSE

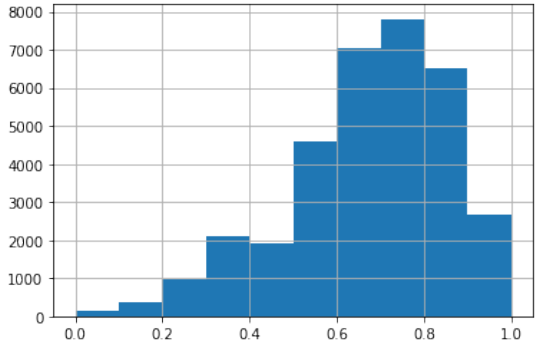


Figure 7: Score Distribution: tune on pSQM

5 Submission to WMT21

For WMT2021 Metrics Task, we submitted our cushLEPOR system scores for zh=>en and en=>de language pairs, both segment-level and system-level evaluation. The training and development set we used are exact the ones from last section (Section 4). We can not tune our cushLEPOR model parameters on en=>ru language pair from the WMT21 official data, because the human labelled MQM and pSQM scores as validation data that cushLEPOR requires do not exist from last year WMT20 set. Instead, we submitted hLEPOR metric for EN=>RU using the parameter settings in hLEPOR as men-

tioned in the last section. We carried out evaluation on all four official data-sets: **newstest2021** (traditional task), **florestest2021** (sentences translated as part of the WMT News translation task), **tedtalks** (additional sets of sentences translated by WMT21 translation systems in the TED talks domain), and **challengeset** (synthetic outputs generated specifically to challenge automatic metrics).

5.1 Submitted Parameter Setting

The optimised parameter values set for our zh=>en submission to WMT21 is displayed below.

For cushLEPOR(LM) using LaBSE training:

- **alpha** = 2.85,
- **beta** = 4.73,
- **n** = 1,
- **weight_elp** = 1.01,
- **weight_pos** = 11.13,
- **weight_pr** = 4.62

For cushLEPOR(pSQM) using professional translator labelled SQM training:

- **alpha** = 9.09,
- **beta** = 3.55,
- **n** = 3,
- **weight_elp** = 1.01,
- **weight_pos** = 14.98,
- **weight_pr** = 1.57

The optimised parameter values set for our en=>de submission to WMT21 is displayed below:

For cushLEPOR(LM) using LaBSE training:

- **alpha** = 2.95,
- **beta** = 2.68,
- **n** = 2,
- **weight_elp** = 1.0,
- **weight_pos** = 11.79,
- **weight_pr** = 1.87

For cushLEPOR(pSQM) using professional translator labelled SQM training:

- **alpha** = 1.13,
- **beta** = 1.71,
- **n** = 2,
- **weight_elp** = 1.06,
- **weight_pos** = 11.90,
- **weight_pr** = 1.01

5.2 Official Results from Metrics Task

The official results from WMT2021 Metrics task show that cushLEPOR(LM) ranks in the **first cluster** in performance on **News test** data with single reference evaluated on overall English-to-German, Chinese-to-English and English-to-Russian where professional human evaluation data is available (*Ref.* Table 8 "Metric rankings based on pairwise accuracy" in Findings paper (Freitag et al., 2021)). Furthermore, in the language specific ranking, cushLEPOR(LM) also wins **English-to-German** and **Chinese-to-English** language pairs, including TED data condition. Our hLEPOR baseline metric wins **English-to-Russian** TED domain language specific ranking (Ref. Table 12 "Summary of language-specific results" in the official findings paper (Freitag et al., 2021)). The official result on "System-level Pearson correlations for **English-to-German**" (Table 23 of findings) shows that cushLEPOR(LM) achieves score 0.938 in News domain, ranking *number 1 in Cluster 1 metrics*, out of overall 29 metric submissions.

6 Discussions and Future Work

In this work, we described cushLEPOR, a customised hLEPOR metric which can be automatically trained and optimised using both human labelled MQM scores, as well as large scale pre-trained language model (LM) LaBSE towards better agreement to human experts level judgements and distilled LM performance respectively, and reducing cost at the meantime, e.g. the manual tuning from hLEPOR and high computational demand from LMs.

We also optimised cushLEPOR towards human translators' evaluation scores, i.e. pSQM, which showed much improved performance than BLEU and original hLEPOR (with default parameters). Our research is in line with the MT evaluation guideline suggestions from the very recent work (Marie et al., 2021) that better evaluation metrics in

correlation to human judgement shall be tested and deployed. Or human judgements shall be carried out directly wherever possible.

We have some findings during the experimental investigation: 1) cushLEPOR trained on LaBSE can replace LaBSE to carry out similarity calculation task in MT evaluation, which is much more light weighted and low cost from computational power and complexity point of view. 2) we can choose alternative pre-trained language models (LMs) in the future to boost performance. 3) this cushLEPOR optimisation framework proves to be functional, offering high performance towards pre-trained LMs, much improved agreement of cushLEPOR to LaBSE scores in comparison to hLEPOR (as in Figure 1 and 2). 4) optimised cushLEPOR achieves better agreement towards professional translator's evaluation (pSQM).

Optuna, the hyper parameter optimisation toolkit we used, can generate different set of cushLEPOR parameter values in different runs, which could be an consistency issue. However, we believe it optimises the performance of cushLEPOR towards the highest agreement to the reference scoring (pre-trained LMs or human evaluations), but not to ensure the same set of parameter values to be generated, so this will not be a problem. We will carry out further analysis on this aspect in the future work.

The hybrid version of hLEPOR (Han et al., 2013b) use POS features to function as pseudo synonyms to capture alternative correct translations. However it relays on POS taggers for target language, which does not exist for newly proposed languages, and its tagging accuracy may be low, and it costs extra processing steps. In the future work, we plan to carry out integrated model which combine the POS tagging as a command function in data pre-processing for hybrid cushLEPOR.

Overall, cushLEPOR achieved the first cluster performances in *News* Domain data on *Chinese-English* and *English-German* in WMT2021 Metrics task, while hLEPOR wins *TED* domain data on *English-Russian* (Freitag et al., 2021). In the future work, we plan to carry out optimisation of cushLEPOR on more language pairs as well as more domains. We will keep our updated parameter set for extended languages and domains available on our cushLEPOR open-sourced platform.

Acknowledgements

The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. We thank the following open-source teams: Google Research for sharing their human evaluation data on MQM and pSQM scores, Optuna open-source hyper-parameter optimisation framework, and LaBSE pre-trained LMs.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902.

Alexandra Alekseeva, Serge Gladkoff, Irina Sorokina, and Lifeng Han. 2021. Monte carlo modelling of confidence intervals in translation quality evaluation (tqe) and post-editing dstance (ped) measurement. In *Metrics 2021: Workshop on Informetric and Scientometric Research (SIG-MET), 23-24 Oct 2021*. Association for Information Science and Technology.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2546–2554, Red Hook, NY, USA. Curran Associates Inc.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*. O'Reilly Media Inc.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Gleb Erofeev, Irina Sorokina, Lifeng Han, and Serge Gladkoff. 2021. cushLEPOR uses LABSE distilled knowledge to improve correlation with human translations. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 421–439, Virtual. Association for Machine Translation in the Americas.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *arXiv e-prints*, page arXiv:2104.14478.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Six Conference on Machine Translation*. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics. *arXiv e-prints*, page arXiv:2102.01672.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1183–1191.

Aaron L. F. Han, Derek F. Wong, and Lidia S. Chao. 2012. LEPOR: A robust evaluation metric for machine translation with augmented factors. In *Proceedings of COLING 2012: Posters*, pages 441–450, Mumbai, India. The COLING 2012 Organizing Committee.

Aaron Li-Feng Han, Derek F. Wong, Lidia S. Chao, Yi Lu, Liangye He, Yiming Wang, and Jiaji Zhou. 2013a. A description of tunable machine translation evaluation systems in WMT13 metrics task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 414–421, Sofia, Bulgaria. Association for Computational Linguistics.

Lifeng Han. 2014. *LEPOR: An Augmented Machine Translation Evaluation Metric*. University of Macau, MSc. Thesis.

Lifeng Han, Gareth Jones, Alan Smeaton, and Paolo Bolzoni. 2021a. Chinese character decomposition for neural MT with multi-word expressions. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 336–344, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Lifeng Han, Alan Smeaton, and Gareth Jones. 2021b. Translation quality assessment: A brief survey on manual and automatic methods. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age*, pages 15–33, online. Association for Computational Linguistics.

Lifeng Han, Derek F. Wong, Lidia S. Chao, Liangye He, Yi Lu, Junwen Xing, and Xiaodong Zeng. 2013b. Language-independent model for machine translation evaluation with reinforced factors. In *Machine Translation Summit XIV*, pages 215–222. International Association for Machine Translation.

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

Zeyang Liu, Ke Zhou, and Max L. Wilson. 2021. Meta-evaluation of Conversational Search Evaluation Metrics. *arXiv e-prints*, page arXiv:2104.13453.

Arle 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*, 0:455–463.

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.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online. Association for Computational Linguistics.

Shaimaa Marzouk. 2021. An in-depth analysis of the individual impact of controlled language rules on machine translation output: a mixed-methods approach. *Machine Translation*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards More Challenging and Nuanced Multilingual Evaluation. *arXiv e-prints*, page arXiv:2104.07412.

Appendices

Appendix A: hLEPOR parameters

The word level hLEPOR default parameters manually tuned for WMT2013 MT evaluation task across language pairs (Han et al., 2013a,b) are displayed as below. Both Python (`https://pypi.org/project/hLepor/`) and Perl (`https://github.com/lHan87/aaron-project-hlepor`) version codes can be applied to:

On English-to-Czech/Russian (EN=>CS/RU):

- **alpha** = 9.0,
- **beta** = 1.0,
- **n** = 2,
- **weight_elp** = 2.0,
- **weight_pos** = 1.0,
- **weight_pr** = 7.0.

On English-to-German (EN=>DE):

- **alpha** = 9.0,
- **beta** = 1.0,
- **n** = 2,
- **weight_elp** = 3.0,
- **weight_pos** = 7.0,
- **weight_pr** = 1.0.

On Czech / Spanish / Russian to English (CS/ES/RU =>EN):

- **alpha** = 1.0
- **beta** = 9.0
- **n** = 2
- **weight_elp** = 2.0
- **weight_pos** = 1.0
- **weight_pr** = 7.0

On German/French-to-English (DE/FR=>EN) and English-to-Spanish/French (EN=>ES/FR):

- **alpha** = 9.0
- **beta** = 1.0
- **n** = 2
- **weight_elp** = 2.0
- **weight_pos** = 1.0
- **weight_pr** = 3.0

MTEQA at WMT21 Metrics Shared Task

Mateusz Krubiński[1], Erfan Ghadery[2], Marie-Francine Moens[2], and Pavel Pecina[1]

[1]Charles University, Faculty of Mathematics and Physics
{krubinski,pecina}@ufal.mff.cuni.cz
[2]KU Leuven, Department of Computer Science
{erfan.ghadery,sien.moens}@kuleuven.be

Abstract

In this paper, we describe our submission to the WMT 2021 Metrics Shared Task. We use the automatically-generated questions and answers to evaluate the quality of Machine Translation (MT) systems. Our submission builds upon the recently proposed MTEQA framework. Experiments on WMT20 evaluation datasets show that at the system-level the MTEQA metric achieves performance comparable with other state-of-the-art solutions, while considering only a certain amount of information from the whole translation.

1 Introduction

The goal of automatic Machine Translation (MT) evaluation is to automatically evaluate the output quality produced by MT systems. Metrics used for this task assign a score by comparing the MT output to either a reference translation or to the source sentence. The main indicator that is used to assess the performance of a specific metric is the correlation with human judgement computed for outputs from several systems. It was recently shown that metrics based on contextualized embeddings, such as YISI (Lo, 2019) or ESIM (Mathur et al., 2019), are able to achieve better performance than the most common BLEU (Papineni et al., 2002).

In this paper, we describe application of the recently proposed metric – MTEQA (Krubiński et al., 2021) for the task of evaluating the quality of MT outputs in the context of the WMT21 Metric task.

The MTEQA[1] framework is inspired by previous works on evaluating abstractive summaries. It builds upon the fact that state-of-the-art (neural) MT systems tend to produce a fluent output but sometimes fail in adequacy of the translation. It leverages the recent progress in Question Generation (QG) and Question Answering (QA) to formulate and answer questions based on the MT output.

[1]https://github.com/ufal/MTEQA

2 Related Work

2.1 MT Evaluation

Metrics that are most widely used for automatic evaluation of MT outputs produce a score by comparing surface-level forms of hypothesis and reference translation. The most common one, BLEU, is a modified version of n-gram precision calculated by averaging over different values of n with penalization for too short translations (brevity penalty). The recently proposed CHRF (Popović, 2015) considers the character-level n-grams, making it possible to reward partially matched tokens. Recently, various works (e.g., Lo, 2019; Mathur et al., 2019; Bawden et al., 2020) explored the usage of contextualized word- or sentence-level embeddings to compare the numerical representations of reference and hypothesis. Such metrics enable explicit regression towards the desired human-produced labels.

2.2 Question-based Evaluation

Previous works examined the usage of reading comprehension tests to measure the quality and "usefulness" of MT systems (Tomita et al., 1993; Fuji et al., 2001; Castilho and Guerberof Arenas, 2018). Berka et al. (2011) were the first to use the *yes/no* type of questions for manual evaluation of MT systems, examining the English-to-Czech direction. Scarton and Specia (2016) approached the problem of document-level Quality Estimation (QE) by extending the CREG corpus (Ott et al., 2012) of German documents designed for reading comprehension exercises.

More work on the questions-based evaluation was done in the context of text summarization. Eyal et al. (2019) proposed the APES metric for the task of evaluating abstractive text summarization. They used the reference summary to produce fill-in-the-blank type of questions, by finding all possible entities using a NER system. The APES score for a given summarization model is the percentage

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1024–1029
November 10–11, 2021.

Reference	Extracted Answers	Generated Questions	MT output	Test Answers
The 56-year-old Macura studied at Prague University of Economics.	56	How old is Macura?	Fifty-six-year-old Macura graduated from the University of Economics in Prague.	Fifty-six
	Prague University of Economics	Where did Macura study?		University of Economics
Um 19 Uhr haben wir das Auto gepackt und sind an Bord der Fähre nach Portsmouth gegangen.	Portsmouth	Wo sind wir hin, nachdem wir das Auto gepackt haben?	Gegen 19 Uhr haben wir das Auto gepackt und die Fähre nach Portsmouth bestiegen.	Portsmouth
	19 Uhr	An welchem Datum haben wir das Auto gepackt?		19 Uhr

Figure 1: Example of the Extracted Answers, Generated Questions and corresponding Test Answers from a *newstest2021* reference file.

of questions that were answered correctly (using a Question Answering system), averaged over the whole test-set. Scialom et al. (2019) extended their work into unsupervised settings by generating questions from the source document. The FEQA (Durmus et al., 2020) and QAGS (Wang et al., 2020) metrics further extend the idea by automatically generating human-readable questions.

2.3 MTEQA

MTEQA is the first MT metric based on the principles of question answering.

The automatically generated pairs of a question and its (gold-standard) answer from the reference translation are used by a question answering system to provide a new (test) answer given the question and the MT output (translation) used as the context.

The generated (test) answer is then compared to the gold-standard answer, using the string-comparison metric. The final score for a given MT output is the average taken over all of the question/answer pairs generated for a corresponding reference.

3 Experiments

Our implementation of the MTEQA metric is based on the state-of-the-art system capable of solving the initial three tasks of the procedure: answer extraction, question generation, question answering. It is the T5 model (Raffel et al., 2020) fine-tuned on the SQuADv1 dataset (Rajpurkar et al., 2016) by Patil (2020) and available from GitHub[2]. The limitation of the T5 model is that it was trained on English data and most importantly tuned on the SQuADv1 dataset which is in English. Thus, this model only allows evaluation of MT systems translating from any language to English.

To overcome that, we used the multilingual mT5 model (Xue et al., 2021) and fine-tuned it on machine translation of SQuADv1 dataset. We exploited the existing translations into German (Lewis et al., 2020) and into Czech (Macková and Straka, 2020) which allows score translations into German (xx-de) and into Czech (xx-cs) directions. Due to time constraints we were not able to train QA and QG systems in other languages.

Figure 1 presents examples of extracted answers and generated questions.

3.1 Baseline

The baseline implementation is based on the T5 model tuned on the SQuADv1 dataset and used to generate: 1) the gold-standard answers from the reference translations, 2) a question for each gold-standard answer, 3) a test answer for each question and MT output (context) pair. The test answers are compared by the word-level F1 score commonly used for QA evaluation (Rajpurkar et al., 2016; Trischler et al., 2017; Chen et al., 2019; Durmus et al., 2020).

For each of the MT systems participating in WMT20 News translation task (Barrault et al., 2020), we compute both segment-level scores and a single system-level score, an average of segment-level scores. We report the system-level Pearson correlation with a DA human assessment using the *newstest2020* references We report correlation for a English → German, English → Czech and a few to-English directions (see Table 1, row MTEQA F1). We also include an average over all of the

[2] https://github.com/patil-suraj/question_generation

	cs-en 12	de-en 12	zh-en 16	avg	en-de 14	en-cs 12
MTEQA F1	0.782*	0.997*	0.952*	0.893*	0.946*	0.845*
MTEQA F1 KEYPHRASE	0.851*	**0.998***	0.944*	0.896*	0.941*	0.877*
MTEQA CHRF KEYPHRASE	**0.890***	**0.998***	0.951*	**0.905***	0.952*	0.859*
SENTBLEU	0.844	0.978	0.948	0.859	0.934	0.840
BLEU	0.851	0.985	0.956	0.854	0.928	0.825
PRISM	0.818	**0.998**	0.957	0.880	**0.958**	**0.949**
YISI-2	0.764	0.988	**0.964**	0.821	0.899	0.714

Table 1: System-level Pearson correlation for selected metrics used for measuring MT quality with DA human assessment over MT systems using the *newstest2020* references. Average (avg) is computed over all to-English directions available. Number below the language pair indicates the number of systems considered. Figures without * are taken from Mathur et al. (2020a).

to-English directions, which were part of WMT20 Metric Task (Mathur et al., 2020b) evaluation campaign[3]. Other metrics are included for a comparison. At the segment-level, we report the Kendall's Tau correlation of segment-level metric scores with DARR human assessment scores, see Table 2. We use the same Kendall's Tau-like formulation which was used by Mathur et al. (2020b) in WMT20 evaluation campaign.

On average, the baseline outperforms the traditional MT evaluation metrics (SENTBLEU, BLEU) as well as the recently proposed ones that performed very well in the WMT20 Metric Task (PRISM, YISI-2), though for some of the translation directions (e.g. cs-en) MTEQA F1 is much worse (but for cs-en YISI-2 also does not beat BLEU). The segment-level correlation is much lower, even negative for some directions (e.g. zh-en) .

3.2 Generating Additional Answers

Since the QG system generates a single question for each sub-sequence of words marked as an extracted answer, the limit factor is the number of gold-standard answers we extract. To generate more questions we need more keyphrases to be asked about.

Considering the whole predictive power of the MTEQA metric is based on questions, we used linguistic processing of the sentence based on Part-of-Speech (POS) pattern matching and Named Entity Recognition (NER) to extract more keyphrases.

Given a sentence as the input, first, we parse the sentence using UDPipe (Straka et al., 2016) to extract part of speech (POS) tags. Then, we extract phrases that are matched with one of the patterns in our POS pattern bank. The POS pattern bank is created by parsing the sentences from XQuAD (Artetxe et al., 2020) dataset, extracting the POS patterns corresponding to the gold-standard answers, and taking the most frequent patterns. This dataset contains professional translations of the development set of SQuADv1, translated into various languages from different language families and using different scripts. Second, we extract named entities mentioned in the input sentence using a combination of two multilingual NER models, POLYGLOT-NER (Al-Rfou et al., 2015), and Stanza (Qi et al., 2020). Finally, we output the union of the extracted phrases and named entities as the potential answers. At both system- and segment-level using the MTEQA F1 KEYPHRASE variant yields improvements for most of the translation directions.

3.2.1 Tuning the Answer Comparison Metric

The choice of the Answer Comparison Metric can have a considerable impact on the final performance. Using the word-level F1 metric, given the gold-standard answer *"Tchaikovsky"*, both the *"Tchaikovski"* and *"Beethoven"* would get the same score. In the context of MT, it may be worth to consider a more fine-grained comparison.

We decided to use the CHRF (Popović, 2015) metric, since it operates on the level of characters, and enables scoring even partial matches. Using the MTEQA CHRF KEYPHRASE variant yields further improvements at both system- and segment-level.

For the WMT21 Metrics Shared Task we submit this variant of the metric – the gold-standard answers are extracted by POS pattern matching and NER, and the chrF metric is used for answer comparison (MTEQA CHRF KEYPHRASE).

[3] cs, de, ja, pl, ru, ta, zh, iu, km, ps → en

	cs-en	de-en	zh-en	en-de	en-cs
MTEQA F1	-0.422^*	0.041^*	-0.430^*	-0.581^*	-0.480^*
MTEQA F1 KEYPHRASE	-0.108^*	0.273^*	-0.058^*	-0.016^*	0.100^*
MTEQA CHRF KEYPHRASE	0.017^*	0.327^*	0.030^*	0.159^*	0.227^*
SENTBLEU	0.068	0.413	0.093	0.303	0.432
PRISM	**0.143**	**0.475**	**0.167**	**0.447**	**0.619**
YISI-2	0.068	0.413	0.116	0.296	0.187

Table 2: Segment-level Kendall's Tau correlation for a few metrics used for measuring MT quality with DARR human assessment scores, over MT systems using the *newstest2020* references. Numbers without * are taken from (Mathur et al., 2020a).

4 MQM scores

Recently, Freitag et al. (2021) demonstrated that the WMT DA method traditionally used for human evaluations has actually lower correlation with expert-based labels than the Multidimensional Quality Metrics (MQM) scoring method developed in the EU QTLaunchPad and QT21 projects. Following their findings, the WMT21 Metric Task will report the correlation with MQM labels in the official results.

To provide a more complete picture of the performance of the MTEQA metric, we also report correlation with the MQM assessments. Table 3 presents the system-level Pearson correlation of the metric with both the MQM and DA labels for 8 systems that were re-annotated by Freitag et al. (2021) and are available from GitHub[4].

The results are surprising and to a large extent unintuitive. Metrics performing well in comparison with MQM are often bad in comparison with DA.

5 Conclusions

In this paper we described our submission to the WMT21 Metrics Shared Task. We showed that the degree to which the MT output can be used to answer questions about the reference can be used as a proxy to evaluate the translation quality.

We showed a gradual improvement of our submission. We examined a linguistically motivated way of extracting keyphrases from the sentence, and showed that it boosts both the segment- and system-level correlation with DA human judgments. We were able to further boost the final performance by using the CHRF metric to compare the reference and test answers.

Finally, we examined the performance against the MQM labels and compared the performance against the DA labels.

[4] https://github.com/google/wmt-mqm-human-evaluation

Acknowledgements

This work was supported by the European Commission via its H2020 Program (contract no. 870930) and CELSA (project no. 19/018), and has been using data and tools provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (project no. LM2018101).

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Rachel Bawden, Biao Zhang, Andre Tättar, and Matt Post. 2020. ParBLEU: Augmenting metrics with automatic paraphrases for the WMT'20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 887–894, Online. Association for Computational Linguistics.

Jan Berka, Martin Černý, and Ondřej Bojar. 2011. Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95.

	zh-en		en-de	
	MQM	DA	MQM	DA
MTEQA CHRF KEYPHRASE	0.630	**0.818**	0.761	0.394
PRISM	0.778	0.351	**0.989**	0.607
COMET	**0.889**	0.188	0.965	**0.628**
PARBLEU	0.380	0.565	0.722	0.218
CHRF	0.523	0.579	0.853	0.576
TER	0.352	0.511	0.810	0.477

Table 3: System-level Pearson correlation for selected metrics used for measuring MT quality with the DA and MQM labels, computed for the *newstest2020* references and the 8 MT systems re-annotated by Freitag et al. (2021).

Sheila Castilho and Ana Guerberof Arenas. 2018. Reading comprehension of machine translation output: What makes for a better read? In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 79–88, Alacant/Alicante, Spain. European Association for Machine Translation.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *arXiv preprint arXiv:2104.14478*.

Masaru Fuji, Hatanaka N, Ito E, Kamei S, Kumai H, Sukehiro T, Yoshimi T, and Isahara Hitoshi. 2001. Evaluation method for determining groups of users who find mt useful. In *MT Summit VIII: Machine Translation in the Information Age*, pages 103–108.

Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens, and Pavel Pecina. 2021. Just ask! evaluating machine translation by asking and answering questions. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Kateřina Macková and Milan Straka. 2020. Reading comprehension in czech via machine translation and cross-lingual transfer. In *23rd International Conference on Text, Speech and Dialogue*, pages 171–179, Cham, Switzerland. Springer.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020a. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Niels Ott, Ramon Ziai, and Detmar Meurers. 2012. Creation and analysis of a reading comprehension exercise corpus. *Multilingual corpora and multilingual corpus analysis*, 14:47.

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.

Suraj Patil. 2020. Question generation. https://github.com/patil-suraj/question_generation.

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. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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'16)*, pages 3652–3658, Portorož, Slovenia. European Language Resources Association (ELRA).

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Masaru Tomita, Shirai Masako, Tsutsumi Junya, Matsumura Miki, and Yoshikawa Yuki. 1993. Evaluation of mt systems by toefl. In *Proceedings of the Theoretical and Methodological Implications of Machine Translation (TMI-93)*, pages 252–265.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Are References Really Needed? Unbabel-IST 2021 Submission for the Metrics Shared Task

Ricardo Rei[1,2,4] **Ana C. Farinha**[1] **Chrysoula Zerva**[2,3] **Daan van Stigt**[1] **Craig Stewart**[1]
Pedro G. Ramos[1] **Taisiya Glushkova**[2,3] **André F. T. Martins**[1,2,3] **Alon Lavie**[1]
[1]Unbabel [2]Instituto Superior Técnico [3]Instituto de Telecomunicações [4]INESC-ID
[2,3,4]Lisbon, Portugal
{ricardo.rei, chrysoula.zerva, taisiya.glushkova, andre.t.martins}@tecnico.ulisboa.pt
{catarina.farinha, daan.stigt, craig.stewart, pedro.ramos, alon.lavie}@unbabel.com

Abstract

In this paper, we present the joint contribution of Unbabel and IST to the WMT 2021 Metrics Shared Task. With this year's focus on *Multidimensional Quality Metric* (MQM) as the ground-truth human assessment, our aim was to steer COMET towards higher correlations with MQM. We do so by first pretraining on *Direct Assessments* and then fine-tuning on z-normalized MQM scores. In our experiments we also show that reference-free COMET models are becoming competitive with reference-based models, even outperforming the best COMET model from 2020 on this year's development data. Additionally, we present COMETINHO, a light-weight COMET model that is 19x faster on CPU than the original model, while also achieving state-of-the-art correlations with MQM. Finally, in the "QE as a metric" track, we also participated with a QE model trained using the OPENKIWI framework leveraging MQM scores and word-level annotations.

1 Introduction

In this paper, we present the joint contribution of Unbabel and IST to the WMT 2021 Shared Task on Metrics. We participated in the segment-level and system-level tracks, as well as the "QE as a Metric" task.

Similar to our participation last year (Rei et al., 2020b), most of the models are based on the COMET framework[1] (Rei et al., 2020a). In last year's shared task (Mathur et al., 2020), COMET along with other trainable metrics such as PRISM (Thompson and Post, 2020) and BLEURT (Sellam et al., 2020) showed superior correlations with the *Direct Assessments* (DA) collected for the News Translation Shared Task. This year, we build on top of the models used last year to take into account that human assessments will be carried out using variants of the *Multidimensional Quality Metric* (MQM) (Lommel et al., 2014) framework and no longer based on DA (Graham et al., 2013). For this reason, we extended our training dataset to include DA evaluations from WMT ranging 2015 to 2020, with the exception of *en-de* and *zh-en* for which we do not include the 2020 data given that the same is included in the MQM development data (Freitag et al., 2021). Finally, we fine-tuned these new models on the z-normalized MQM scores provided for this year's shared task.

One of the remaining redeeming qualities of automated metrics such as BLEU (Papineni et al., 2002) is that they are incredibly light-weight. Despite the higher correlation with human judgement, trainable metrics tend to be slower to run. In an effort to close this gap we present COMETINHO, a light-weight model based on the COMET framework that replaces the original XLM-R large encoder with MiniLMv2 (Wang et al., 2020). This model is approximately 19x faster at inference time compared to the original COMET model (Rei et al., 2020a) and maintains state-of-the-art correlations with MQM in reference-based evaluations.

For the "QE as a metric" track, we show that reference-free evaluation models can reach surprisingly high correlations with human judgements and are competitive with their corresponding reference-based models. Last year we also participated with a similar model in the Metrics Shared Task, but here we elaborate in more detail on the primary differences between this model architecture and other COMET models.

Finally, and for the first time, we submit and describe a reference-free model that in addition to learning from MQM scores also makes use of word-level error annotations. This is possible this year given the shift in evaluation method from DA

[1]**C**rosslingual **O**ptimized **M**etric for **E**valuation of **T**ranslation hosted at: https://github.com/Unbabel/COMET

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1030–1040
November 10–11, 2021.

Tags	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	BAD	BAD
MT	the	main	purpose	of	this	project	is	to	design	a	car	for	blind	driving.

Source: 这个项目的主要目的 是设计一辆盲人驾驶的车。
Reference: the main goal of this project is to develop a car for the blind.

Table 1: Example of word-level OK and BAD tags produced by our OPENKIWI model trained with word-level annotation spans. This translation received an overall sentence score of 0.2 and the model was able to identify that the words "blind driving" are translation errors giving a good insight on why the sentence score is low.

to MQM. This model uses the OPENKIWI [2] architecture and its word-level tagging feature to predict OK/BAD word tags along with a sentence-level quality score.

2 The COMET Framework

For a more comprehensive description of the COMET architecture we direct the reader to the original paper (Rei et al., 2020a). Below we will highlight some relevant features that contrast with the COMET reference-free model (COMET-QE). In COMET we encode segment-level representations using the pretrained, cross-lingual, model XLM-RoBERTa (Conneau et al., 2020). Even though we encode the source, the hypothesis, and the reference (i.e. the human curated translation of the source) separately, their embeddings are mapped into a shared feature space. Subsequently, we obtain combined features using the three embeddings ($\boldsymbol{s}$, $\boldsymbol{h}$, and $\boldsymbol{r}$, for the source, hypothesis, and reference, respectively): $\boldsymbol{h} \odot \boldsymbol{s}$, $\boldsymbol{h} \odot \boldsymbol{r}$, $|\boldsymbol{h} - \boldsymbol{s}|$, and $|\boldsymbol{h} - \boldsymbol{r}|$. These features, concatenated to $\boldsymbol{r}$ and $\boldsymbol{h}$ and the resulting vector is the input to a feed-forward regressor.

2.1 Reference-free COMET

The architecture of the COMET model used in the "QE as a metric" task (COMET-QE) is very similar to the main COMET model (Rei et al., 2020a) briefly described above and RUSE (Shimanaka et al., 2018). The biggest difference being that in the COMET-QE model the reference is not used and, consequently, the combination of features used as input to the feed-forward regressor are also different from reference-based COMET. In this case, the combined features are simply: $\boldsymbol{h} \odot \boldsymbol{s}$ and $|\boldsymbol{h} - \boldsymbol{s}|$; the final vector to the feed-foward regressor being the concatenation of the latter features together with $\boldsymbol{h}$ and $\boldsymbol{s}$. A schematic representation is shown in Figure 1.

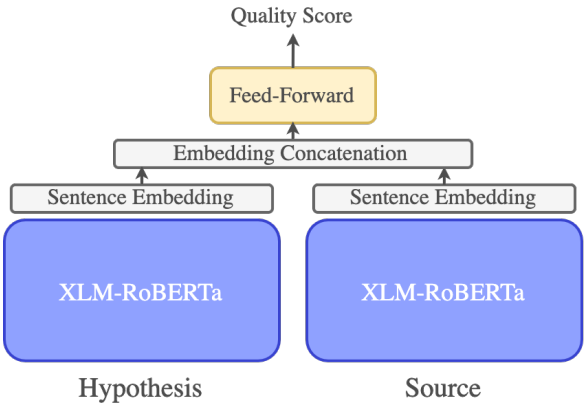


Figure 1: The COMET-QE model follows the dual encoder architecture proposed in RUSE (Shimanaka et al., 2018) but replacing the reference translation with the source sentence.

3 Lightweight COMET: COMETINHO

Our light-weight version of the original COMET model is almost an exact replica in terms of architecture save that we replaced the underlying pre-trained encoder with MiniLMv2 (Wang et al., 2020) which is a distilled version of XLM-R large (Conneau et al., 2020). This distilled model is made available by HuggingFace Transformers (Wolf et al., 2020): `nreimers/mMiniLMv2-L6-H384-distilled-from-XLMR-Large`

Our COMETINHO models are 19x faster on CPU and 14.3x times faster on GPU than COMET models based on XLM-R large. Also, in terms of disk footprint, these models are 5x smaller [3].

4 The OPENKIWI Framework

When using the MQM framework for the calculation of the quality score, human annotators seek to identify and annotate error spans at the word-level, as well as the severity of those errors. We

[2] **OpenKiwi** hosted at: https://github.com/Unbabel/OpenKiwi

[3] Contrastive inference times were tested using a 2.3 GHz Intel Core i5 for CPU, and using a Nvidia T4 for GPU.

leveraged these word-level annotations using the OpenKiwi framework (Kepler et al., 2019), by transforming each word into an OK or BAD tag. In the OpenKiwi architecture, in contrast with COMET-QE, source and hypothesis are jointly encoded. A sentence pair representation is then obtained using average pooling over the hypothesis word embeddings and then used as features to a feed-forward regression layer that learns to produce a sentence level score. At the same time, the word embeddings from the hypothesis are used to predict OK/BAD tags and therefore, the model is trained in a multitask setting (regression and sequence labelling).

5 Corpora

In this year's shared task the organisers provided a development set with MQM annotations for the *en-de* and *zh-en* participating systems on WMT20 (Freitag et al., 2021). Apart from the official development data we used all the Direct Assessments available from previous years.

5.1 Multi-dimensional Quality Metric Corpus

In this corpus, for each language pair, each translation was annotated by 3 raters from a pool of 6. Following what is a common practice for the DA's we convert the segment-level scores of each annotator into a z-normalized score and the final translation quality score is an average of the 3 z-scores. Also, because the sign of these MQM annotations is the opposite of the Direct Assessments we invert the score. Subsequently we generate a train and test split leaving 20% of the documents for each language pair for testing. This results in a total of 11230 *en-de* training samples and 15600 *zh-en* training samples, with testsets of 2950 and 4400 samples, respectively. All results reported in this paper are with respect to the above train and test split. The documents contained in each split are listed in the Appendix of this paper.

Annotators are not always consistent and the annotations of one annotator might differ from another (Graham et al., 2017). With this in mind, we decided to calculate the Kendall's Tau correlation between all annotators as a measure of inter-annotator agreement (Figure 2). The inter-annotator Kendall Tau can then be used as a ceiling effect for the developed metrics which ideally should behave as an additional annotator.

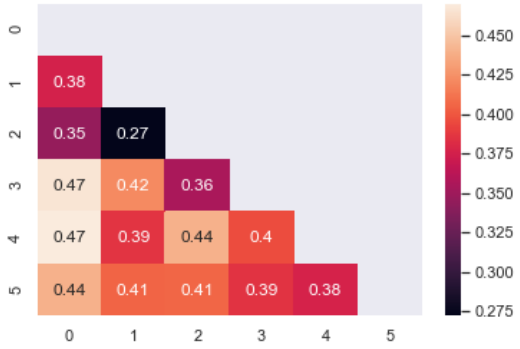


Figure 2: Kendall Tau Correlations between the *en-de* annotators used to develop the shared task development set (Freitag et al., 2021).

For training of the OpenKiwi model described herein we used proprietary MQM data from the customer support domain, covering several industries such as tech industry and travel industry. This data is composed by 1.1M (source, hypothesis) pairs with corresponding MQM annotations from 38 language pairs mostly out-of-english.

5.2 Direct Assessments

Each year, the WMT News Translation shared task organisers collect human judgements in the form of Direct Assessments. Those assessments are then used in the Metrics task to measure the correlation between metrics and therefore decide which metric works best. In recent years researchers have been using these annotations to create trainable metrics that regress on these scores (Shimanaka et al., 2018; Sellam et al., 2020; Rei et al., 2020a). We follow the same approach and use Direct Assessments ranging from 2015 to 2020 for training. The collective corpora contain a total of 33 language pairs including low-resource languages such as English-Tamil (*en-ta*) and a total of 795269 tuples with source, hypothesis, reference and direct assessment z-score. The only exception to this data is that we did not include the *en-de* and *zh-en* assessment from 2020 because they overlap with the MQM development data described in section 5.1.

6 Segment-level task

The COMET framework is highly flexible and easy to adapt to different types of human judgements (Rei et al., 2020a). This year we first pre-trained on the DA collected from 2015 to 2020 except for *en-de* and *zh-en* as described above. Like in Glushkova et al. (2021) we trained 5 models for 1 epoch each using 5 different seeds and created an ensembled

	Nº Segments	zh-en 4400		en-de 2950			
		Pearson	Kendall	Pearson	Kendall	Pearson Avg.	Kendall Avg.
Baselines	BLEURT	0.492	0.405	0.107	0.060	0.299	0.232
	PRISM	0.399	0.337	0.072	0.020	0.235	0.178
	BERTSCORE	0.441	0.344	0.116	0.060	0.279	0.202
	BLEU	0.196	0.275	0.062	0.004	0.129	0.140
	CHRF	0.267	0.219	0.119	0.059	0.193	0.139
	COMET-DA (2020)	0.538	0.435	0.425	0.282	0.481	0.359
Ref. based	COMET-DA (2021)	0.559	0.454	0.464	0.309	0.511	0.382
	COMET-MQM (2021)	**0.717**	**0.546**	**0.488**	**0.361**	**0.602**	**0.454**
	COMETINHO-DA	0.484	0.386	0.299	0.204	0.392	0.295
	COMETINHO-MQM	0.670	0.496	0.311	0.237	0.490	0.367
Ref. Free	COMET-QE-DA (2021)	0.567	0.436	**0.497**	0.308	0.532	0.372
	COMET-QE-MQM (2021)	**0.720**	**0.531**	0.470	**0.359**	**0.595**	**0.445**
	OPENKIWI	0.522	0.385	0.448	0.287	0.485	0.336

Table 2: Segment-level correlations on the *en-de* and *zh-en* testset.

model (COMET-DA). During our experiments we tested two ensembling techniques; averaging the different model predictions and averaging the parameters from the 5 models. Those two approaches had similar results but in the end we decided to use the later one for performance.

Subsequently, we fine-tuned each of the 5 models on the MQM data provided as development for another epoch. As before, we performed weight averaging to obtain an ensemble of those models (COMET-MQM). In both the pre-training and fine-tuning we only perform 1 training epoch in order to ensure that the final models are able to generalise to many language pairs and do not overfit to the News domain. This is especially important since the MQM dataset only contains *en-de* and *zh-en*.

For COMETINHO, as previously mentioned, we used the distilled version of XLM-R (MiniLMv2), available through Hugging Face, and we followed the same training recipe where we pre-train the model using DA's for 1 epoch and then we adapt the model to the MQM data for another epoch.

7 System-level task

For the System-level task we compute the system-level score for each system by averaging the segment-level scores obtained. This follows the same approach used to compute system-level scores based on segment-level human annotations such as DA's and MQM which means that a metric that achieves strong segment-level correlation should also achieve strong system-level performances.

8 QE as a Metric Task

We trained a reference-free model (COMET-QE) in the same way we did for reference-based COMET models described in section 6. As described in section 2.1, the primary difference between the two models is the inclusion or exclusion of the source as input.

9 Experimental Results

9.1 Segment-level task

Reference-based segment-level correlations on the *en-de* and *zh-en* testsets are shown in Table 2. We used both Pearson and Kendall Tau correlation metrics to evaluate our models. As baselines we used lexical metrics such as CHRF (Popović, 2015) and BLEU (Papineni et al., 2002), an embedding-based metric BERTSCORE (Zhang et al., 2020) and three trainable-metrics; BLEURT (Sellam et al., 2020), PRISM (Thompson and Post, 2020) and COMET-DA (2020) (Rei et al., 2020b).

The fact that the COMET-DA (2021) gives higher correlations than the COMET-DA (2020) shows that adding more training data and combining checkpoints trained on different seeds already provides a boost in performance. However, fine-

		All systems			Human vs MT		
		en-de	en-zh		en-de	en-zh	
Nº Comparisons		45	45		21	16	
		Kendall		Avg	Kendall		Avg
Baselines	BLEU	0.378	0.311	0.345	0.095	0.077	0.086
	CHRF	0.444	0.422	0.433	0.143	0.000	0.072
	BERTSCORE (F1)	0.356	0.356	0.356	0.143	0.000	0.072
	PRISM	0.444	0.422	0.433	0.143	0.077	0.110
	COMET-DA (2020)	0.822	0.533	0.678	0.714	0.231	0.473
Ref. based	COMET-DA (2021)	0.844	0.489	0.667	0.761	0.231	0.496
	COMET-MQM (2021)	0.867	0.778	0.823	0.762	0.875	0.819
	COMETINHO-DA	0.533	0.378	0.456	0.238	0.000	0.119
	COMETINHO-MQM	0.355	0.311	0.333	0.095	0.000	0.048
Ref. Free	COMET-QE-DA (2021)	0.778	0.778	0.778	0.667	0.938	0.803
	COMET-QE-MQM (2021)	0.933	0.800	0.867	**1.000**	**1.000**	**1.000**
	OPENKIWI	0.822	0.733	0.778	0.762	0.769	0.766

Table 3: System-level Kendall's Tau (τ) correlations for all system combinations (on the left) and Human vs MT (on the right).

tuning on the MQM development data was the most significant addition to previous work: the COMET-MQM (2021) model increased on average more than 0.1 Pearson correlation. This improvement is consistent with regard to the two COMETINHO models (with COMETINHO-MQM having notably higher correlations than COMETINHO-DA). Nevertheless, the fact that COMETINHO-DA has competitive or state-of-the-art performance with all the other metrics such as BLEURT, PRISM, and BERTSCORE, while also being much faster, presents an ideal opportunity for future work to investigate the incorporation of trainable metrics into the training objectives of MT systems.

For reference-free metrics, the fine-tuning on the MQM data, on average, gave a boost in performance (the only exception being the Pearson correlation for the *en-de* where COMET-QE-DA has a slightly higher correlation than COMET-QE-MQM). Overall, it is somewhat surprising that COMET-QE-* (2021) and COMET-* (2021) show relatively comparable correlations, suggesting that using the reference as input for MT evaluation might be less useful than expected and could feasibly become redundant. This surprising result was also reported by Kocmi et al. (2021) and is especially important since curating reference sentences is usually costly and time consuming and can introduce undesired bias in the evaluation (Freitag et al., 2020).

Finally, the OPENKIWI model has competitive correlations when looking to other trainable metrics and to COMET models that were not fine-tuned on the MQM development data. This add further weight to the suggestion above that references might not add substantial value to MT evaluation. Its performance is even more surprising when considering the fact that this model was train with data from a completely different domain.

It is worth highlighting that the Kendall's Tau correlations for all models (with exception of the two reference-based COMETINHO models) are in the range obtained for correlations between different annotators, for *en-de*, Figure 1. This further validates the value of our models.

9.2 System-level task

System-level results are presented in Table 3 where we report a Kendall Tau correlation defined as follows:

$$\tau = \frac{Concordant - Discordant}{Concordant + Discordant} \quad (1)$$

where *Concordant* defined as the number of times a metric agrees with humans that a given system x is better than a given system y and *Discordant* is the opposite. These decisions are the computed for all combinations of systems in the testset.

Due to the low number of systems and the relative proximity of the ground-truth MQM system scores we also compare metrics on their ability to distinguish human references from MT outputs. With reference to table 7 in the appendix we note that, for *zh-en*, all 8 MT systems demonstrate comparable performance but that there is a clear separation of human translations. For that reason Table 3 also presents the Kendall Tau correlations considering only "Human" systems against MT systems where we can observe that reference-free metrics achieve better performance. This results confirms the finding from last year's shared task (Mathur et al., 2020) where COMET-QE was highlighted as being the only metric able to differentiate human translations from MT.

10 Related work

Classic n-gram matching MT evaluation metrics such as BLEU (Papineni et al., 2002) have been adopted by the MT community as a primary form of MT evaluation, yet, in the recent years of the WMT Metrics shared task (Bojar et al., 2017; Ma et al., 2018, 2019; Mathur et al., 2020) these classic metrics have been outperformed first by embedding-based alternatives and more recently by trainable metrics based on pre-trained models.

With the rise of word embeddings (Pennington et al., 2014; Peters et al., 2018; Devlin et al., 2019), metrics such as BLEU2VEC (Tättar and Fishel, 2017) and MEANT 2.0 (Lo, 2017) replaced the typical word/n-gram matching by fuzzy matches based on distributional word representations. These metrics appeared for the first time at the WMT Metrics task in 2017 with MEANT 2.0-SRL achieving the highest results at segment-level. In 2018 and 2019 YISI-1 (Lo, 2019), a successor of MEANT 2.0 (Lo, 2017), was among the winners of the WMT Metrics task. YISI-1 (Lo, 2019) mostly takes advantage of BERT embeddings (Devlin et al., 2019) to create soft alignments between hypothesis and reference.

Trainable metrics started as simple regressions based on lexical features (e.g BLEND (Ma et al., 2017)) but nowadays these metrics also use embeddings to extract features that are then used to regress on quality assessments. The first of such metrics were RUSE (Shimanaka et al., 2018) and ESIM (Mathur et al., 2019) which were based on RNN encoders and worked mostly for English. In 2020, BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020a) were proposed. Both metrics used pre-trained transformer based encoders to extract sentence-level features that are then passed to a regression model; the difference is that COMET also extracts features for the source segment which was something overlooked by predecessor metrics. In the 2020 Metrics Shared task both COMET and BLEURT achieved some of the highest correlations with human judgements and shared the podium with PRISM (Thompson and Post, 2020)

11 Conclusions

In this paper we present the Unbabel-IST's contribution to the WMT 2021 Metrics shared task which for the first time, introduced evaluation using MQM. Our specific contributions include; the fine-tuning of Direct Assessment based models on MQM data which yields impressive gains on the described test sets and a new, lightweight COMET model which achieves comparable performance to its predecessors. Such a light model can provide interesting opportunities for future work into the incorporation of modern metrics into MT training. Finally, but perhaps our most important contributions; we further validate the observations in (Kocmi et al., 2021) that QE as a metric is becoming competitive as an alternative to reference-based evaluation, and, we show that a word-level QE system can be successfully trained on MQM annotations and be competitive with current trainable metrics while providing some intuition about "what" is wrong with a specific translation.

Acknowledgments

We are grateful to Fabio Kepler and José G. C. de Souza for their valuable feedback and discussions. This work was supported by the P2020 programs MAIA (contract 045909) and Unbabel4EU (contract 042671), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag, George F. Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *CoRR*, abs/2104.14478.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-Aware Machine Translation Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Online. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, A. Moffat, and J. Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

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.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *CoRR*, abs/2107.10821.

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. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle 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*, 0:455–463.

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: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. 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, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. 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: Shared Task Papers*, pages 751–758, Belgium, Brussels. Association for Computational Linguistics.

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.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2020. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *CoRR*, abs/2012.15828.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

A.1 COMET Hyper-Parameters

In Table 5 is an excerpt of the training configuration used for training the COMET-DA model and Table 5 for the COMET-QE-DA. Then these models are fine-tuned for 1 extra epoch with same hyperparameters except the `learning_rate` that is decreased to $1.0e-05$ and the `nr_frozen_epochs` which we increase to 1 to completely freeze the encoder model.

A.2 OPENKIWI Hyper-Parameters

The hyperparameters used for the OpenKiwi model are expressed in Table 4 and follows the configurations proposed in the sample file of the `github` repository[4].

System	
batch_size	2
Encoder	
hidden_size	1024
Decoder	
bottleneck_size	1024
dropout	0.05
hidden_size	1024
Optimizer	
class_name	adam
encoder_learning_rate	0.0001
learning_rate_decay	1.0
learning_rate_decay_start	0
learning_rate	0.0001
Trainer	
training_steps	2180
early_stop_patience	10
validation_steps	0.5
gradient_accumulation_steps	4
gradient_max_norm	1.0

Table 4: Hyperparameters for OPENKIWI MQM model

nr_frozen_epochs	0.3
keep_embeddings_frozen	True
optimizer	AdamW
encoder_learning_rate	1.0e-05
learning_rate	3.1e-05
layerwise_decay	0.95
encoder	XLM-RoBERTa
pretrained_model	xlm-roberta-large
pool	avg
layer	mix
dropout	0.15
batch_size	4
gradient_accumulation_steps	4
hidden_sizes	[3072, 1024]
epochs	1

Table 5: Hyper-parameters for fine-tuning Reference-based COMET model on Direct Assessments.

nr_frozen_epochs	0.3
keep_embeddings_frozen	True
optimizer	AdamW
encoder_learning_rate	1.0e-05
learning_rate	3.1e-05
layerwise_decay	0.95
encoder	XLM-RoBERTa
pretrained_model	xlm-roberta-large
pool	avg
layer	mix
dropout	0.15
batch_size	4
gradient_accumulation_steps	4
hidden_sizes	[2048, 1024]
epochs	1

Table 6: Hyper-parameters for fine-tuning Reference-free COMET model on Direct Assessments.

[4] https://github.com/Unbabel/OpenKiwi/blob/master/config/xlmroberta.yaml

en-de		zh-en	
System	MQM	System	MQM
Human-B.0	0.794	Human-A.0	3.114
Human-A.0	0.933	Human-B.0	3.149
Human-P.0	1.547	Huoshan_Translate.919	5.077
Tohoku-AIP-NTT.890	2.043	Tencent_Translation.1249	5.163
OPPO.1535	2.284	OPPO.1422	5.309
Tencent_Translation.1520	2.333	THUNLP.1498	5.389
Online-B.1590	2.516	DeepMind.381	5.442
eTranslation.737	2.530	WeChat_AI.1525	5.469
Huoshan_Translate.832	2.600	DiDi_NLP.401	5.484
Online-A.1574	3.189	Online-B.1605	5.512

Table 7: System-level Ranking and corresponding MQM scores for the test split described in section 5.1

A.3 Train/Test Split Documents

In our train/test split described in section 5.1 we leave the following documents for testing:

- *reuters.276709*
- *cnbc.com.33889*
- *cnn.385672*
- *aj-english.8643*
- *express.co.uk.10983*
- *cbsnews.302258*
- *sky.com.20683*, *chicago_defender.80*
- *sciencedaily.com.75569*
- *seattle_times.7141*
- *huffingtonpost.com.19389*
- *huffingtonpost.com.19385*
- *upi.205721*
- *dailymail.co.uk.365293*
- *upi.205735*
- *standard.co.uk.14562*
- *foxnews.100085*
- *allafrica.15342*
- *abcnews.364021*
- *kcal.279*
- *sky.com.20667*
- *en.ndtv.com.13143*
- *reuters.276541*
- *heraldscotland.com.7318*
- *foxnews.100073*
- *upi.205695*
- *tsrus.cn.2113*
- *chinanews.com.102574*
- *chinanews.com.102805*
- *chinanews.com.102708*
- *xinhua-zh-01.6415*
- *chinanews.com.102657*
- *chinanews.com.102700*
- *chinanews.com.102573*
- chinanews.com.102534
- *chinanews.com.102914*
- *tsrus.cn.2112*
- *xinhua-zh-01.6608*
- *australian-zh.104*
- *chinanews.com.102580*
- *xinhua-zh-01.6520*
- *chinanews.com.102767*
- *chinanews.com.102748*
- *chinanews.com.102807*
- *international_times-zh.165*
- *chubun-zh.1066*
- *international_times-zh.160*
- *international_times-zh.150*
- *xinhua-zh-01.6434*
- *xinhua-zh-01.6586*
- *xinhua-zh-01.6307*
- *xinhua-zh-01.6529*
- *chinanews.com.102780*
- *hunan_ribao-zh.199*
- *chinanews.com.102737*
- *chinanews.com.102722*
- *chinanews.com.102709*

Regressive Ensemble for Machine Translation Quality Evaluation

Michal Štefánik and **Vít Novotný** and **Petr Sojka**
Masaryk University, Faculty of Informatics, MIR group https://mir.fi.muni.cz
{stefanik.m,witiko}@mail.muni.cz, sojka@fi.muni.cz

Abstract

This work introduces a simple regressive ensemble for evaluating machine translation quality based on a set of novel and established metrics. We evaluate the ensemble using a correlation to expert-based MQM scores of the WMT 2021 Metrics workshop. In both monolingual and zero-shot cross-lingual settings, we show a significant performance improvement over single metrics. In the cross-lingual settings, we also demonstrate that an ensemble approach is well-applicable to unseen languages. Furthermore, we identify a strong reference-free baseline that consistently outperforms the commonly-used *BLEU* and *METEOR* measures and significantly improves our ensemble's performance.

1 Introduction

Automated evaluation of text generation is challenging due to many orthogonal qualitative aspects that the user expects from a text generation system. In machine translation, we can observe errors of the so-called critical category (Wulczyn et al., 2017) such as hallucinating (Lee et al., 2019), omitting parts of the input from translation, or the negation of meaning (Matusov, 2019).

Consider an example of the last category:

- *Reference*: "I never wrote this article, I just edited it."
- *Hypothesis 1*: "It is not my article, I just edited it."
- *Hypothesis 2*: "I never wrote this article, I never edited it."

In this example, all *BERTScore* (Zhang et al., 2019), *BLEURt* (Sellam et al., 2020), and *Prism* (Thompson and Post, 2020b) metrics rank *Hypothesis 2* higher than *Hypothesis 1*. In *BLEURt* and *Prism*, this can be due to a known vulnerability of Transformers, which rely on a lexical intersection (McCoy et al., 2019) if such a heuristic fits the problem sufficiently well. Trivially, just counting the negations can easily remedy this specific problem. However, such a heuristic would fail in many other cases, such as when we adjust *Hypothesis 1* to "It is an article of somebody else, I just edited it.".

Such cases motivate our ensemble approach that aims to expose both surface and deeper semantic properties of texts and subsequently learn to utilize these for the specific task of translation evaluation. Even though the objective might constrain the particular metrics, data set, or systematically fail in some cases, another metric or their combination in the ensemble allows the flaw to be corrected.

2 Metrics for machine translation evaluation

This section reviews the related work, focusing on the metrics we used in our ensemble.

The standard and still widely-used surface-level metrics for the evaluation of machine translation quality are *BLEU* (Papineni et al., 2002), *ROUGE* (Lin, 2004), and *TER* (Snover et al., 2006). Surface-level metrics are not able to capture the proximity of meaning in cases where one text paraphrases the other, which is an ability commonly observed in deep neural language models (Lewis et al., 2020). One metric that addresses this flaw is *METEOR* (Banerjee and Lavie, 2005), which utilizes WordNet to account for synonymy, word inflection, or token-level paraphrasing.

Evaluation of semantic text equivalence is closely related to a problem of accurate textual representations (embeddings). The traditional method that we identify as relevant for the evaluation of segment-sized texts is FastText (Bojanowski et al., 2017). FastText learns representations of character n-grams from which it creates a unified representation of tokens by averaging. Additionally, a distance of a pair of texts can be computed directly from the token-level embeddings using methods such as the soft vector space model (the soft co-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1041–1048
November 10–11, 2021.

sine measure, *SCM*) (Novotný, 2018), or by solving a minimum-cost flow problem (the word mover's distance, *WMD*) (Kusner et al., 2015).

Similar matching is performed by *BERTScore* (Zhang et al., 2019), which uses internal token embeddings of a selected BERT layer optimal for the task. Although the token representations are multilingual in some models (Devlin et al., 2018), which makes *BERTScore* usable without references, we are not aware of prior work evaluating it as such. A possible drawback of cross-lingual alignment using a max-reference matching scheme of *BERTScore* lies in a possibility of a significant mismatch of sub-word tokens in source and target text. In contrast, the metric that we refer to as *WMD-contextual* uses the same embeddings as *BERTScore* but uses the network-flow optimization matching scheme of *WMD*.

Task-agnostic methods have recently been outperformed by methods that fine-tune a pre-trained model for a related objective: *BLEUrt* (Sellam et al., 2020) fine-tunes a *BERT* (Devlin et al., 2018) model directly on Direct Assessments of submissions to WMT to predict the judgements using a linear head over contextual embeddings of a classification *[CLS]* token. *Comet* (Rei et al., 2020) learns to predict Direct Assessments from tuples of source, reference, and translation texts with the triplet objective or the standard MSE objective.

Some of the most recent work incorporates latent objectives and/or data sets. For instance, *Prism* (Thompson and Post, 2020b,a) learns a language-agnostic representation from multilingual paraphrasing in 39 languages, thus being one of the few well-performing reference-free metrics. The orthogonality of its training objective might lower its correlation to other methods that use contextual embeddings.

3 Methodology

Our methodology aims to answer the following major question with additional supporting questions:

1. **Can an ensemble of surface, syntactic, and semantic-level metrics significantly improve the performance of single metrics?**

2. Can such an approach be applied cross-lingually, i.e., on languages that it has not been trained on?

3. Can surface-level metrics in reference-free configuration achieve results comparable to the reference-based ones?

4. Are contextual token representations important for evaluating semantic equivalence, or can these be replaced with pre-inferred token representations?

3.1 Experimental setup

We perform our primary evaluation on Multidimensional Quality Metrics (MQM) data set (Freitag et al., 2021), where we use averaged judgement scores as our gold standard. Where multiple judgements are available for the given pair of a source and a hypothesis, we average the scores over the judgements and consider this average as our gold standard. We split the samples into train (80%) and test (20%) subsets based on unique source texts.

In our experimental framework, which we release as an open-source Python library and Docker image for ease of reproduction[1][2], we implement a selected set of the metrics based on their guidelines, together with a bunch of novel metrics, introduced in Section 3.2 aiming to provide additional, orthogonal insight of textual equivalence.

Subsequently, we train a regressive ensemble on the *standardized* metric features of the whole train set, intending to predict the averaged MQM expert judgements. We evaluate the ensemble, together with all other selected metrics using pairwise Spearman's rank correlation (Spearman's ρ) with the MQM judgements on the held-out 20% test split.

In addition to our primary evaluation on the MQM data set, we perform our experiments on the Direct Assessments (DA) from WMT 2015 and 2016, and a dev set of Catastrophic errors from the Post-editing Dataset (Fomicheva et al., 2020a) of the Multilingual Quality Estimation Dataset (MLQE-PE) (Fomicheva et al., 2020b) used for evaluation at the Quality Estimation workshop of WMT 2021. Refer to Section 3.6 for a detailed description of our experiments on DA and MLQE-PE.

We performed all our evaluations on segment-level judgements. To minimize the impact of calibration for each of the specific metrics to evaluation, we report Spearman's rank correlation coefficient, reflecting the mutual qualitative ordering rather than particular values of the judgements.

[1] https://github.com/MIR-MU/regemt
[2] https://hub.docker.com/r/miratmu/regemt

3.2 Novel metrics

In addition to a selected set of metrics based on a literature review, we implement a set of novel metrics that allows our ensemble to reflect a wider variance of properties of the evaluated texts.

3.2.1 Soft Cosine Measure

The soft cosine measure (*SCM*) (Novotný, 2018) is the cosine similarity of texts in the soft vector space model, where the axes of terms are at an angle corresponding to their cosine similarity S in a token embedding space:

$$\text{SCM}(\vec{x}, \vec{y}) = \frac{\vec{x}^\mathsf{T} \cdot S \cdot \vec{y}}{\sqrt{\vec{x}^\mathsf{T} \cdot S \cdot \vec{x}} \cdot \sqrt{\vec{y}^\mathsf{T} \cdot S \cdot \vec{y}}} \quad (1)$$

where $\vec{x}$ is a weighted bag-of-words (BoW) vector of a reference (or a source in a reference-free setting), and $\vec{y}$ is a weighted BoW vector of a hypothesis.

We use *SCM* with two token representations:

1. We use the static token representations of FastText (Grave et al., 2018). We refer to the resulting metric as *SCM*.

2. We use the contextual token representations of *BERT* (Devlin et al., 2018) using the methodology of *BERTScore* (Zhang et al., 2019). We collect representations of all tokens segmented by the WordPiece (Wu et al., 2016) tokenizer, and we treat each unique (token, context) pair as a single term in our vocabulary.

   Subsequently, we *decontextualize* these representations as follows: For each WordPiece token, we average the representations of all (token, context) pairs in the training corpus. We refer to the resulting metric as *SCM-decontextualized*. Due to the multilingual character of the learned BERT token representations, this metric is applicable both in reference-based and source-based approaches.

In addition to two token representations, we also use two different SMART weighting schemes of Salton and Buckley (1988) for the BoW vectors $\vec{x}, \vec{y}$ and the construction of the term similarity matrix S:

1. We use raw term frequencies as *weights* in the BoW vectors, the `nnx` SMART weighting scheme, and we construct the term similarity matrix in the vocabulary order. We refer to the resulting metrics as *SCM* and *SCM-decontextualized*.

2. We use term frequencies discounted by inverse document frequencies as weights in the BoW vectors, the `nfx` SMART weighting scheme, and we construct the term similarity matrix in the decreasing order of inverse document frequencies (Novotný, 2018, Section 3). We refer to the resulting metrics as *SCM-tfidf* and *SCM-decontextualized-tfidf*.

3.2.2 Word Mover's Distance

The Word Mover's Distance (*WMD*) (Kusner et al., 2015) finds the minimum-cost flow F between vector space representations of two texts:

$$\text{WMD}(\vec{x}, \vec{y}) = \text{minimum cumulative cost } F^\mathsf{T} \cdot S$$
$$\text{subject to } \sum_j F_{ij} = x_i, \sum_i F_{ij} = y_j, \quad (2)$$

where $\vec{x}$ is an ℓ_1-normalized weighted BoW vector of a reference (or a source in reference-free setting), $\vec{y}$ is an ℓ_1-normalized weighted BoW vector of a hypothesis, and S is a term similarity matrix.

Similar to *SCM* described in the previous section, we experiment with two token representations: FastText embeddings of whole tokens (*WMD*) and decontextualized embeddings of WordPiece tokens (*WMD-decontextualized*). Additionally, we also use the contextual embeddings of WordPiece tokens (*WMD-contextual*) to show the impact of decontextualization on the metric performance: If the impact is negligible, future work could avoid the costly on-the-fly inference of BERT representation and significantly reduce the vocabulary size.

Similarly to *SCM*, we also use two different weighting schemes: raw term frequencies (*WMD-**) and term frequencies discounted by inverse document frequencies (*WMD-*-tfidf*).

3.2.3 Compositionality

Our custom metric that we refer to as Compositionality constructs a transition graph of an arbitrary text G_t based on directed, pairwise transitions of the tokens' part-of-speech (PoS) categories. As the models for PoS tagging are language-dependent, we use the compliant—though not always systematically aligned—schemes of tagging used for training the taggers in English (Weischedel et al., 2013), German (Brants et al., 2002), Chinese (Weischedel et al., 2013), and Norwegian (Unhammer and Trosterud, 2009).

Subsequently, we row-normalise the values of matrix G_t and we define a distance metric of *Compositionality* C for $x = G_{t_1}$ and $y = G_{t_2}$:

$$C(x, y) = \ell_1\text{-norm}(x_{ii} - y_{ii}), \quad (3)$$

where the PoS tags $i \in x$ and $i \in y$.

In our submission, we apply this metric only if the language belongs to a set of the languages for which we have a tagger: English, German, Chinese, or Norwegian. In reference-based evaluations, this constraint applies to the target language; in the case of reference-free evaluations, it applies to both source and target languages.

3.3 Ensemble

We ensemble the aforementioned metrics as predictors in a regression model, minimizing the residual between the average segment-level MQM scores and predicted targets.

We experiment with a wide range of simple regressors and observe a superior performance of simple approaches of fully connected, two-layer perceptron with 100-dimensional hidden layer and ReLu activation and linear regression with squared residuals. We report the results for *RegEMT* as the best-performing one of these classifiers picked on a 20% held-out validation subset of the train data set.

In addition to the ensemble of all available metrics, we evaluate a baseline regressive ensemble *Reg-base* using solemnly two surface-level features: character-level source and target length according to WordPiece (Wu et al., 2016) tokens.

3.4 Cross-lingual experiments

As expert judgments are incredibly costly to obtain, it is unrealistic to expect that the training data for the trained systems will be available in the future for a vast majority of language pairs containing under-resourced languages. To estimate the performance of all metrics on uncovered language pairs, we perform a cross-lingual evaluation on average MQM judgements of two available language pairs: ***zh-en*** and ***en-de***.

Where applicable, we fit the metric parameters on the train split of the non-reported language pair. Subsequently, we evaluate and report the results on a test split of the reported pair to MQM judgements.

3.5 Ablation study

To understand the impact of individual metrics in their roles as predictors for our ensemble, we use their pairwise correlations for systematic feature elimination.

In our ablation study, we iteratively select the metric with the highest Spearman's ρ to any other metric. We eliminate the selected metric from our ensemble by fitting a new regression model on the remaining features. We continue until all metrics are eliminated and evaluate the ensemble at each step of the process.

3.6 Additional evaluations

To allow for additional insight into the consistency of the results to other relevant evaluation sources, together with an evaluation of the metrics in the novel application of critical error recognition, we perform the experiments analogically on a DA data set of the WMT submissions from years 2015 and 2016, as well as to the Critical Errors dev set of MLQE-PE data set for reference-free metrics.

In the case of DA judgements, we use the assessments from the year 2015 as a training split and assessments from the year 2016 as a test split.

In the case of MLQE-PE, we split the data analogically to MQM by splitting the unique source texts in an 80:20 ratio. In this case, we consider as gold judgements the mean severity of error assigned by three annotators to each of the translations.

4 Results

Correlations to MQM judgements. Table 1 lists correlations to MQM for source-based, i.e., reference-free metrics (upper) and reference-based metrics (middle). Results reported for *RegEMT* fit a selected regression model on the estimates of all the other metrics available for a given evaluation scheme. As described in Section 3.3, we pick the evaluated regression model based on its performance on a held-out portion of the train set: a two-layer perceptron for the source-based ***zh-en*** pair and a simple linear regression in all other cases, with negligible mutual differences between regression models in performance on the validation set (below 2%). In the reports suffixed with ***X***, we fit the regression model on the other language pair than the one used for the evaluation.

The results suggest that a simple regressive ensemble can benefit from the variance of the predictors in a majority of the evaluated configurations, including the cross-lingual settings and other evaluated datasets. We observe the highest margins in correlations in the case of MQM judgements.

	RegEMT	*Prism*	*BERTScr*	*WMD-cont*	*WMD-dec*	*WMD-dec-tf*	*SCM-dec*	*SCM-dec-tf*	*Compos*	*Reg-base*	*Comet*	*SCM*	*SCM-tf*	*WMD*	*WMD-tf*	*BLEUrt*	*BLEU*	*METEOR*
MQM-src ***zh-en***	**.59**	.36	.44	.44	.29	.17	.19	.13	.13	.34								
MQM-src ***zh-en-X***	**.49**	.36	.44	.44	.29	.17	.19	.13	.13	.34								
MQM-src ***en-de***	**.36**	.09	.14	.06	.04	.07	.03	.02	.23	.28								
MQM-src ***en-de-X***	**.31**	.09	.14	.06	.04	.07	.04	.02	.23	.28								
MQM-ref ***zh-en***	**.62**	.45	.45	.43	.27	.21	.10	.26	.01	.35	.51	.19	.35	.29	.27	.48	.25	.28
MQM-ref ***zh-en-X***	**.62**	.45	.45	.43	.27	.21	.09	.25	.01	.31	.51	.19	.35	.29	.27	.48	.25	.28
MQM-ref ***en-de***	**.60**	.32	.22	.25	.32	.28	.33	.18	.12	.27	.48	.06	.14	.13	.07	.10	.13	.20
MQM-ref ***en-de-X***	.38	.32	.22	.25	.32	.28	.34	.17	.12	.29	**.48**	.06	.14	.13	.07	.10	.13	.20
DA 2016-src	**.84**	.72	.74	.73	.57	.51	.37	.51	.29	.18	.82	.39	.45	.44	.42	.81	.42	.50
DA 2016-tgt	.68	**.70**	.34	.25	.09	.10	.10	.24	.13	.04								
catastrophic-src	**.29**	.26	.13	.11	.12	.15	.13	.09	.10	.09								

Table 1: Results for Spearman's correlations with selected gold standards. From top: correlation of source-based metrics (top) and reference-based metrics (middle) to averaged scores of MQM expert judgements for specified languages. Results suffixed with ***X*** are evaluated cross-lingually: both ensemble metrics are trained on other language pairs than evaluated. (Bottom): Results for other data sets: Direct assessments of WMT 2016 submissions and dev set of Catastrophic translations from MLQE-PE data set; reported values are an average of correlations over all available language pairs.

Baseline ensemble. Table 1 shows that *Reg-base*, using only the counts of reference and hypothesis Word-pieces, demonstrates its consistent superiority over the standard surface-level metrics of *BLEU* and *METEOR*, even in the cross-lingual vs. monolingual comparison. With respect to the MQM judgements, the correlations of *Reg-base* are reasonably consistent; hence, in the reference-free cases of ***en-de*** language pair, the correlation of *Reg-base* is very close to the correlations of *RegEMT*.

Importance of contextualization. The results in Table 1 are inconsistent concerning the importance of contextualization in token-level metrics (*WMD-cont** vs. *WMD-dec** and *SCM-cont** vs. *SCM-dec**). We observe a significant (15–16%) decrease of correlation between a contextualized and decontextualized versions of WMD in *all* cases of ***zh-en*** language pair. The situation differs in ***en-de*** pair, where for the reference-based case, the correlation of decontextualized version of WMD is superior by 7%.

Metrics correlations. Table 2 demonstrates mutual correlations of the evaluated metrics. We see the strong pairwise correlations among the metrics based on contextualized representations, such as between *Comet*, *Prism*, *BERTScore* and *BLEURt*; all of these are higher than 0.79. The situation is similar among the metrics based on static token representations of *SCM* and *WMD*, both with and without TF-IDF.

In contrast, we observe a low correlation of *BLEU* and *METEOR* to *Reg-base* forming a cluster of surface-level metrics.

Ablation study. Figure 1 displays performance development in Spearman's ρ of the regressive ensemble when we incrementally eliminate the metrics from the set of ensembled predictors. Following the methodology described in Section 3.5, the exact ordering of the metrics in ablation for ***zh-en*** pair is shown in Table 2, and we observe it to be similar also for the other language pair of MQM.

In ensembles of reference-based metrics (left), we observe a high consistency throughout the removal of most of the metrics. A longer consistency in ***zh-en*** case is attributed to a consistent performance in ensembling *BLEURt* (removed in step 14) and *METEOR* (removed in step 15). These metrics only reach the correlation of 0.48 and 0.28, respectively, when evaluated independently. In ***en-de*** case, the most significant drops can be attributed to a removal of the best-performing *Comet* (step 9) and *Prism* (step 11).

Ensemble of source-based metrics (right) shows significant drops in ***zh-en*** pair after removing *Prism* (step 3) and *WMD-contextual* (step 4). In ***en-de*** language pair, the correlation is relatively low throughout the whole ablation process. The least corre-

	0)	1)	2)	3)	4)	5)	6)	7)	8)	9)	10)	11)	12)	13)	14)	15)	16)	17)	18)
0) *MQM avg scr*	1.	.62	.51	.45	.45	.43	.27	.21	.1	.26	.19	.35	.29	.27	.48	.25	.28	.01	.35
1) *RegEMT*	.59	1.	.56	.5	.44	.42	.25	.23	.4	.28	.15	.36	.3	.3	.55	.26	.32	.11	.51
2) *Comet*			1.	.79	.82	.82	.71	.61	.54	.62	.57	.64	.64	.6	.83	.64	.63	.44	.4
3) *Prism*	.36	.5		1.	.9	.9	.85	.77	.63	.71	.71	.77	.79	.77	.8	.81	.77	.54	.19
4) *BERTScr*	.44	.48		.13	1.	.13	.06	.01	.11	.1	.13	.04	.76	.73	.82	.77	.74	.56	.18
5) *WMD-cont*	.44	.5		.53	.88	1.	.92	.8	.68	.69	.71	.74	.77	.73	.82	.81	.75	.58	.18
6) *WMD-dec*	.29	.42		.22	.4	.6	1.	.89	.81	.7	.77	.72	.77	.77	.75	.81	.71	.66	.37
7) *WMD-dec-tf*	.17	.3		.08	.12	.32	.69	1.	.71	.66	.73	.71	.74	.8	.63	.74	.64	.6	.32
8) *SCM-dec*	.19	.26		.03	.21	.35	.71	.49	1.	.51	.67	.49	.55	.58	.57	.61	.53	.65	.58
9) *SCM-dec-tf*	.13	.26		.2	.32	.43	.47	.26	.29	1.	.58	.72	.64	.61	.62	.62	.58	.48	.19
10) *SCM*											1.	.78	.9	.86	.54	.75	.74	.67	.37
11) *SCM-tf*												1.	.85	.84	.6	.71	.72	.52	.13
12) *WMD*													1.	.94	.6	.81	.83	.6	.17
13) *WMD-tf*														1.	.58	.78	.78	.58	.22
14) *BLEUrt*															1.	.6	.58	.43	.11
15) *BLEU*																1.	.85	.66	.27
16) *METEOR*																	1.	.56	.15
17) *Compos*	.13	.11		.13	.06	.01	.11	.1	.13	.04								1.	.52
18) *Reg-base*	.34	.44		.37	.07	.07	.24	.19	.28	.02								.34	1.

Table 2: Pairwise Spearman's correlations of the evaluated metrics and their correlations to averaged MQM judgements for ***zh-en*** language pair. Top-right triangle: mutual correlations of reference-based metrics, bottom-left triangle: correlations of metrics supporting multilingual source-based evaluation.

lated and hence the last ones eliminated are *WMD-decontextualized-tfidf* (step 6) and *Prism* (step 7).

5 Discussion

Regressive ensemble. Following the objectives that we set in Section 3, we empirically confirm that an ensemble can push the quality of modeling the expert judgements in most of the configurations while performing close-to-the-best metrics on the others. Additionally, we demonstrate that such ensemble is transferable to new language pairs and that its use is motivated by qualitative gains even in cross-lingual settings.

At the same time, one must acknowledge the limitations that an ensemble system exposes compared to single and unsupervised metrics. An ensemble might inherit the systematic biases of each of its metrics. This problem is observable in the results of the source-based ***en-de*** pair of MQM in Table 1, where the ensemble follows the low correlations of its ensembled metrics. Further, relying entirely on the metrics' consistency, the ensemble will inevitably expose errors in domains where some metrics behave markedly out of their usual range.

On the other hand, we argue that this might rarely be the case with the surface-level metrics that are mainly unsupervised. We suspect it to be unlikely with learnable metrics, too, having their output space constrained by the range of their imitated metrics.

Values of correlations in Table 2 and partially also the threshold metrics in Figure 1 suggest that our ensemble relies primarily on trained contextualized metrics with regards to their correlation with the target as summarized in Table 1. We suspect that oversampling of under-represented categories of errors would increase the significance of other types of metrics, as the under-represented error categories would be the ones where the fine-tuned metrics perform worse.

Baseline ensemble. Surprisingly, our baseline ensemble *Reg-base* consistently outperforms other standard surface-based metrics such as *BLEU* (Papineni et al., 2002). This suggests possible applicability of surface-level metrics also in reference-free evaluation.

We suspect that the baseline features of length based on a multilingual WordPiece tokenizer (Wu et al., 2016) might reflect on the missing or inappropriately added segments more *strictly* than other surface-level metrics. At the same time, these errors are usually highly weighted in the overall score.

Table 2 shows considerable orthogonality of *Reg-base* to other metrics. This motivates the inclusion

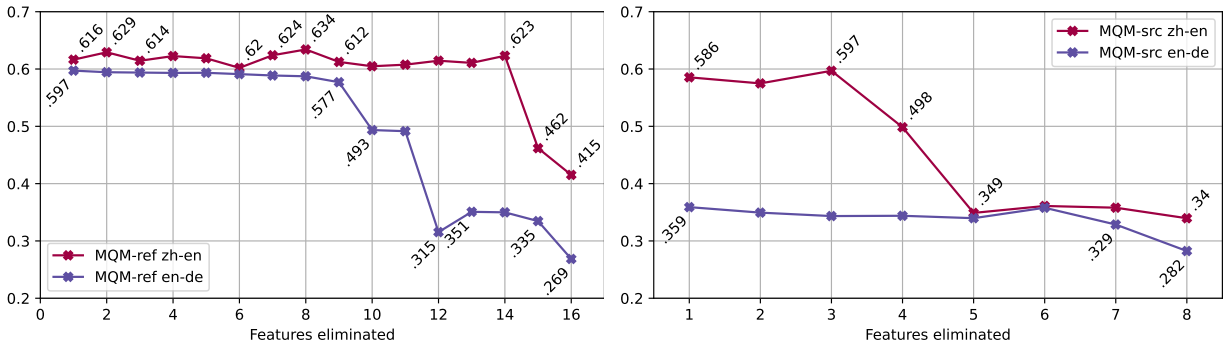


Figure 1: Ablation study: correlation of ensemble estimator to averaged MQM judgements during incremental elimination of the most-correlated metric from the ensembled predictors. Left: ablation of reference-based metrics, right: ablation of source-based metrics.

of the other weak surface metrics into the baseline ensemble to alleviate some of its apparent flaws.

Impact of contextualization. Based on the results of *WMD-** described in Section 3.2.2, one can not draw a consistent conclusion regarding the impact of contextualization. On average, decontextualization has decreased the performance of *WMD* by 8%, but the original motivation of a significant improvement in the usability of estimators might compensate. On the other hand, *WMD-decontextualized* and *WMD-decontextualized-tfidf* reached a considerable improvement of 16–18% as compared to *WMD* and *WMD-tfidf* using FastText embeddings, while losing none of their flexibility.

> "It is the harmony of the diverse parts, their symmetry, their happy balance; in a word it is all that introduces order, all that gives unity, that permits us to see clearly and to comprehend at once both the ensemble and the details."
>
> Henri Poincaré

6 Conclusion

This work evaluates the potential of ensembling multiple diverse metrics (*RegEMT*) for an evaluation of machine translation quality and offers a new simple baseline metric *Reg-base* that achieves better results than *BLEU* and *METEOR* by using just the source and reference lengths. We measure significant gains in Spearman's correlation to MQM with *RegEMT* compared to standalone metrics and we demonstrate that even simple linear estimators can benefit from the expressivity that the methods of all levels of representation provide. Additionally, as we demonstrate, the ensemble based on metrics supporting multilingualism can push the quality further even on unseen language pairs.

We recognize the inherent limitations of the regressive ensemble, which is inevitably slower, resource-heavier, and prone to inherit latent inductive biases of underlying metrics or their combinations. However, *RegEMT* shows the agility of the simple ensemble approach, which is in contrast to attempts to learn the full complexity of quality estimation through a single objective and allows the quality estimator to avoid the blind spots of particular metrics. We hope that our results will motivate future work in the ensemble evaluation.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, USA. ACL.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the ACL*, 5:135–146.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proc. of the workshop on Treebanks and Linguistic Theories*, pages 24–41.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805v2.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. MLQE-PE: A Multilingual Quality Estimation and Post-Editing Dataset. ArXiv:2010.04480.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020b. Unsupervised Quality Estimation for Neural Machine Translation. *Transactions of the ACL*, 8:539–555.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. ArXiv:2104.14478.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomáš Mikolov. 2018. Learning word vectors for 157 languages. ArXiv:1802.06893v2.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From Word Embeddings To Document Distances. In *Proc. of International Conference on Machine Learning*, volume 37, pages 957–966, Lille, France. PMLR.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in Neural Machine Translation. Accepted ICLR 2019 Conference Blind Submission.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proc. of the 58th Annual Meeting of the ACL*, pages 7871–7880.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. ACL.

Evgeny Matusov. 2019. The Challenges of Using Neural Machine Translation for Literature. In *Proc. of the Qualities of Literary Machine Translation*, pages 10–19, Dublin, Ireland. European Association for Machine Translation.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proc. of the 57th Annual Meeting of the ACL*, pages 3428–3448, Florence, Italy. ACL.

Vít Novotný. 2018. Implementation Notes for the Soft Cosine Measure. In *Proc. of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 1639–1642, New York, NY, USA. ACM.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting on ACL*, pages 311–318, USA. ACL.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. ACL.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proc. of the 58th Annual Meeting of the ACL*, pages 7881–7892.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Brian Thompson and Matt Post. 2020a. Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121. ACL.

Brian Thompson and Matt Post. 2020b. Paraphrase Generation as Zero-Shot Multilingual Translation: Disentangling Semantic Similarity from Lexical and Syntactic Diversity. In *Proc. of the 5th Conference on Machine Translation (Vol. 1)*, pages 561–570. ACL.

Kevin Unhammer and Trond Trosterud. 2009. Reuse of Free Resources in Machine Translation between Nynorsk and Bokmål. In *Proc. of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 35–42, Alicante. Universidad de Alicante.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Norouzi, Macherey, Krikun, Cao, Gao, Macherey, Klingner, Shah, Johnson, Liu, Kaiser, Gouws, Kato, Kudo, Kazawa, Stevens, Kurian, Patil, Wang, Young, Smith, Riesa, Rudnick, Vinyals, Corrado, Hughes, and Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Wikipedia Talk Labels: Personal Attacks.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. ArXiv:1904.09675v3.

Multilingual Machine Translation Evaluation Metrics Fine-tuned on Pseudo-Negative Examples for WMT 2021 Metrics Task

Kosuke Takahashi[1], **Yoichi Ishibashi**[1], **Katsuhito Sudoh**[1,2], **Satoshi Nakamura**[1]
[1] Nara Institute of Science and Technology
[2] PRESTO, Japan Science and Technology Agency
{takahashi.kosuke.th0, ishibashi.yoichi.ir3, sudoh, s-nakamura}@is.naist.jp

Abstract

This paper describes our submission to the WMT2021 shared metrics task. Our metric is operative to segment-level and system-level translations. Our belief toward a better metric is to detect a significant error that cannot be missed in the real practice cases of evaluation. For that reason, we used pseudo-negative examples in which attributes of some words are transferred to the reversed attribute words, and we build evaluation models to handle such serious mistakes of translations. We fine-tune a multilingual largely pre-trained model on the provided corpus of past years' metric task and fine-tune again further on the synthetic negative examples that are derived from the same fine-tune corpus. From the evaluation results of the WMT21's development corpus, fine-tuning on the pseudo-negatives using WMT15-17 and WMT18-20 metric corpus achieved a better Pearson's correlation score than the one fine-tuned without negative examples. Our submitted models are named C-SPEC (Cross-lingual Sentence Pair Embedding Concatenation) and C-SPECpn, are the plain model using WMT18-20 and the one additionally fine-tuned on negative samples, respectively.

1 Introduction

Recent studies of automatic evaluation is mostly based on the family models of BERT (Devlin et al., 2019). BERTscore (Zhang et al., 2020), BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020) have shown a strong correlation with human judgement scores. However, we reported in our previous study (Takahashi et al., 2020), it is hard for BERT based metrics to correctly evaluate the translation errors that are annotated with low Direct Assessment (DA) score.

Upon the problems of evaluating poor quality translations, Sudoh et al. (2021) has attempted to solve the problem by creating a different human annotation set and corpus. Compared to DA, their idea is to make a clear definition of critical translation errors and let models learn the critical errors that can cause a serious misunderstanding.

Following the idea, we used pseudo-negative examples to train the evaluation model. Since, empirically, the cases of the evaluation failure happens frequently with the nouns translation errors, we generated pseudo-negative sentences by transferring the attribute of nouns with Word Attribute Transfer (Ishibashi et al., 2020). This system is based on our previous work (Takahashi et al., 2020), with an extension with fine-tuning with the pseudo-negative examples.

2 Related Work

BERTscore (Zhang et al., 2020), BERT regressor (Shimanaka et al., 2019), BLEURT (Sellam et al., 2020), and COMET (Rei et al., 2020) are applications of BERT to the machine translation evaluation. BERTscore measures the similarity of reference and hypothesis translation by the cosine-similarity of the token embeddings for each token in the reference and hypothesis. It uses a pre-trained BERT model without fine-tuning on evaluation data. Instead, BERT regressor and BLEURT are fully parameterized and require a human annotated evaluation corpus to fine-tune the models. Both metrics have the same model architecture; a linear layer is attached on top of the BERT encoder. They encode a paired reference and hypothesis sentence with BERT and predict the human evaluation score. Additionally, BLEURT conducts warm-up training of BERT before fine-tuning on an evaluation corpus. The model architecture of our submission is similar to BERT regressor and BLEURT, but its uniqueness comes from using the synthetic negative data to fine-tune the models to evaluate poor translations better.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1049–1052

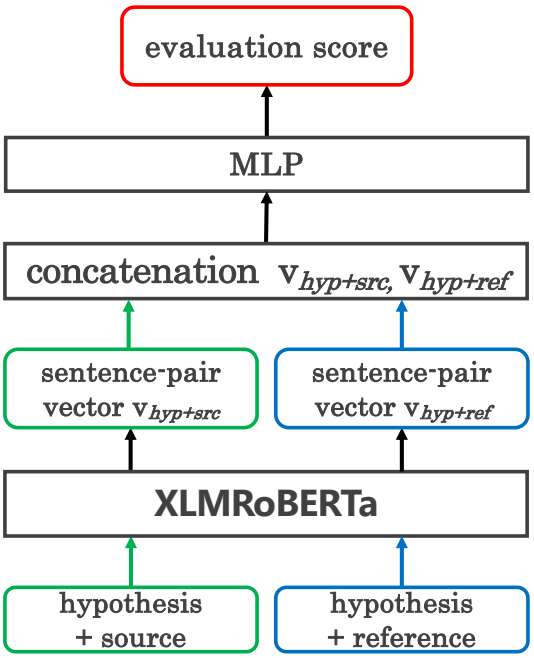


Figure 1: Architecture of C-SPEC

3 Our system

3.1 Model architecture

We extend the BERT regressor (Shimanaka et al., 2019) and use a cross-lingual language models, XLMRoBERTa (Conneau et al., 2020), to utilize a source sentence as a pseudo reference. In order to obtain a sentence-pair vector from source language and target language sentences together, our model encodes input sentences with a cross-lingual language model instead of monolingual BERT.

The model procedure is illustrated in the Figure 1. Our metric, called C-SPEC (Cross-lingual Sentence Pair Embedding Concatenation), uses paired inputs of hypothesis-source and hypothesis-reference. It introduces another vector for hypothesis-source ($v_{hyp+src}$) in addition to the standard one for hypothesis-reference pair ($v_{hyp+ref}$) to make an ensemble evaluation. Both sentence vectors are concatenated and used to predict the evaluation score in multi-layer perceptron (MLP). At first of the evaluation process, the cross-lingual language model encodes an input sentence into a sentence-pair vector. Then, using the sentence-pair vector, a MLP outputs the final evaluation score in regression manner. In training, we used standardized z score of DA (Direct Assessment; Graham et al. (2013)) as the ground truth and updated the model parameters by backpropagation (Rumelhart et al., 1986) with Mean Squared Error (MSE) Loss.

Our model was trained by the following steps. Firstly, in order to speed up and stabilize the training procedure, our models were trained on the corpus of WMT2015-2016's DA. Secondly, the models were additionally trained on WMT2015-2017's DA, WMT2018-2020's DA, or WMT2015-2020's DA. Thirdly, they were fine-tuned with the pseudo negative examples. Lastly, they were fine-tuned again on the WMT20's MQM segment-level corpus.

In each step of fine-tuning, we initialize the output-layer and only inherit the parameters of XLMRoBERTa. We tried three different conditions in the second step because DA corpus after WMT2018 is relatively noisy, and removing those data may play out well.

In the system-level evaluation, we simply averaged the segment-level evaluation scores for each system.

3.2 Word Attribute Transfer

We used the reflection-based word attribute transfer Ishibashi et al. (2020) for data augmentation. This transfer can make conversion of words into a certain word attribute, such as *queen* to *king*, using parameterized mirrors composed of two multi-layer perceptrons.

For the pseudo-negative hypothesis generation, we used two types of word attribute transfer in gender (male/female) and antonym. The word attribute transfer was applied onto all the words in an input sentence, and words having a target attribute were rewritten into their transferred counterparts while those that were not related to the target attribute were kept unchanged. For example, a sentence *"It is our duty to remain at his sides", he said, to applause* is transferred into *"It is our duty to change at his sides", he said, to whisper*, by the antonym transfer. Note that the word attribute transfer may not make any changes on an input sentence when all the words were identified as non-related words. We eliminated such sentences from our pseudo-negative examples.

3.3 Fine-tuning using pseudo-negative examples

Our pseudo-negative examples were obtained from the reference sentences in the training corpus of all-English that was used to firstly fine-tune a model, because the word attribute transfer model works only in English. However, we did not have any DA scores on these pseudo-negative examples. So, we used them to fine-tune the evaluation models

in classification manner. We introduced a different output layer on the top of the model illustrated in Figure 1 to classify an input example into the following categories:

1. A hypothesis is the same as its original system translation.
2. A hypothesis is the same as its reference.
3. A hypothesis is from the pseudo-negative examples

In the fine-tuning, we used three types of inputs corresponding to the classes above, and the models were trained to discriminate them. We expected such fine-tuned models to identify the serious word choice translation errors given in the pseudo-negative examples. We call the metric trained using the pseudo-negative examples C-SPECpn (pn:psuedo-negative).

4 Segment-level evaluation experiments

Our experiment was conducted on the development data for WMT21 metric task, which is randomly selected 10% of WMT20 MQM segement-level corpus. All the results were calculated by the Pearson's correlation with the MQM segment scores.

4.1 Results

The results of the WMT20 MQM segment-level corpus are shown in Table 1.

From the results, the models trained on negative examples of WMT15-17 and WMT18-20 overcame the plain models in Pearson's correlation. Among the models, the best score was archived by the one trained on WMT18-20 with the negative examples. Although WMT15-20 is a larger corpus than WMT18-20, the score of plain models was negligible at best, and the model trained on WMT18-20 and with negative examples did not overcome the plain one.

In order to figure out whether and how fine-tuning on the negative examples had impact on the evaluation performance, we calculated the Pearson's correlation for each small chunk of segment-level MQM scores and visualized the gap between models' outputs in Figure 2. Both the models trained on WMT15-17 and WMT18-20 with negative examples performed better in the MQM range of [-25.0, -5.0) and [-0.1, 0.0]. This suggests that using negative examples can improve the performance of evaluating high and critically low quality translations. However, the model trained on WMT15-20 with negative examples dropped its performance in the [-25.0, -5.0) range compared to the plain model. We assume the reason of the score drop is that the model was overly fine-tuned to the high quality translations, as it can be seen that the Pearson's correlation score in the [-0.1, 0.0] improved tremendously.

5 Conclusion

In this paper, we presented a BERT-based multilingual evaluation metric that is boosted by pseudo-negative examples to evaluate poor translations more precisely. Our model leverages our previous work Takahashi et al. (2020) and have shown an improvement of Pearson correlation when fine-tuning on the synthetic examples in the WMT15-17 and WMT18-20 corpus settings.

Acknowledgements

This work is supported by JST PRESTO (JPMJPR1856). With the support of JST, we used RIKEN's miniRAIDEN in a part of the experiments in this work.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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 and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yoichi Ishibashi, Katsuhito Sudoh, Koichiro Yoshino, and Satoshi Nakamura. 2020. Reflection-based word attribute transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*,

Table 1: Pearson's correlation with MQM segment scores in WMT2020. C-SPEC stands for a plain model fine-tuned without negative examples. C-SPECpn is a model fine-tuned on negative examples.

model	en-de	zh-en	avg	all
C-SPEC w/ WMT15-17	0.609	0.773	0.691	0.787
C-SPEC w/ WMT18-20	0.612	0.805	0.708	0.813
C-SPEC w/ WMT15-20	0.603	0.798	0.700	0.808
C-SPECpn w/ WMT15-17	**0.626**	0.809	0.717	0.817
C-SPECpn w/ WMT18-20	0.619	**0.824**	**0.721**	**0.829**
C-SPECpn w/ WMT15-20	0.309	0.715	0.512	0.724

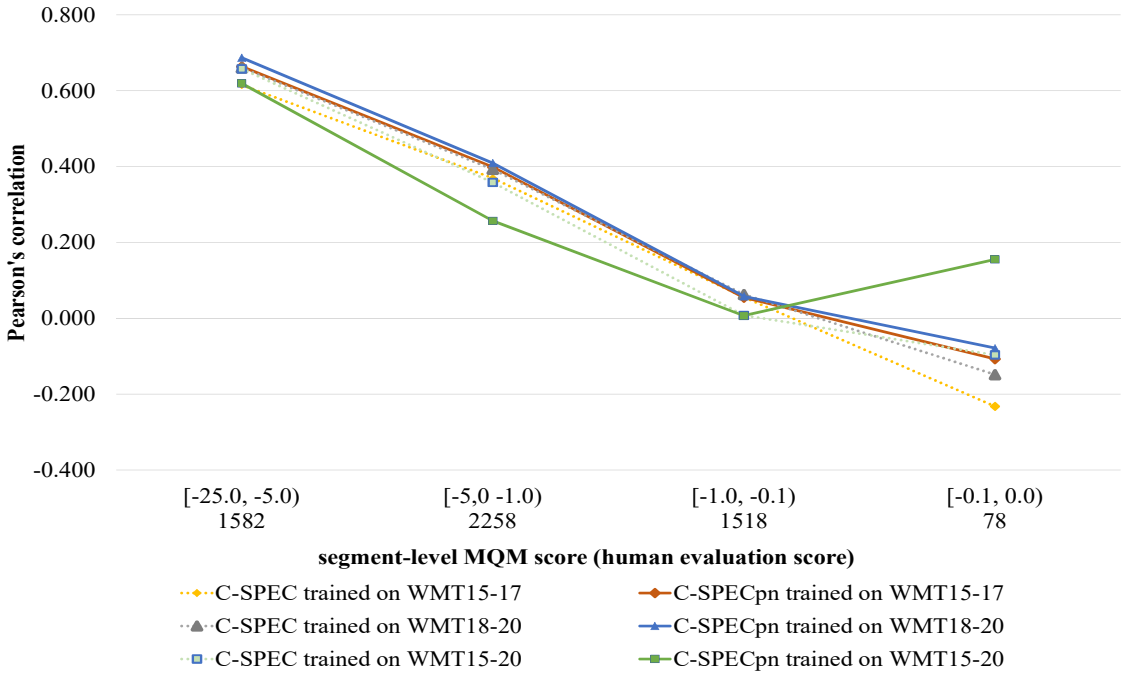


Figure 2: Pearson's correlation for each small segment-level MQM ranges. The amount of each segment is written below the range description.

pages 51–58, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

David Rumelhart, Geoffrey Hinton, and Ronald Williams. 1986. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine Translation Evaluation with BERT Regressor. *arXiv preprint*, abs/1907.12679.

Katsuhito Sudoh, Kosuke Takahashi, and Satoshi Nakamura. 2021. Is this translation error critical?: Classification-based human and automatic machine translation evaluation focusing on critical errors. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 46–55, Online. Association for Computational Linguistics.

Kosuke Takahashi, Katsuhito Sudoh, and Satoshi Nakamura. 2020. Automatic machine translation evaluation using source language inputs and cross-lingual language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3553–3558, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

RoBLEURT Submission for the WMT2021 Metrics Task

Yu Wan[1*] **Dayiheng Liu**[2†] **Baosong Yang**[2] **Tianchi Bi**[2] **Haibo Zhang**[2]
Boxing Chen[2] **Weihua Luo**[2] **Derek F. Wong**[1†] **Lidia S. Chao**[1]
[1]NLP[2]CT Lab, University of Macau
nlp2ct.ywan@gmail.com, {derekfw,lidiasc}@um.edu.mo
[2]DAMO Academy, Alibaba Group
{liudayiheng.ldyh,yangbaosong.ybs,tianchi.btc,
zhanhui.zhb,boxing.cbx,weihua.luowh}@alibaba-inc.com

Abstract

In this paper, we present our submission to Shared Metrics Task: **RoBLEURT** (**R**obustly **O**ptimizing the training of **BLEURT**). After investigating the recent advances of trainable metrics, we conclude several aspects of vital importance to obtain a well-performed metric model by: 1) jointly leveraging the advantages of source-included model and reference-only model, 2) continuously pre-training the model with massive synthetic data pairs, and 3) fine-tuning the model with data denoising strategy. Experimental results show that our model reaching state-of-the-art correlations with the WMT2020 human annotations upon 8 out of 10 to-English language pairs.

1 Introduction

Automatically evaluating the adequacy of machine translation (MT) candidates is crucial for judging the quality of MT systems. N-gram-based metrics, such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and chrF++ (Popovic, 2015, 2017), have dominated in the topic of MT metric. Despite the success, recent studies (Smith et al., 2016; Mathur et al., 2020a) also pointed out that, N-gram-based metrics often fail to robustly match paraphrases and capture distant dependencies. As MT systems become stronger in recent decades, these metrics show lower correlations with human judgements, leading the derived results unreliable.

One arising direction for metric task is using trainable model to evaluate the semantic consistency between candidates and golden references via predicting scores. BERTScore (Zhang et al., 2020), BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020) have shown higher correlations with human judgements than N-gram-based automatic metrics. Benefiting from the powerful pre-trained language models (LMs), e.g., BERT (Devlin et al., 2019), those fine-tuned metric models first derive the representation of each input, then introduce an extra linear regression module to give predicted score which describes to what degree the MT system output adequately expresses the semantic of source/reference contents. Furthermore, related work (Takahashi et al., 2020; Rei et al., 2020) reports that, metrics which additionally introduces source sentences into inputs can further boost the performance of metric model.

To push such "model as a metric" approach further, we present RoBLEURT – Robustly optimizing the training of BLEURT (Sellam et al., 2020), to achieve a better consistency between model predictions and human assessments. Specifically, for low-resource scenarios, using only hypotheses and references can give more accurate results, alleviating the sparsity of source-side language; for the high-resource language pairs, we format the model input as the combination of source, hypothesis and reference sentences, making model attending to both source input and target reference when evaluating the consistency of semantics. Then, we collect massive pseudo data from real MT engines tagged by pseudo scores with strong baselines for supervised model pre-training. As to the fine-tuning phase, we rescore the noisy WMT metric data of previous years with strong metric baselines, which are then utilized to fine-tune our model. Experimental results show that, following the setting of WMT2021 metric task, our RoBLEURT model outperforms the reported results of state-of-the-art metrics on multilingual-to-English language pairs.

2 RoBLEURT

2.1 Combining Multilingual and Monolingual Language Model

Same as previous years, translation tasks cover both low-resource and high-resource scenarios. To

[*]Work was done when Yu Wan was interning at DAMO Academy, Alibaba Group.
[†]Corresponding authors.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1053–1058

give higher reliable outputs, we believe our metric model can benefit from separately pre-trained and fine-tuned over each kind of scenarios:

- For low-resource multilingual-to-English language pairs, we can hardly obtain massive parallel data with high quality, nor access well-performed automatic translation systems to produce syntectic data for pre-training. We mainly consider model outputs and gold references as our model inputs. Thus we mainly consider the monolingual English language model (called RoBLEURT-NoSrc) in this scenario.

- As to high-resource language pairs, they do not suffer from limitations above, thus can benefit from the information of source input, model output and target reference. A multilingual version of pre-trained LM (called RoBLEURT-Src) can be used for this scenario.

The main architecture of our model is Transformer (Vaswani et al., 2017), which has been widely used in recent researches. As related studies point out that RoBERTa (Liu et al., 2019) outperforms conventional BERT (Devlin et al., 2019), we employ the well-trained model checkpoint from RoBERTa family. Besides, the conventional BLEURT model is trained based on uncased-BERT, which tokenizes the input sentences with the lower case format whereas RoBERTa uses case-sensitive tokenizer, which may be helpful to distinguish more information. Moreover, model with larger scale is generally related with better performance and higher capacity of available knowledges.

Recently, several approaches which further fine-tune RoBERTa model can give better performance over multiple natural language inference tasks. To make sure our model can also benefit from this, we finally use `RoBERTa-large-mnli`[1] and `RoBERTa-large-xnli`[2] (Conneau et al., 2020) for low-resouce and high-resource language pairs, respectively.

2.1.1 Model Combination

We are also interested in exploring whether we can boost the performance of combine RoBLEURT-NoSrc and RoBLEURT-Src. Combining the outputs from models trained with different settings is widely used in MT tasks (Barrault et al., 2020). In this paper, We simply use weighted combination of all available well-trained models.

2.1.2 Input Formatting

Our model consists of a well-trained RoBERTa model to obtain segment-level representations. Here we also try with two solutions: the model input includes source sentence (RoBLEURT-Src) or not (RoBLEURT-NoSrc). For the former, the model input is formatted as:

<s> hyp' </s> </s> ref </s>. (1)

As the latter, due to the number of input sentences is larger than RoBERTa predefined training format, we redesigned the input format as:

<s> src </s> </s> hyp' </s> </s> ref </s>. (2)

2.1.3 Prediction Module

To obtain a scalar value as predicted score, we directly derive the representation at the first position of input $\mathbf{X} \in \mathcal{R}^{1\times d}$ as the representation of input tuple, where d is the size of hidden states. It is then fed to projection layer, after which we yield a scalar for describing how adequately the hypothesis express the semantics:

$$s = \mathbf{W}\mathbf{X}^{\top} + b, \tag{3}$$

where $\mathbf{W} \in \mathcal{R}^{1\times d}, b \in \mathcal{R}^{1}$ are both trainable parameters.

During training, the learning objective is to reduce the mean squared error (MSE) between model prediction s and annotated score $score$:

$$\mathcal{L} = (s - score)^2. \tag{4}$$

2.2 Continuous Pre-training with Synthetic Data

Continuous Pre-training the model on synthetic data is proven helpful to improve the performance (Sellam et al., 2020), where BLEURT obtain the synthetic data by randomly perturbing 1.8 million segments from Wikipedia for this continuous pre-training (also called mid-training). However, we doubt that applying datasets out of MT domain, or even use learning signals tagged from non-reliable automatic metrics (e.g., BLEU), may harm the model learning during pre-training phase. As a consequence, we consider collecting synthetic data

[1] https://huggingface.co/roberta-large-mnli
[2] https://huggingface.co/joeddav/xlm-roberta-large-xnli

with real MT models over MT task datasets. To this end, we first collect the available translation outputs by using accessible engines[3] to generate MT hypotheses. Specifically, we collect high-quality cross-lingual parallel MT training data, including Czech (cs) / German (de) / Japanese (ja) / Russian (ru) / Chinese (zh) – English (en), from the WMT News translation track of each year. By taking the source side (cs/de/ja/ru/zh) as input for translation engines, we collect multiple triples formatted as (src, hyp, ref), where src, hyp, ref represent source, hypothesis, and reference respectively.

Adding Noise to Data As Sellam et al. (2020) demonstrated that, when collecting synthetic data for pre-training metric model, adding noise to data is helpful for model learning. Due to the high quality of automatically generated MT candidates in recent decades, such noise can smoothen the distribution of semantic consistency over whole dataset, which benefits the metric model learning. We thus follow their research, randomly select 30% of collected data to be added with noise at the hypothesis side. More specifically, we use the "word drop" noise – randomly dropping words with a randomized ratio for chosen sentence – to achieve such goal of quality reduction. Finally, we obtain a synthetic dataset formatted as (src, hyp', ref), where hyp' is the noisy hypothesis.

Data Pseudo Labeling As our model tends to be a regression model – predicting score for each inputted triplet, supervisedly guiding the model learning with given scores is essential. To give more adequate scores for each data item, we use COMET (Rei et al., 2020) for tagging each triplet, resulting into the data items formatting as quadruple $(src, hyp', ref, score)$. To make sure our model should be stably trained, we rescale the scores with Z-score format following Sellam et al. (2020).

2.3 Fine-tuning with Data Denoising Strategy

As reported in Sellam et al. (2020), Ma et al. (2019), and Mathur et al. (2020b), noisy data may give incorrect judgements on the reliability of one specific MT metric. After collecting the data from previous years, we find out that the DA datasets from year 2018-2019 are recognized as noisy ones, however they contribute a considerable portion to the available DA datasets. To give more accurate learning signals for training, we believe identifying the noisy data items is of vital importance. Specifically, we prepare the required metrics following two methods:

- RoBLEURT checkpoints. We first train several RoBLEURT models with different portions of training data, as well as multiple experiments by setting different random seeds. Here we use both RoBLEURT and RoBLEURT-NoSrc settings, and derives 4 checkpoints following each setting.
- Available well-performed checkpoints. We collect the officially released COMET[4] and BLEURT checkpoints[5].

After collecting the predictions with all checkpoints above, we identify the noisy data items by computing the variance of rankings within whole dataset. Finally, we rescore those noisy items with those models, tagging pseudo labels for fine-tuning. Besides, to guarantee the scores are unbiased, we re-normalize them within the dataset of each year by Z-score following Sellam et al. (2020).

3 Experiments

3.1 Settings of Continuous Pre-training

Synthetic Data Collection To continue pre-training the model, we simply collect parallel data from the previous WMT conferences, taking the training data from MT track cs/de/ja/ru/zh-en language pairs to obtain high-resource pseudo data. Finally, for each language pair, we collect 2.0 million quadruples for metric model pre-training. For low-resource scenarios, we reuse the datasets above, where the only difference is removing the source sentences.

As to development set, we directly collect the direct assessment (DA) dataset from the WMT2020 Metrics task track. We evaluate the model performance following DARR assessments (Ma et al., 2019; Rei et al., 2020), and choose the best checkpoint for fine-tuning.

Hyper-parameters During the continuous pre-training, we determine the maximum learning rate as $5 \cdot 10^{-6}$, training steps as 0.5M and warm-up steps as 50K. The learning rate first linearly warms up from 0 to maximum learning rate, then decays to

[3] We use own MT engines to obtain translation hypotheses.

[4] https://github.com/Unbabel/COMET

[5] https://github.com/google-research/bleurt

Model	High-Resource						Low-Resource			
	cs	de	ja	ru	zh	iu	km	pl	ps	ta
Baseline										
SENTBLEU	6.8	41.3	18.8	-0.5	9.3	18.2	22.6	-2.4	9.6	16.2
TER	-4.0	35.5	4.4	-11.7	-1.0	2.1	12.5	-17.2	-3.6	4.6
CHRF++	9.0	43.5	4.4	-11.7	-1.0	24.6	27.5	3.4	14.5	18.6
BLEURT (Sellam et al., 2020)	12.6	45.6	25.8	9.3	13.7	25.8	32.7	5.7	**20.7**	23.0
COMET (Rei et al., 2020)	12.9	48.5	27.4	15.6	17.1	28.1	29.8	9.9	15.8	24.1
SOTA Results (Mathur et al., 2020b)	14.3	48.5	27.7	15.6	17.1	28.1	**33.0**	10.9	**20.7**	25.3
Our method										
ROBLEURT	**15.2**	**49.3**	**29.1**	**17.3**	**17.7**	**29.0**	31.4	**13.2**	20.1	**25.4**

Table 1: DARR Kendall correlation (%) over WMT2020 data for each language pair (xx-en). Results of baseline systems are conducted from official report (Mathur et al., 2020b). Best viewed in bold.

0 till the end of training. To avoid over-fitting, we apply the dropout ratio as 0.1. We conduct the pre-training experiments with 8 Nvidia V100 GPUs, where each batch size for each GPU device contains 4 quadruplets. To avoid memory issues during pre-training, we simply reduce the number of total tokens, leaving 128 and 192 for ROBLEURT-NOSRC and ROBLEURT-SRC, respectively.

3.2 Settings of Fine-tuning

Data Collection We fine-tune our model with the WMT2015-2019 dataset as training set, where the WMT2018-2019 subsets are processed with our data denoising strategy as discussed in § 2.3. To directly confirm the effectiveness of our approach, we simply use WMT2020 dataset as dev set to compare reported results in WMT2020 metric task.

To select the model for participating the WMT2021 metric task, we divide the WMT2020 dataset into 4 folds, where the data items are firstly gathered with the identical source and reference sentence. For each fold, we select the corresponding fold of the WMT2020 subset as the dev set, and use the combination of the WMT2015-2019 dataset and the other unused WMT2020 subsets as the training set.

Hyper-parameters During fine-tuning, we set the training steps and warm-up steps as 20K and 2K, respectively. The other hyper-parameters are identical to those of pre-training phase. For each fine-tuning experiment, we determine the batch size as 16, and whole training process requires one single Nvidia V100 GPU.

Main Results We first testify the effectiveness of our approach by comparing with the results from the WMT2020 Metrics Task submissions. To be fairness, all of the model based metric baselines are trained on the WMT2015-2019 dataset. As shown in Table 1, comparing to baselines, our ROBLEURT achieves the best performance on cs/de/ja/ru/zh/iu/pl/ta-to-en settings, and achieves competitive results on km-to-en and ps-to-en.

4 Ablation Studies

4.1 Model Pedestal and Size

We first investigate the impact of model pedestal for metric task. As shown in Table 3, using RoBERTa-large instead of RoBERTa-base model as the base of ROBLEURT-SRC model gives a better performance. Furthermore, using the fine-tuned checkpoint RoBERTa-large-xnli can further improves the performance. This indicates our view, that powerful pre-trained LM, as well as the carefully re-optimized variants, can boost the performance of fine-tuned metric model.

4.2 Pre-training

To identify the improvement after introducing extra pre-training steps for metric model, we conduct the results in Table 4 for comparison. As seen, the performance drops significantly without pre-training phase. This caters to the previous findings (Sellam et al., 2020), where pre-training with pseudo data helps the supervised learning of metric model.

4.3 Data Denoising Strategy

As reported in (Sellam et al., 2020), the WMT2018-2019 DA subsets are bothered with noisy labels. We also investigate the impact of those data, whether introducing them into model training, or even clean them via rescoring with stronger metric. We thus arrange such ablation study during fine-tuning, and results are conducted in Table 5. Although the noisy portion contributes a great share

Model	cs	de	ja	ru	zh	iu	km	pl	ps	ta
RoBLEURT-NoSrc	13.5	46.9	27.4	10.8	14.8	28.2	30.6	8.3	14.7	25.0
RoBLEURT-Src	14.1	47.9	28.7	11.7	14.9	27.5	29.9	6.4	16.0	24.0
RoBLEURT	**15.2**	**49.3**	**29.1**	**17.3**	**17.7**	**29.0**	**31.4**	**13.2**	**20.1**	**25.4**

Table 2: DARR Kendall correlation (%) over WMT2020 data with model combination. For each setting, we present the averaged correlation with well-trained 3 models. Combining both RoBLEURT-Src and RoBLEURT-NoSrc models can give significant improvement.

Model	cs-en	de-en	ja-en	ru-en	zh-en
base	11.7	44.3	24.1	9.1	12.1
large	12.4	46.2	26.2	**12.0**	14.1
large-xnli	**14.1**	**47.9**	**28.7**	11.7	**14.9**

Table 3: DARR Kendall correlation (%) over WMT2020 data with different pedestals for RoBLEURT-Src setting. Larger model size can give better performance for metric model, and finetuned RoBERTa-large-xnli model can push the improvement further.

Model	cs-en	de-en	ja-en	ru-en	zh-en
w/o pretrain	10.6	44.8	21.4	6.1	10.2
w pretrain	**14.1**	**47.9**	**28.7**	**11.7**	**14.9**

Table 4: DARR Kendall correlation (%) over WMT2020 data with data filtering. We use RoBLEURT-Src model to conduct the results. Simply removing the noisy portion does not help the model training. However, reintroducing them into training set after rescoring them gives a significant improvement.

Model	cs-en	de-en	ja-en	ru-en	zh-en
full set	9.1	45.0	23.5	8.1	9.8
& remove	13.4	46.8	26.1	**11.7**	14.1
& rescoring	**14.1**	**47.9**	**28.7**	**11.7**	**14.9**

Table 5: DARR Kendall correlation (%) over WMT2020 data with data filtering. We use RoBLEURT-Src model to conduct the results. Simply removing the noisy portion does not help the model training. However, reintroducing them into training set after rescoring them gives a significant improvement.

of full training set (237K vs. 247K), the performance of RoBLEURT model trained without these noisy items does not diminish significantly. After rescoring with available checkpoints, these data segments further improves model performance.

4.4 Model Combination

We first identify whether introducing source side information to metric model helps training. As seen in Table 2, accepting source (row RoBLEURT-Src) than not (row RoBLEURT-NoSrc) as extra input significantly improves the correlation scores. However, for low-resource scenarios, experimental results show that source-side information does not help much for model training. This indicates that source information does not provide help for model training over low-resource scenarios, as the inadequacy of pre-training data may harms model training if source side is introduced. To derive better performance, one general idea is to combine several well-trained models during inference. We also explore whether combining both RoBLEURT-Src and RoBLEURT-NoSrc models can give better performance.

As shown in Table 2, directly averaging scores from multiple models lead to a significant performance drop. On the contrary, our model, which takes models over both RoBLEURT-NoSrc and RoBLEURT-Src settings can effectively leverage the predictions, achieving significant performance gain across all language pairs.

5 Conclusion

In this paper, we describe our submission metric – RoBLEURT, from the perspective of combining multilingual and monolingual language model, continuous pre-training with the massive synthetic data pairs, and fine-tuning with data denoising strategy. Experimental results confirms the effectiveness of our pipeline, demonstrating state-of-the-art correlations with the WMT2020 human annotations upon 8 out of 10 to-English language pairs.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (2018YFB1403202), the Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST).

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation WMT20. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL:HLT), Volume 1 (Long and Short Papers)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 Metrics Shared Task. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

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)*.

Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Maja Popovic. 2017. chrF++: words helping character n-grams. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Aaron Smith, Christian Hardmeier, and Joerg Tiedemann. 2016. Climbing Mont BLEU: The Strange World of Reachable High-BLEU Translations. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

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: Technical Papers (ATMA)*.

Kosuke Takahashi, Katsuhito Sudoh, and Satoshi Nakamura. 2020. Automatic Machine Translation Evaluation using Source Language Inputs and Cross-lingual Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (NIPS)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations (ICLR)*.

Linguistic evaluation for the 2021 state-of-the-art Machine Translation systems for German to English and English to German

Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova, Sebastian Möller
German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
firstname.lastname@dfki.de

Abstract

We are using a semi-automated test suite in order to provide a fine-grained linguistic evaluation for state-of-the-art machine translation systems. The evaluation includes 18 German to English and 18 English to German systems, submitted to the Translation Shared Task of the 2021 Conference on Machine Translation. Our submission adds up to the submissions of the previous years by creating and applying a wide-range test suite for English to German as a new language pair. The fine-grained evaluation allows spotting significant differences between systems that cannot be distinguished by the direct assessment of the human evaluation campaign. We find that most of the systems achieve good accuracies in the majority of linguistic phenomena but there are few phenomena with lower accuracy, such as the idioms, the modal pluperfect and the German resultative predicates. Two systems have significantly better test suite accuracy in macro-average in every language direction, Online-W and Facebook-AI for German to English and VolcTrans and Online-W for English to German. The systems show a steady improvement as compared to previous years.

1 Introduction

Evaluation in NLP and particularly in Machine Translation (MT) is an essential process for identifying flaws and leading further system improvements. Nevertheless, the exact method of evaluation to be used varies, given the quality requirements of the particular use case. Whereas the vast majority of the evaluation methods reside on metrics or direct assessment by humans to produce a single quality score given an entire test set, a recent trend has opted to evaluating the details of the produced translations, with major focus on their correctness from a linguistic perspective. For this reason, the translation systems are not tested based on generic test-sets, but they are given input which is particularly crafted to trial their performance. Most commonly, this is done with the help of a test suite (cf. Guillou and Hardmeier, 2016; Isabelle et al., 2017b; Burchardt et al., 2017).

The paper at hand describes the use of a test suite in order to evaluate 18 German to English and 18 English to German MT systems that participated at the Shared Task of the Sixth Conference on Machine Translation (WMT21)[1]. The evaluation is performed by an extensive test suite that tests a wide range of linguistically motivated phenomena. In addition to our contributions in the previous years, which focused only on German to English, this year we are presenting for the first time results with an extensive test suite with a similar logic for the opposite direction English to German. Our German to English test set contains 5,560 test sentences, covering 107 linguistic phenomena that are organized in 14 categories. The English to German test set contains 4,443 test sentences, covering 111 linguistic phenomena that are organized in 12 categories.

2 Related Work

Test suites have already been used since the beginnings of MT in the 1990s (King and Falkedal, 1990; Way, 1991; Heid and Hildenbrand, 1991). With the rise of deep learning, the quality of MT outputs has improved significantly, which in turn lead to a recent revival of test suites that focus on the evaluation of specific linguistic phenomena (e.g., pronoun translation (Guillou and Hardmeier, 2016), or on the comparison of different MT technologies (Isabelle et al., 2017a; Burchardt et al., 2017), and Quality Estimation methods (Avramidis et al., 2018).

Within the scope of the test suite track of the Conference on Machine Translation, several test suites for multiple language directions have

[1] http://statmt.org/wmt21/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1059–1073
November 10–11, 2021.

Lexical Ambiguity	
Er las gerne Novellen.	
He liked to read novels.	fail
He liked to read novellas.	pass
Phrasal verb	
Warum starben die Dinosaurier aus?	
Why did the dinosaurs die?	fail
Why did the dinosaurs die out?	pass
Why did the dinosaurs become extinct?	pass
Ditransitive Perfect	
Ich habe Tim einen Kuchen gebacken.	
I have baked a cake.	fail
I baked Tim a cake.	pass

Table 1: Examples of passing and failing MT outputs

been introduced. These test suites focus on one or multiple different phenomena, such as conjunctions (Popović, 2019), grammatical contrasts (Cinkova and Bojar, 2018), discourse (Bojar et al., 2018; Rysová et al., 2019), domain-specific translations (Vojtěchová et al., 2019), gender coreference (Kocmi et al., 2020), markables (Zouhar et al., 2020), morphology (Burlot et al., 2018), pronouns (Guillou et al., 2018), or word sense disambiguation (Rios et al., 2018; Raganato et al., 2019; Scherrer et al., 2020). In contrast to the majority of these test suites, our test suite does not focus on a single phenomenon but performs a systematic evaluation of more than one hundred phenomena per language direction.

3 Methods

Our test suite consists of two test sets (one per language direction) that have been created manually with the aim of testing the performance of MT systems. They cover a wide variety of linguistic phenomena which are grouped in different categories. While there is a big overlap between the linguistic categories and phenomena in the two test sets, there are also many differences as the categories and phenomena are language-specific. Some exemplary test sentences can be seen in Table 1.[2] Each linguistic phenomenon in the test suite is represented by multiple test sentences. Each test sentence is tied to a number of rules that determine whether a translation of the sentence would be deemed correct or incorrect. The performance of an MT system with regard to the linguistic phenomena is then evaluated by observing the amount of test sentences that are translated correctly.

[2]A larger set of exemplary test sentences can be found in the GitHub repository: https://github.com/DFKI-NLP/TQ_AutoTest.

3.1 Application of the test suite

The construction of the test suite has been described in detail in the papers for the test suite track from the previous years. Figure 1 depicts the preparation and application of the test suite with steps *a* to *c* representing the construction. The application starts with step *d*: The test sentences are given as input to the MT systems. The MT outputs are then evaluated by the set of rules which define whether the phenomenon under inspection is translated correctly or not (step *e*). The rules consist of regular expressions and fixed strings. When the rules cannot be applied to a translation to automatically determine whether it is correct or incorrect, the test sentence is marked with a warning. Those warnings are consequently inspected manually by a human annotator with linguistic knowledge who decides on the correctness of the translation and adapts the set of rules accordingly (step *e*).

Thereafter, the phenomenon-specific translation accuracy is calculated by dividing the number of correctly translated test sentences of a phenomenon by the total number of test sentences of that phenomenon:

$$\text{accuracy} = \frac{\text{correct translations}}{\text{sum of test items}}$$

Since the aim of this evaluation is to compare the systems in a fair way, we include only the test items that do not contain any warnings for any of the systems in the calculation. Test items that have an unresolved warning for at least one system are excluded from the calculation. Unfortunately, this reduces the amount of the test items by removing properly validated ones, and this is where we see the importance of the extensive manual evaluation and the creation of rules with good coverage.

To define which systems perform better for a particular phenomenon (or category), we compare all systems to the one with the highest accuracy. When we compare the highest scoring system with the rest, we confirm the significance of the comparison with a one-tailed Z-test with $\alpha = 0.95$. The systems which do not differ significantly from the best system are considered to be in the first performance cluster and indicated with boldface in the tables. The boldfaces therefore have a meaning only for the respective row of the table.

The average scores are computed in three different ways, because each category or phenomenon has a different amount of test items. *Micro-average* aggregates the contributions of all test items to

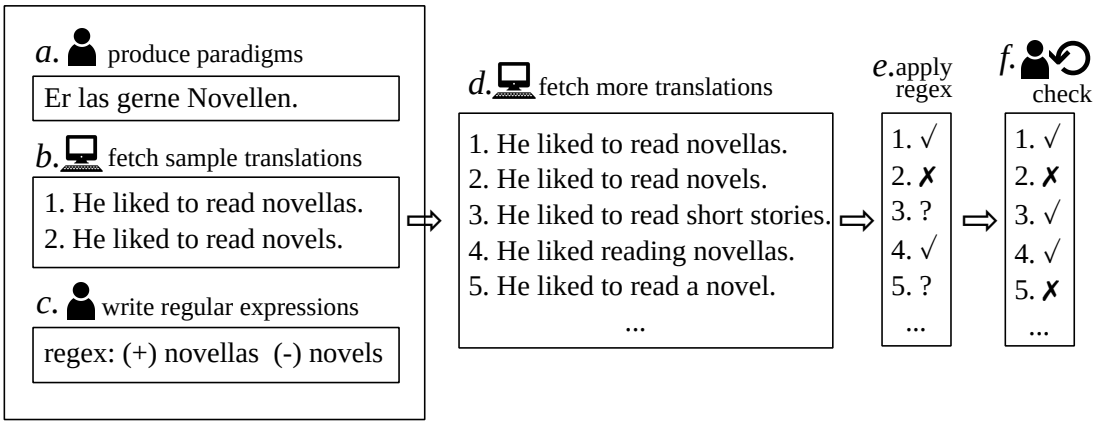


Figure 1: Example of the preparation and application of the test suite for one test sentence

compute the average percentages, *category macro-average* computes the percentages independently for each category and then averages them (i.e. treating all categories equally), and *phenomenon macro-average* computes the percentages independently for each phenomenon and then takes the average (i.e. treating all phenomena equally).

3.2 Experiment setup

In the evaluation presented in this paper, we obtained translations of our test suite by 36 systems that are part of the *news translation task* of the Sixth Conference on Machine Translation (WMT21). In previous years, we solely applied our test suite to the German to English MT outputs. However, this year, we did not only analyse the MT outputs from 18 German to English systems, but also from 18 English to German systems.

While there were already many rules for the evaluation of German to English MT output in our test suite, very few rules were available for the other language direction when we received the translations. Therefore, a significantly bigger amount of manual work was involved in the evaluation this year. For German to English there were on average 5.76% of warnings when we received the translations, while for English to German there were on average 84.21% of warnings. The manual evaluation process was conducted by three annotators with linguistic knowledge over the course of seven weeks and involved around 80 person hours. After the extensive manual evaluation, there were on average 3.04% of warnings for German to English and 4.87% for English to German.

As we explained previously, in order to have a fair comparison between the systems we excluded items where at least one system has an unresolved warning. Therefore, in the results that we are presenting in this paper we can only use 3,806 out of the 5,560 (68.4%) test items for German to English and 3,096 out of the 4,443 (69.7%) test items for English to German for the systems comparison.

4 Results

The accuracies resulting from the application of the test suite on the system outputs can be seen in the tables in the Appendix. We first present the results aggregated in categories (Tables 4 and 5) so one can have a broad overview of the systems performance, whereas afterwards a yearly comparison with last years (Table 6) and the detailed phenomenon-level results (Tables 7 and 8) are shown. The systems are ordered based on their macro-average accuracy, from high to low.[3]

4.1 Comparison between systems

For German to English, two systems have the highest category macro-averaged accuracy, Online-W and FacebookAI, whereas when considering the phenomenon macro-averaged accuracy, the significantly best systems are FacebookAI and Online-A. UEdin, Online-A and borderline compete with the best systems when the micro-average is considered, mainly because of their good accuracies on phenomena related to *verb tense/aspect/mood*, where there are many individual phenomena with a lot of test items in one category. Overall, the average accuracies are very high, with the lowest system (happyface) having a micro-average of 72.3%. Despite the high accuracies there is definitely room

[3]For German-English the two VolcTrans system variations appear as one system, since they delivered the same output. This is not the case for the English-German direction where they appear separately.

for improvement.

For English to German, based on the category macro-average, FacebookAI and VolcTransAT share the first position. Based on the micro-average and the phenomenon macro-average however, FacebookAI, Online-B and VolcTrans-GLAT share the first position. The accuracies for this language direction in overall are much higher on the micro-average, but not on the macro-average. However, due to the fact that the test items are different in their nature and in the amount, we cannot make a direct comparison between the two language directions.

4.2 Categories

For some categories, the accuracies have reached very high numbers, which is the case for *negation* and *punctuation*, both having a 100% for German to English. Concerning punctuation, in the previous years we had seen individual systems with considerable punctuation errors, which seem to not appear this year. However, the high scores do not necessarily mean that all problems for these phenomena are solved. It could rather mean that our test suite does not cover the current edge cases, a consideration that is subject to further research. Other categories such as *composition*, *subordination* and *named entities & terminology* reach an average of more than 90% accuracy in German to English. The worst performing category in German to English is *false friends*, where all systems perform 64-86%. *Ambiguity*, *verb tense/aspect/mood* and *multi-word expressions* (MWE) also perform relatively low, with accuracies less than 85%.

For English to German, there are no categories for which all systems reach an accuracy of 100%. However, there are several categories with average accuracies above 95%, that is *function words*, *negation*, *verb tense/aspect/mood*, and *subordination*. The category with the lowest average is *coordination & ellipsis*, with an average accuracy of only 70.8%. The individual systems reach a wide range of 58.6% to 81.6% accuracy for this category while for most other systems, the range is not as big for the systems. There are two more categories with a relatively low accuracy on average (below 85%), namely *verb valency* (81.4 % accuracy) and *ambiguity* (83.3% accuracy).

4.3 Phenomena

For **German to English**, the most difficult phenomena this year remain the *modal pluperfect* (*negated* and *non-negated*), the *resultative predicates* and the *idioms*. Online-W does impressively well with *idioms*, achieving almost 60%, with another two systems, FacebookAI and Online-A, reaching 33.3%. These numbers were significantly lower in the previous years, which indicates an improvement in this direction. There are some phenomena for which all systems reached 100% accuracy, such as *negation*, *internal possessor*, *comma*, *ditransitive perfect*, and *intransitive future I*.

Idiom	
Er redet um den heißen Brei herum.	
He's talking around the hot porridge.	fail
He's talking around the bush.	fail
He's beating around the bush.	pass
Modal pluperfect	
Sie hatten lesen wollen.	
They wanted to read.	fail
They had to read.	fail
They had wanted to read.	pass
Resultative predicate	
Lisa fuhr das Auto kaputt.	
Lisa drove the car broken.	fail
Lisa broke the car.	pass
Lisa crashed the car.	pass

Table 2: Examples of De-En linguistic phenomena with low accuracy with passing and failing MT outputs

Table 2 contains translation examples from linguistic phenomena with the lowest accuracy for German to English. *Idioms* are types of multiword expressions. The meaning of an idiom goes beyond the meanings of its individual elements. Most idioms are very language-specific and therefore difficult to translate. For the German idiom "um den heißen Brei herumreden", there is the equivalent English idiom "to beat about the bush". The first incorrect translation contains a direct translation of all the individual elements of the German idiom. The second incorrect translation, which was produced by several MT systems, is very interesting because it does indeed contain the "bush" of the English idiom. However, it still contains the wrong verb as the verb "is talking" is simply a translation of the German "redet". Therefore, the second translation is still incorrect. Only the third translation which contains the full English idiom is correct.

The second example contains a test sentence from the phenomenon *modal pluperfect*. Modal verbs can usually have several meanings which often leads to translation errors. Furthermore, the tense pluperfect is often mistranslated as preterite, as in the first incorrect translation. The second in-

Idiom	
The mafia boss has spilled the beans.	
Der Mafiaboss hat die Bohnen verschüttet.	fail
Der Mafiaboss hat sich verplappert.	pass
Der Mafiaboss hat es ausgeplaudert.	pass
Pseudogapping	
Jackie likes the doctor but she doesn't the nurse.	
Jackie mag den Arzt, aber sie nicht die Krankenschwester.	fail
Jackie mag den Arzt, aber sie ist nicht die Krankenschwester.	fail
Jackie mag den Arzt, aber nicht die Krankenschwester.	pass
Middle Voice	
This car drives easily.	
Dieses Auto fährt leicht.	fail
Dieses Auto fährt sich leicht.	pass
Das Auto ist leicht zu fahren.	pass

Table 3: Examples of En-De linguistic phenomena with low accuracy with passing and failing MT outputs

correct translation additionally leaves out the German modal verb "wollen" ("to want") which completely changes the meaning of the translation.

Resultative predicates contain a verb and an adjective which describes the result of the verb action. *Resultative predicates* do not exist that way in English, which makes them hard to translate. In the example at hand, the meaning of the German sentence is that Lena drove the car which resulted in the car being broken. A literal translation like in the first translation is ungrammatical. The second and third translation are possible correct translations – even though the "driving" part is left out, these translations are still deemed best options to translate this phenomenon.

In **English to German**, *idioms* show even more difficulties as in German to English (average accuracy only 14.6%, the lowest average accuracy on any phenomenon for this language direction). Here, 9 systems totally fail to translate any idiom, whereas the system with the highest accuracy is an unconstrained system, which may attributed to the fact that additional data led to better coverage of such cases. Furthermore, *middle voice* (45.9%), *pseudogapping* (60.5%), and *stripping* (57.0%) and also have a relatively low accuracy. On the other hand, there were also many phenomena which reached (nearly) 100% accuracy, such as *internal possessor*, *comma*, *indirect speech*, *infinitive clause*, *object clause*, *subject clause*, *passive voice*, and *ditransitive*, *intransitive* and *transitive verbs* in many tenses.

Table 3 covers example translation from low accuracy phenomena for English to German. The first example again contains an *idiom*. The English idiom "to spill the beans" does not have an equivalent idiomatic translation in German. Therefore, the first translation, which is a literal translation of the separate idiom elements, is incorrect. The second and third translation are possible correct translations.

The second example sentence is taken from the phenomenon *pseudogapping*. Put simply, in *pseudogapping*, part of the verb phrase is omitted. In the example at hand, the non-finite verb part "like" is omitted in the second conjunct of the construction. In the first incorrect German translation, the verb has been completely left out in the second conjunct (while the subject "sie" persists). In the second incorrect translation, the second conjunct contains the auxiliary verb 'ist" which also leads to ungrammaticality. The third translation leaves out the non-finite verb part "like" as well as the subject which results in a grammatical German construction.

The third example contains a sentence in *middle voice*. In middle voice, the subject of the verb is neither agent nor patient. A sentence in active voice would be: "I am driving the car.", with the subject ("I") being the agent. A sentence in passive voice would be: "The car is driven by me." with the subject ("the car") being the patient. The subject of the example sentence in Table 3 ("This car") is neither agent nor patient. As middle voice does not exist in German, such sentences have to be translated in other constructions. A literal translation like the first example translation is incorrect. Possible correct translations can be seen in the second and third translation.

5 Comparison with previous years

The progress of the systems performance through the last four years for German-English can be seen in Table 6. The calculation is done based on the common test items without warnings over all these years (4,366 test items), this is why the scores differ slightly from the ones in Table 4. In the first columns of Table 6 the best systems of every year are compared. One can see that the best system of 2021 has significantly better macro-averaged accuracy as compared to the best system of 2020, but when the micro-averaged accuracy is considered, there has been no significant improvement or deterioration. This year's best system also seems

to perform better in a few categories, with most impressive improvements at *false friends* (+14%) and the *non-verbal agreement* (+5%).

Individual systems show some small improvements in general, but the fine-grained evaluation is able to indicate some significant deterioration in particular categories. For example, Online-B, Online-G and VolcTrans, despite their overall improvement, show a significant deterioration regarding *verb tense/aspect/mood*, which reaches a -9% in the case of VolcTrans. Other deteriorations occur for several systems regarding *false friends* and *function words*. This shows that the overall improvement in translation quality may occur at the expense of particular qualitative aspects.

6 Conclusions and Further Work

We presented the result of applying a fine-grained linguistically motivated test suite on the outputs of 36 state-of-the-art machine translation systems, as submitted in the Sixth Conference on Machine Translation. We presented detailed accuracies of translations of 18 German to English as well as 18 English to German MT systems based on more than 3,000 test items each, organized in various linguistic categories and fine-grained phenomena. Additionally, we drew a comparison to previous years' evaluations.

In both language directions, the systems achieve good accuracies in most phenomena or categories and there is some advancement as compared to last year, although there is space for about 10% improvement on the average accuracy. A few phenomena still suffer considerably, such as the *idioms*, the *modal pluperfect* and the German *resultative predicates*, although there is notable improvement as compared to previous years.

As discussed, the very high accuracies for some categories or phenomena raise the question whether the difficulty of the respective test items should be increased. In future work, we plan to investigate this by constructing more test items. Further work includes the development of similar test suites for other language pairs.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) through the project TextQ, and by the German Federal Ministry of Education through the project SocialWear. We thank our colleague Tatjana Zeen for her valuable help 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 AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 243–248, Boston, MA. Association for Machine Translation in the Americas.

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*, pages 545–549, Belgium, Brussels. Association for Computational Linguistics.

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(1):159–170.

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*, pages 550–564, Belgium, Brussels. Association for Computational Linguistics.

Silvie Cinkova and Ondřej Bojar. 2018. Testsuite on Czech–English Grammatical Contrasts. In *Proceedings of the Third Conference on Machine Translation*, pages 565–575, Belgium, Brussels. 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'16)*, pages 636–643, Portorož, Slovenia. European Language Resources Association (ELRA).

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A Pronoun Test Suite Evaluation of the English–German MT Systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, pages 576–583, Belgium, Brussels. Association for Computational Linguistics.

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. 2017a. 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.

Pierre Isabelle, Colin Cherry, and George Foster. 2017b. A Challenge Set Approach to Evaluating Machine Translation.

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.

Tom Kocmi, Tomasz Limisiewicz, and Gabriel Stanovsky. 2020. Gender coreference and bias evaluation at wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 357–364, Online. Association for Computational Linguistics.

Maja Popović. 2019. Evaluating conjunction disambiguation on english-to-german and french-to-german wmt 2019 translation hypotheses. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 464–469, Florence, Italy. Association for Computational Linguistics.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The mucow test suite at wmt 2019: Automatically harvested multilingual contrastive word sense disambiguation test sets for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480, Florence, Italy. Association for Computational Linguistics.

Annette Rios, Mathias Müller, and Rico Sennrich. 2018. The Word Sense Disambiguation Test Suite at WMT18. In *Proceedings of the Third Conference on Machine Translation*, pages 594–602, Belgium, Brussels. Association for Computational Linguistics.

Kateřina Rysová, Magdaléna Rysová, Tomáš Musil, Lucie Poláková, and Ondřej Bojar. 2019. A test suite and manual evaluation of document-level nmt at wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 455–463, Florence, Italy. Association for Computational Linguistics.

Yves Scherrer, Alessandro Raganato, and Jörg Tiedemann. 2020. The mucow word sense disambiguation test suite at wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 365–370, Online. Association for Computational Linguistics.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. Sao wmt19 test suite: Machine translation of audit reports. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 481–493, Florence, Italy. Association for Computational Linguistics.

Andrew Way. 1991. Developer-Oriented Evaluation of MT Systems. In *Proceedings of the Evaluators' Forum*, pages 237–244, Les Rasses, Vaud, Switzerland. ISSCO.

Vilém Zouhar, Tereza Vojtěchová, and Ondřej Bojar. 2020. Wmt20 document-level markable error exploration. In *Proceedings of the Fifth Conference on Machine Translation*, pages 371–380, Online. Association for Computational Linguistics.

Appendix

category	count	Onl-W	Faceb	Onl-B	VolcT	Onl-A	SMU	Onl-G	Huawe	borde	Nemo	uedin	Water	P3AI	ICL	Onl-Y	Manif	happy	avg
Ambiguity	74	**87.8**	**90.5**	**86.5**	**86.5**	**81.1**	**83.8**	**85.1**	**89.2**	**83.8**	**83.8**	**83.8**	75.7	79.7	**86.5**	**82.4**	**81.1**	60.8	82.8
Composition	43	**97.7**	**97.7**	**100.0**	**100.0**	**97.7**	**95.3**	**97.7**	**95.3**	**95.3**	93.0	**97.7**	**97.7**	**97.7**	**95.3**	93.0	93.0	74.4	95.2
Coordination & ellipsis	57	89.5	89.5	89.5	89.5	87.7	86.0	86.0	87.7	89.5	87.7	87.7	86.0	87.7	77.2	87.7	89.5	80.7	87.0
False friends	36	**86.1**	**80.6**	**75.0**	**75.0**	**83.3**	**83.3**	**80.6**	63.9	**77.8**	**72.2**	66.7	**80.6**	**80.6**	**72.2**	**75.0**	**69.4**	63.9	75.7
Function word	40	**92.5**	**92.5**	**92.5**	**92.5**	**90.0**	**85.0**	**95.0**	**92.5**	**85.0**	**92.5**	**92.5**	**92.5**	**85.0**	**87.5**	**90.0**	72.5	80.0	88.8
LDD & interrogatives	103	**91.3**	**91.3**	**91.3**	**91.3**	**91.3**	**91.3**	**90.3**	**93.2**	**90.3**	**91.3**	**89.3**	**92.2**	**89.3**	**91.3**	**88.3**	57.3	74.8	87.9
MWE	66	**90.9**	**86.4**	**83.3**	**83.3**	**86.4**	**86.4**	**84.8**	**86.4**	**86.4**	**86.4**	**83.3**	**80.3**	**84.8**	**86.4**	**81.8**	**84.8**	69.7	84.2
Named entity & terminology	71	**95.8**	**94.4**	**93.0**	**93.0**	**94.4**	**93.0**	**94.4**	**95.8**	**91.5**	**91.5**	**95.8**	**88.7**	**90.1**	**91.5**	**93.0**	**90.1**	83.1	92.3
Negation	14	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Non-verbal agreement	57	**98.2**	**94.7**	**98.2**	**98.2**	**93.0**	**91.2**	89.5	**93.0**	**93.0**	**93.0**	89.5	89.5	**91.2**	**93.0**	84.2	**93.0**	73.7	91.5
Punctuation	18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Subordination	115	**92.2**	**93.9**	**95.7**	**95.7**	**93.9**	**92.2**	**93.0**	**92.2**	**92.2**	**93.9**	**93.9**	**94.8**	**93.0**	**93.9**	**93.9**	**93.9**	87.0	93.2
Verb tense/aspect/mood	3058	**87.3**	**87.3**	79.6	79.6	**86.4**	**85.8**	80.5	82.7	**86.5**	83.9	**86.9**	84.1	81.3	82.6	77.7	84.1	71.1	82.8
Verb valency	54	88.9	90.7	92.6	92.6	90.7	90.7	87.0	90.7	90.7	90.7	88.9	88.9	88.9	90.7	85.2	90.7	81.5	89.4
micro-average	3806	**88.3**	**88.2**	82.0	81.9	**87.3**	86.6	82.4	84.3	**87.1**	85.1	**87.4**	85.0	82.8	83.9	79.7	84.0	72.3	84.0
macro-average	3806	**92.7**	**92.1**	91.2	91.2	91.1	90.3	90.3	90.2	90.1	90.0	89.7	89.3	89.2	89.2	88.0	85.7	78.6	89.4

Table 4: Accuracies (%) of successful translations on a category level for German-English. Boldface indicates the significantly best performing systems in each row.

categ	count	Faceb	VolcA	Onl-W	Onl-A	Huawe	Nemo	Onl-B	VolcG	uedin	P3AI	eTran	happy	nucle	Onl-Y	Manif	BUPT	ICL	Onl-G	avg
Ambiguity	23	**91.3**	**95.7**	**95.7**	**91.3**	**87.0**	**87.0**	**91.3**	**91.3**	**82.6**	**82.6**	**78.3**	73.9	73.9	69.6	**82.6**	69.6	**82.6**	73.9	83.3
Coordination & ellipsis	87	**81.6**	**71.3**	**71.3**	**73.6**	**77.0**	**75.9**	**80.5**	**79.3**	69.0	69.0	66.7	64.4	65.5	**71.3**	63.2	63.2	58.6	**72.4**	70.8
False friends	38	92.1	92.1	89.5	86.8	86.8	84.2	84.2	84.2	86.8	86.8	86.8	86.8	81.6	86.8	86.8	86.8	84.2	84.2	86.5
Function word	35	**97.1**	**97.1**	**100.0**	**97.1**	**97.1**	**94.3**	**100.0**	**100.0**	**100.0**	**97.1**	**94.3**	**97.1**	**97.1**	**97.1**	65.7	**97.1**	**100.0**	**97.1**	95.9
MWE	98	**89.8**	**93.9**	**91.8**	85.7	**87.8**	**88.8**	**90.8**	**90.8**	82.7	85.7	84.7	**89.8**	82.7	82.7	83.7	81.6	80.6	81.6	86.4
Named entity & terminology	82	**93.9**	**97.6**	**93.9**	**93.9**	**93.9**	89.0	**93.9**	**93.9**	**92.7**	89.0	**93.9**	90.2	90.2	**92.7**	89.0	**92.7**	81.7	80.5	91.3
Negation	15	100.0	100.0	100.0	93.3	93.3	100.0	93.3	93.3	100.0	100.0	93.3	93.3	86.7	100.0	100.0	93.3	93.3	93.3	95.9
Non-verbal agreement	68	**100.0**	**98.5**	**97.1**	95.6	95.6	92.6	92.6	92.6	92.6	89.7	91.2	92.6	88.2	89.7	92.6	88.2	88.2	89.7	92.6
Punctuation	37	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	78.4	78.4	91.9	81.1	78.4	81.1	86.5	75.7	78.4	78.4	78.4	70.3	86.5
Subordination	161	**99.4**	**98.1**	**98.1**	**99.4**	95.7	**99.4**	**98.1**	**98.1**	**98.1**	**98.8**	**98.1**	**96.9**	**97.5**	93.8	**96.9**	94.4	92.5	96.3	97.2
Verb tense/aspect/mood	2366	98.6	97.9	97.3	96.9	96.1	97.4	**99.0**	**99.1**	**99.2**	97.4	98.4	96.7	97.3	90.7	98.6	94.8	95.2	94.7	97.0
Verb valency	96	**90.6**	81.3	**85.4**	81.3	**84.4**	81.3	**83.3**	**83.3**	81.3	**83.3**	**84.4**	80.2	80.2	77.1	81.3	77.1	75.0	74.0	81.4
micro-average	3106	**97.4**	96.5	95.9	95.3	94.7	95.6	**96.9**	**96.9**	96.5	95.1	95.8	94.4	94.5	89.4	95.2	92.3	92.1	92.0	94.8
macro-average	3106	**94.5**	**93.6**	93.3	91.2	91.2	90.8	90.5	90.4	89.7	88.4	87.4	86.9	85.6	85.6	84.9	84.8	84.2	84.0	88.7

Table 5: Accuracies (%) of successful translations on a category level for English-German. Boldface indicates the significantly best performing systems in each row.

category	count	best				Faceb		Onl-B				Volc		Onl-A				Onl-G				uedin				Onl-Y		
		2018	2019	2020	2021	2019	2021	2018	2019	2020	2021	2020	2021	2018	2019	2020	2021	2018	2019	2020	2021	2018	2019	2020	2021	2018	2019	2021
Ambiguity	76	76	92	83	86	92	89	76	78	79	86	78	86	68	70	78	82	72	75	84	86	50	62	75	84	67	79	83
Composition	45	98	98	98	96	98	98	98	98	98	100	98	100	80	93	93	96	71	82	96	98	76	84	93	96	89	91	93
Coordination & ellipsis	43	88	88	88	91	88	91	88	88	91	88	88	88	86	86	86	88	49	60	77	88	81	81	86	88	77	86	88
False friends	36	75	75	72	86	75	81	75	78	81	75	81	75	72	72	69	83	72	72	78	81	53	67	72	67	67	92	75
Function word	52	81	92	94	96	92	96	81	81	94	88	85	88	88	90	90	92	50	96	96	98	83	90	92	96	92	94	88
LDD & interrogatives	73	85	90	92	92	90	92	85	85	89	96	90	96	81	79	85	92	67	77	90	90	75	77	90	86	85	82	90
MWE	64	75	84	84	89	84	84	75	75	81	81	78	81	69	69	75	83	67	72	83	83	58	63	73	81	73	75	78
Named entity & terminology	57	91	91	96	96	91	93	91	91	88	91	91	91	91	91	95	93	88	86	91	93	82	91	95	96	91	89	93
Negation	17	94	100	100	100	100	100	94	94	100	100	100	100	100	100	100	100	65	100	100	100	100	94	100	100	100	100	100
Non-verbal agreement	56	88	91	91	96	91	95	88	88	88	98	89	98	77	84	84	91	57	80	91	91	71	84	89	89	79	82	86
Punctuation	35	97	97	97	97	97	100	97	97	94	100	100	100	100	100	100	100	83	83	86	86	94	91	100	100	100	100	100
Subordination	83	88	93	95	94	93	93	88	89	96	98	94	98	96	86	95	95	84	93	96	93	92	89	96	95	94	94	95
Verb tense/aspect/mood	3676	77	82	88	86	82	86	77	77	79	78	87	78	75	87	81	85	49	69	83	79	79	84	83	85	74	75	76
Verb valency	53	83	89	89	87	89	91	83	83	92	91	91	91	79	83	87	87	72	79	89	89	74	77	85	89	81	83	83
micro-avg	4366	78	83	88	87	83	87	78	78	80	80	88	80	76	86	82	86	52	71	84	80	78	83	84	85	75	77	78
macro-avg	4366	85	90	91	92	90	92	85	86	89	91	89	91	83	85	87	90	68	80	89	90	76	81	88	89	83	87	88

Table 6: Accuracies (%) of the German to English systems that were submitted also in previous years.

phenomenon	count	Onl-W	Faceb	Onl-B	VolcT	Onl-A	SMU	Onl-G	Huawe	borde	Nemo	uedin	Water	P3AI	ICL	Onl-Y	Manif	happy	avg
Ambiguity	74	**87.8**	**90.5**	**86.5**	**86.5**	**81.1**	**83.8**	**85.1**	**89.2**	**83.8**	**83.8**	**83.8**	75.7	79.7	**86.5**	**82.4**	**81.1**	60.8	82.8
Lexical ambiguity	61	**91.8**	**91.8**	**88.5**	**88.5**	**82.0**	**86.9**	**86.9**	**88.5**	**86.9**	**85.2**	**85.2**	78.7	**82.0**	**88.5**	**85.2**	**82.0**	62.3	84.8
Structural ambiguity	13	**69.2**	**84.6**	**76.9**	**76.9**	**76.9**	**69.2**	**76.9**	**92.3**	**69.2**	**76.9**	**76.9**	**61.5**	**69.2**	**76.9**	**69.2**	**76.9**	53.8	73.8
Composition	43	**97.7**	**97.7**	**100.0**	**100.0**	**97.7**	**95.3**	**97.7**	**95.3**	**95.3**	93.0	**97.7**	**97.7**	**97.7**	**95.3**	93.0	93.0	74.4	95.2
Compound	25	**96.0**	**96.0**	**100.0**	**100.0**	**96.0**	**92.0**	**96.0**	**96.0**	**92.0**	88.0	**96.0**	**96.0**	**96.0**	**92.0**	88.0	**92.0**	84.0	93.9
Phrasal verb	18	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.4**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.4**	61.1	97.1
Coordination & ellipsis	57	89.5	89.5	89.5	89.5	87.7	86.0	86.0	87.7	89.5	87.7	87.7	86.0	87.7	77.2	87.7	89.5	80.7	87.0
Gapping	15	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.3	93.3	93.3	93.3	100.0	100.0	86.7	97.6
Right node raising	15	80.0	80.0	80.0	80.0	80.0	73.3	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	73.3	80.0	73.3	78.8
Sluicing	13	100.0	100.0	92.3	92.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.1
Stripping	14	**78.6**	**78.6**	**85.7**	**85.7**	**71.4**	**71.4**	**64.3**	**71.4**	**78.6**	**71.4**	**78.6**	**71.4**	**78.6**	35.7	**78.6**	**78.6**	**64.3**	73.1
False friends	36	**86.1**	**80.6**	**75.0**	**75.0**	**83.3**	**83.3**	**80.6**	63.9	**77.8**	**72.2**	66.7	**80.6**	**80.6**	**72.2**	**75.0**	**69.4**	63.9	75.7
Function word	40	**92.5**	**92.5**	**92.5**	**92.5**	**90.0**	**85.0**	**95.0**	**92.5**	**85.0**	**92.5**	**92.5**	**92.5**	**85.0**	**87.5**	**90.0**	72.5	80.0	88.8
Focus particle	21	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**90.5**	**100.0**	**100.0**	**90.5**	**100.0**	**100.0**	**100.0**	**90.5**	**95.2**	**100.0**	**95.2**	85.7	96.9
Modal particle	14	78.6	78.6	78.6	78.6	78.6	71.4	85.7	78.6	71.4	78.6	78.6	85.7	71.4	71.4	71.4	64.3	78.6	76.5
Question tag	5	**100.0**	**100.0**	**100.0**	**100.0**	**80.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**80.0**	**100.0**	**100.0**	**100.0**	0.0	**60.0**	89.4

phenomenon	count	Onl-W	Faceb	Onl-B	VolcT	Onl-A	SMU	Onl-G	Huawe	borde	Nemo	uedin	Water	P3AI	ICL	Onl-Y	Manif	happy	avg
LDD & interrogatives	103	**91.3**	**91.3**	**91.3**	**91.3**	**91.3**	**91.3**	**90.3**	**93.2**	**90.3**	**91.3**	**89.3**	**92.2**	**89.3**	**91.3**	**88.3**	57.3	74.8	87.9
Extended adj. construction	9	**88.9**	**88.9**	**77.8**	**77.8**	**100.0**	**77.8**	**88.9**	**88.9**	66.7	**100.0**	**77.8**	**88.9**	66.7	**88.9**	55.6	66.7	**77.8**	81.0
Extraposition	11	90.9	90.9	90.9	90.9	81.8	100.0	81.8	90.9	100.0	90.9	90.9	90.9	90.9	90.9	90.9	100.0	81.8	90.9
Multiple connectors	13	**84.6**	**84.6**	**84.6**	**84.6**	**84.6**	**92.3**	76.9	**100.0**	**84.6**	**84.6**	**84.6**	**84.6**	**84.6**	**84.6**	76.9	**84.6**	**84.6**	85.1
Pied-piping	14	**92.9**	**85.7**	**85.7**	**85.7**	**85.7**	**92.9**	**85.7**	**85.7**	**92.9**	**85.7**	**85.7**	**92.9**	**92.9**	**85.7**	**85.7**	**92.9**	50.0	86.1
Polar question	12	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**91.7**	93.6
Scrambling	9	88.9	88.9	88.9	88.9	88.9	88.9	88.9	88.9	88.9	88.9	77.8	77.8	77.8	88.9	88.9	77.8	44.4	83.7
Topicalization	10	70.0	**90.0**	**90.0**	**90.0**	**90.0**	**100.0**	**90.0**	**90.0**	**90.0**	**90.0**	**90.0**	**100.0**	**90.0**	**90.0**	**100.0**	**90.0**	**80.0**	90.0
Wh-movement	25	**100.0**	**96.0**	**100.0**	**100.0**	**96.0**	84.0	**100.0**	**96.0**	**92.0**	**92.0**	**96.0**	**96.0**	**96.0**	**96.0**	**96.0**	8.0	80.0	89.6
MWE	66	**90.9**	**86.4**	**83.3**	**83.3**	**86.4**	**86.4**	**84.8**	**86.4**	**86.4**	**86.4**	**83.3**	**80.3**	**84.8**	**86.4**	**81.8**	**84.8**	69.7	84.2
Collocation	16	**100.0**	**100.0**	**93.8**	**93.8**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**87.5**	**100.0**	**100.0**	**93.8**	**100.0**	81.3	97.1
Idiom	12	58.3	33.3	25.0	25.0	33.3	25.0	25.0	25.0	25.0	25.0	16.7	16.7	16.7	25.0	16.7	16.7	16.7	25.0
Prepositional MWE	19	**94.7**	**94.7**	**94.7**	**94.7**	**100.0**	**100.0**	**94.7**	**100.0**	**100.0**	**100.0**	**94.7**	**94.7**	**100.0**	**100.0**	**94.7**	**100.0**	78.9	96.3
Verbal MWE	19	**100.0**	**100.0**	**100.0**	**100.0**	**94.7**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	84.2	98.8
Named entity & terminology	71	**95.8**	**94.4**	**93.0**	**93.0**	**94.4**	**93.0**	**94.4**	**95.8**	**91.5**	**91.5**	**95.8**	**88.7**	**90.1**	**91.5**	**93.0**	**90.1**	83.1	92.3
Date	17	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.1	100.0	100.0	94.1	100.0	99.3
Domainspecific term	10	80.0	80.0	70.0	70.0	70.0	70.0	80.0	80.0	70.0	70.0	80.0	60.0	70.0	70.0	70.0	70.0	60.0	71.8
Location	19	94.7	94.7	94.7	94.7	94.7	94.7	94.7	94.7	89.5	89.5	94.7	89.5	94.7	94.7	94.7	94.7	78.9	92.9
Measuring unit	19	**100.0**	**94.7**	**94.7**	**94.7**	**100.0**	**94.7**	**100.0**	**100.0**	**94.7**	**94.7**	**100.0**	**89.5**	**89.5**	**89.5**	**94.7**	**89.5**	78.9	94.1
Proper name	6	100.0	100.0	100.0	100.0	100.0	100.0	83.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.0
Negation	14	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Non-verbal agreement	57	**98.2**	**94.7**	**98.2**	**98.2**	**93.0**	**91.2**	89.5	**93.0**	**93.0**	**93.0**	89.5	89.5	**91.2**	**93.0**	84.2	**93.0**	73.7	91.5
Coreference	19	**100.0**	**89.5**	**94.7**	**94.7**	84.2	78.9	73.7	78.9	78.9	78.9	73.7	78.9	73.7	78.9	73.7	78.9	52.6	80.2
External possessor	20	**95.0**	**95.0**	**100.0**	**100.0**	**95.0**	**95.0**	**95.0**	**100.0**	**100.0**	**100.0**	**95.0**	**90.0**	**100.0**	**100.0**	80.0	**100.0**	70.0	94.7
Internal possessor	18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Punctuation	18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Comma	18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Subordination	115	**92.2**	**93.9**	**95.7**	**95.7**	**93.9**	**92.2**	**93.0**	**92.2**	**92.2**	**93.9**	**93.9**	**94.8**	**93.0**	**93.9**	**93.9**	**93.9**	87.0	93.2
Adverbial clause	17	82.4	**88.2**	**100.0**	**100.0**	**94.1**	**94.1**	**88.2**	**94.1**	**94.1**	**88.2**	**94.1**	**94.1**	**94.1**	**94.1**	**94.1**	**94.1**	**100.0**	93.4
Cleft sentence	14	92.9	92.9	92.9	92.9	92.9	85.7	85.7	92.9	85.7	92.9	92.9	92.9	92.9	92.9	92.9	92.9	85.7	91.2
Free relative clause	12	91.7	83.3	83.3	83.3	83.3	83.3	91.7	75.0	83.3	91.7	91.7	91.7	83.3	83.3	83.3	83.3	83.3	85.3
Indirect speech	9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	77.8	98.7
Infinitive clause	17	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.1	99.7
Object clause	14	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	85.7	99.2
Pseudo-cleft sentence	9	66.7	88.9	77.8	77.8	77.8	77.8	77.8	66.7	77.8	77.8	66.7	77.8	66.7	88.9	77.8	88.9	55.6	75.8
Relative clause	13	92.3	92.3	100.0	100.0	92.3	84.6	92.3	92.3	84.6	92.3	92.3	92.3	92.3	84.6	100.0	84.6	92.3	91.9
Subject clause	10	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	90.0	100.0	90.0	98.8
Verb tense/aspect/mood	3058	**87.3**	**87.3**	79.6	79.6	**86.4**	**85.8**	80.5	82.7	**86.5**	83.9	**86.9**	84.1	81.3	82.6	77.7	84.1	71.1	82.8
Conditional	14	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	92.9	100.0	92.9	99.2
Ditransitive - future I	23	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	91.3	99.5
Ditransitive - future I subjunct. II	28	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.4	99.8
Ditransitive - future II	14	100.0	100.0	100.0	100.0	100.0	100.0	92.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	92.9	99.2
Ditransitive - future II subjunct. II	27	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.3	99.8

phenomenon	count	Onl-W	Faceb	Onl-B	VolcT	Onl-A	SMU	Onl-G	Huawe	borde	Nemo	uedin	Water	P3AI	ICL	Onl-Y	Manif	happy	avg
Ditransitive - perfect	23	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Ditransitive - pluperfect	27	**100.0**	**92.6**	63.0	63.0	**92.6**	**92.6**	48.1	77.8	**96.3**	85.2	**100.0**	85.2	85.2	85.2	7.4	**92.6**	**92.6**	80.0
Ditransitive - pluperf. subjunct. II	29	**100.0**	**96.6**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**93.1**	**100.0**	**96.6**	**100.0**	**100.0**	62.1	97.0
Ditransitive - present	26	84.6	**96.2**	88.5	88.5	88.5	**100.0**	76.9	**92.3**	**96.2**	**96.2**	**96.2**	**92.3**	**96.2**	88.5	80.8	**96.2**	76.9	90.3
Ditransitive - preterite	21	**100.0**	**90.5**	**100.0**	**100.0**	85.7	**90.5**	85.7	**90.5**	85.7	**95.2**	**95.2**	**90.5**	**95.2**	**95.2**	85.7	**90.5**	81.0	91.6
Ditransitive - preterite subjunct. II	17	100.0	100.0	100.0	100.0	94.1	100.0	94.1	94.1	100.0	100.0	100.0	88.2	100.0	100.0	100.0	100.0	94.1	97.9
Imperative	15	**93.3**	**93.3**	**86.7**	**86.7**	**93.3**	**93.3**	**93.3**	**86.7**	**86.7**	**86.7**	**86.7**	**80.0**	**93.3**	**93.3**	**86.7**	**86.7**	60.0	87.5
Intransitive - future I	31	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Intransitive - future I subjunct. II	30	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.7	100.0	96.7	100.0	100.0	99.6
Intransitive - future II	34	91.2	**97.1**	**94.1**	**94.1**	91.2	91.2	85.3	**100.0**	**100.0**	**94.1**	**97.1**	79.4	85.3	**97.1**	79.4	**97.1**	55.9	90.0
Intransitive - future II subjunct. II	35	**94.3**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.1**	**100.0**	**100.0**	**100.0**	**100.0**	**94.3**	65.7	**94.3**	**94.3**	82.9	57.1	92.9
Intransitive - perfect	72	**100.0**	**100.0**	**98.6**	**98.6**	**100.0**	95.8	**100.0**	**100.0**	95.8	**100.0**	**100.0**	91.7	**97.2**	94.4	**100.0**	**100.0**	91.7	97.9
Intransitive - pluperfect	31	**87.1**	71.0	19.4	19.4	**93.5**	77.4	41.9	**87.1**	77.4	**87.1**	**87.1**	67.7	**96.8**	71.0	41.9	**83.9**	74.2	69.6
Intransitive - pluperf. subjunct. II	35	**97.1**	**100.0**	**100.0**	**100.0**	**94.3**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.3**	85.7	**94.3**	**100.0**	**94.3**	**100.0**	51.4	94.8
Intransitive - present	25	100.0	100.0	96.0	96.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.0	100.0	92.0	98.8
Intransitive - preterite	49	**95.9**	**95.9**	**98.0**	**98.0**	**95.9**	91.8	**95.9**	**98.0**	93.9	**95.9**	**95.9**	93.9	**98.0**	91.8	91.8	**100.0**	91.8	95.4
Intransitive - preterite subjunct. II	23	**82.6**	**87.0**	**91.3**	**91.3**	**78.3**	**82.6**	**82.6**	**91.3**	**82.6**	**95.7**	**82.6**	69.6	**82.6**	**82.6**	**91.3**	**95.7**	**87.0**	85.7
Modal - future I	115	**93.0**	**93.9**	84.3	84.3	**95.7**	**94.8**	**93.9**	89.6	**93.9**	**93.0**	**91.3**	**93.9**	**93.0**	**91.3**	79.1	**90.4**	83.5	90.5
Modal - future I subjunct. II	111	**86.5**	**84.7**	82.0	82.0	**91.9**	81.1	82.0	83.8	80.2	**85.6**	**85.6**	83.8	80.2	73.0	**88.3**	68.5	75.7	82.0
Modal - perfect	113	72.6	**89.4**	68.1	68.1	75.2	**85.8**	73.5	53.1	**85.0**	68.1	78.8	**86.7**	57.5	74.3	70.8	72.6	44.2	72.0
Modal - pluperfect	124	**60.5**	38.7	4.8	4.8	41.1	40.3	17.7	11.3	37.1	29.0	41.1	10.5	25.0	11.3	4.0	13.7	9.7	23.6
Modal - pluperf. subjunct. II	137	**61.3**	**59.1**	**58.4**	**58.4**	**59.9**	**60.6**	**54.0**	**56.2**	**60.6**	**55.5**	**60.6**	**54.0**	**59.9**	**60.6**	**58.4**	**59.9**	39.4	57.4
Modal - present	102	**98.0**	92.2	89.2	89.2	**94.1**	**95.1**	81.4	92.2	**95.1**	**95.1**	**96.1**	**96.1**	86.3	93.1	72.5	93.1	88.2	91.0
Modal - preterite	123	97.6	**99.2**	**98.4**	**98.4**	**100.0**	96.7	97.6	96.7	97.6	91.1	**100.0**	**100.0**	89.4	91.9	96.7	95.9	90.2	96.3
Modal - preterite subjunct. II	111	**82.9**	**86.5**	**79.3**	**78.4**	**87.4**	**85.6**	**87.4**	**85.6**	**85.6**	77.5	**87.4**	**84.7**	73.9	**82.0**	**82.9**	**84.7**	76.6	82.8
Modal neg. - future I	97	**94.8**	**95.9**	**95.9**	**95.9**	**92.8**	**92.8**	**95.9**	**90.7**	87.6	**95.9**	**91.8**	**93.8**	85.6	**91.8**	**95.9**	**89.7**	74.2	91.8
Modal neg. - future I subjunct. II	125	**95.2**	**95.2**	91.2	91.2	**96.0**	**94.4**	**95.2**	**93.6**	**95.2**	**95.2**	**95.2**	**96.0**	91.2	**93.6**	**97.6**	92.0	83.2	93.6
Modal neg. - perfect	87	80.5	**86.2**	71.3	71.3	77.0	**89.7**	79.3	64.4	**90.8**	72.4	**86.2**	**88.5**	70.1	79.3	78.2	79.3	48.3	77.2
Modal neg. - pluperfect	102	**66.7**	38.2	3.9	3.9	27.5	8.8	3.9	7.8	6.9	17.6	31.4	18.6	20.6	**1.0**	7.8	17.6	18.6	17.7
Modal neg. - pluperf. subjunct. II	122	**70.5**	64.8	55.7	55.7	66.4	**72.1**	53.3	66.4	**75.4**	50.0	**80.3**	69.7	**71.3**	**71.3**	50.8	**76.2**	47.5	64.6
Modal neg. - present	125	92.0	96.8	91.2	91.2	96.0	93.6	76.0	96.0	95.2	**98.4**	**100.0**	97.6	86.4	94.4	73.6	92.8	96.8	92.2
Modal neg. - preterite	128	**99.2**	**99.2**	96.9	96.9	**100.0**	**98.4**	**100.0**	**98.4**	**100.0**	**100.0**	**100.0**	**100.0**	97.7	**98.4**	**98.4**	**98.4**	89.8	98.3
Modal neg. - preterite subjunct. II	118	**93.2**	**91.5**	66.9	66.9	83.1	85.6	**93.2**	**91.5**	**94.1**	83.1	83.9	75.4	**89.8**	82.2	73.7	**90.7**	83.9	84.0
Progressive	11	**90.9**	**90.9**	72.7	72.7	72.7	**81.8**	**81.8**	**100.0**	**100.0**	**81.8**	**90.9**	**81.8**	**90.9**	**100.0**	**90.9**	**90.9**	**81.8**	86.6
Reflexive - future I	21	**76.2**	**95.2**	**90.5**	**90.5**	**95.2**	**95.2**	**76.2**	**90.5**	**95.2**	**90.5**	**81.0**	**95.2**	**90.5**	**95.2**	**81.0**	**95.2**	66.7	88.2
Reflexive - future I subjunct. II	32	71.9	**87.5**	**93.8**	**93.8**	**96.9**	**96.9**	71.9	**93.8**	**93.8**	**93.8**	68.8	**96.9**	**93.8**	**93.8**	75.0	**96.9**	62.5	87.1
Reflexive - future II	24	83.3	**95.8**	**95.8**	**95.8**	**91.7**	**95.8**	87.5	**91.7**	**95.8**	**95.8**	87.5	**100.0**	87.5	**91.7**	87.5	**100.0**	62.5	90.9
Reflexive - future II subjunct. II	29	79.3	**89.7**	**96.6**	**96.6**	**86.2**	75.9	79.3	**96.6**	**96.6**	**96.6**	72.4	**96.6**	44.8	**96.6**	**82.8**	**82.8**	44.8	83.2
Reflexive - perfect	27	77.8	**100.0**	**92.6**	**92.6**	**96.3**	**96.3**	**92.6**	**92.6**	**96.3**	**96.3**	**92.6**	**100.0**	81.5	**96.3**	81.5	**96.3**	55.6	90.4
Reflexive - pluperfect	28	71.4	**89.3**	**82.1**	**82.1**	**96.4**	**92.9**	78.6	**96.4**	**92.9**	**89.3**	**92.9**	**96.4**	**92.9**	**92.9**	78.6	**96.4**	53.6	86.8
Reflexive - pluperf. subjunct. II	29	72.4	**82.8**	**93.1**	**93.1**	**89.7**	**82.8**	79.3	75.9	**86.2**	**89.7**	72.4	**86.2**	79.3	**96.6**	75.9	79.3	44.8	81.1
Reflexive - present	23	69.6	**91.3**	**95.7**	**95.7**	**95.7**	**95.7**	78.3	**91.3**	**91.3**	**91.3**	**91.3**	**100.0**	87.0	**91.3**	87.0	**95.7**	73.9	89.5
Reflexive - preterite	17	76.5	**94.1**	**94.1**	**94.1**	82.4	**88.2**	82.4	82.4	**88.2**	**94.1**	**88.2**	**100.0**	**88.2**	82.4	82.4	**94.1**	35.3	85.1

phenomenon	count	Onl-W	Faceb	Onl-B	VolcT	Onl-A	SMU	Onl-G	Huawe	borde	Nemo	uedin	Water	P3AI	ICL	Onl-Y	Manif	happy	avg
Reflexive - preterite subjunct. II	15	80.0	**100.0**	**100.0**	**100.0**	**86.7**	73.3	**86.7**	**93.3**	73.3	**93.3**	**93.3**	**93.3**	**86.7**	73.3	80.0	**93.3**	46.7	85.5
Transitive - future I	40	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5	97.5
Transitive - future I subjunct. II	35	100.0	100.0	97.1	97.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	97.1	99.5
Transitive - future II	35	100.0	97.1	94.3	94.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	97.1	100.0	100.0	97.1	98.8
Transitive - future II subjunct. II	35	97.1	100.0	97.1	97.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	97.1	100.0	100.0	100.0	94.3	99.0
Transitive - perfect	40	100.0	100.0	95.0	95.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.4
Transitive - pluperfect	31	**100.0**	90.3	58.1	58.1	**93.5**	**100.0**	80.6	**100.0**	**100.0**	**100.0**	**100.0**	80.6	**93.5**	**100.0**	58.1	**96.8**	**93.5**	88.4
Transitive - pluperf. subjunct. II	32	**100.0**	**96.9**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.9**	**96.9**	**93.8**	**100.0**	**96.9**	**100.0**	**96.9**	50.0	95.8
Transitive - present	36	**97.2**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	91.7	**100.0**	**94.4**	99.0
Transitive - preterite	26	**92.3**	**100.0**	**96.2**	**96.2**	**100.0**	**96.2**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**92.3**	**92.3**	**100.0**	**100.0**	88.5	97.3
Transitive - preterite subjunct. II	23	**82.6**	**82.6**	**82.6**	**82.6**	**82.6**	**78.3**	**73.9**	**82.6**	**87.0**	**87.0**	**73.9**	**82.6**	**73.9**	**78.3**	**91.3**	**87.0**	56.5	80.3
Verb valency	54	88.9	90.7	92.6	92.6	90.7	90.7	87.0	90.7	90.7	90.7	88.9	88.9	88.9	90.7	85.2	90.7	81.5	89.4
Case government	17	94.1	94.1	94.1	94.1	94.1	94.1	94.1	94.1	94.1	94.1	88.2	88.2	94.1	94.1	82.4	94.1	88.2	92.4
Mediopassive voice	13	100.0	100.0	100.0	100.0	100.0	100.0	84.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	92.3	100.0	84.6	97.7
Passive voice	15	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	86.7	99.2
Resultative predicates	9	44.4	55.6	66.7	66.7	55.6	55.6	55.6	55.6	55.6	55.6	55.6	55.6	44.4	55.6	55.6	55.6	55.6	55.6
micro-average	3806	**88.3**	**88.2**	82.0	81.9	**87.3**	86.6	82.4	84.3	**87.1**	85.1	**87.4**	85.0	82.8	83.9	79.7	84.0	72.3	84.0
phen. macro-average	3806	90.3	**91.6**	88.4	88.4	**90.5**	90.0	87.0	89.9	90.4	90.3	90.0	88.9	87.6	89.0	85.2	87.1	75.8	88.3
categ. macro-average	3806	**92.7**	**92.1**	91.2	91.2	91.1	90.3	90.3	90.2	90.1	90.0	89.7	89.3	89.2	89.2	88.0	85.7	78.6	89.4

Table 7: Accuracies (%) of successful translations on a phenomenon-level granularity for German-English, organized in categories. Boldface indicates the best scoring system in each row, including all systems which are not significantly inferior than the best scoring system. Grey rows average the accuracies of the phenomena per category.

categ	count	Faceb	VolcA	Onl-W	Onl-A	Huawe	Nemo	Onl-B	VolcG	uedin	P3AI	eTran	happy	nucle	Onl-Y	Manif	BUPT	ICL	Onl-G	avg
Ambiguity	23	**91.3**	**95.7**	**95.7**	**91.3**	**87.0**	**87.0**	**91.3**	**91.3**	**82.6**	**82.6**	**78.3**	73.9	73.9	69.6	**82.6**	69.6	**82.6**	73.9	83.3
Lexical ambiguity	23	**91.3**	**95.7**	**95.7**	**91.3**	**87.0**	**87.0**	**91.3**	**91.3**	**82.6**	**82.6**	**78.3**	73.9	73.9	69.6	**82.6**	69.6	**82.6**	73.9	83.3
Coordination & ellipsis	87	**81.6**	**71.3**	**71.3**	**73.6**	**77.0**	**75.9**	**80.5**	**79.3**	69.0	69.0	66.7	64.4	65.5	**71.3**	63.2	63.2	58.6	**72.4**	70.8
Gapping	16	81.3	68.8	56.3	75.0	87.5	81.3	87.5	87.5	68.8	81.3	68.8	62.5	56.3	81.3	62.5	68.8	56.3	75.0	72.6
Pseudogapping	9	88.9	77.8	77.8	55.6	55.6	55.6	55.6	55.6	66.7	55.6	55.6	66.7	44.4	44.4	55.6	55.6	55.6	66.7	60.5
Right node raising	14	92.9	78.6	92.9	78.6	85.7	85.7	71.4	71.4	92.9	92.9	85.7	92.9	92.9	64.3	92.9	92.9	85.7	85.7	85.3
Sluicing	19	**94.7**	**100.0**	**94.7**	73.7	84.2	84.2	78.9	78.9	78.9	78.9	73.7	78.9	78.9	73.7	84.2	73.7	68.4	68.4	80.4
Stripping	19	**73.7**	36.8	42.1	68.4	63.2	**73.7**	**94.7**	**89.5**	42.1	47.4	42.1	42.1	52.6	**73.7**	36.8	42.1	36.8	68.4	57.0
VP-ellipsis	10	**50.0**	**70.0**	**70.0**	**90.0**	**80.0**	**60.0**	**80.0**	**80.0**	**70.0**	**50.0**	**80.0**	40.0	**60.0**	**80.0**	40.0	40.0	**50.0**	**70.0**	64.4
False friends	38	92.1	92.1	89.5	86.8	86.8	84.2	84.2	84.2	86.8	86.8	86.8	86.8	81.6	86.8	86.8	86.8	84.2	84.2	86.5
Function word	35	**97.1**	**97.1**	**100.0**	**97.1**	**97.1**	**94.3**	**100.0**	**100.0**	**100.0**	**97.1**	**94.3**	**97.1**	**97.1**	**97.1**	65.7	**97.1**	**100.0**	**97.1**	95.9
Focus particle	23	95.7	95.7	100.0	95.7	95.7	91.3	100.0	100.0	100.0	95.7	91.3	95.7	95.7	95.7	100.0	95.7	100.0	95.7	96.6
Question tag	12	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	94.4
MWE	98	**89.8**	**93.9**	**91.8**	85.7	**87.8**	**88.8**	**90.8**	**90.8**	82.7	85.7	84.7	**89.8**	82.7	82.7	83.7	81.6	80.6	81.6	86.4
Collocation	16	**100.0**	**100.0**	**100.0**	**87.5**	**93.8**	**100.0**	**100.0**	**100.0**	**87.5**	**100.0**	**93.8**	**93.8**	**87.5**	**87.5**	**87.5**	**87.5**	81.3	**87.5**	93.1
Compound	17	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.1	100.0	100.0	99.7
Idiom	11	**36.4**	**63.6**	**45.5**	0.0	**18.2**	**9.1**	**27.3**	**27.3**	0.0	0.0	**9.1**	**27.3**	0.0	0.0	0.0	0.0	0.0	0.0	14.6

categ	count	Faceb	VolcA	Onl-W	Onl-A	Huawe	Nemo	Onl-B	VolcG	uedin	P3AI	eTran	happy	nucle	Onl-Y	Manif	BUPT	ICL	Onl-G	avg
Nominal MWE	18	**88.9**	**94.4**	**88.9**	**94.4**	**94.4**	**94.4**	**100.0**	**100.0**	83.3	**94.4**	83.3	**100.0**	83.3	**88.9**	**88.9**	83.3	83.3	**88.9**	90.7
Prepositional MWE	15	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Verbal MWE	21	**95.2**	**95.2**	**100.0**	**100.0**	**95.2**	**100.0**	**95.2**	**95.2**	**95.2**	**90.5**	**95.2**	**95.2**	**95.2**	**90.5**	**95.2**	**95.2**	**90.5**	85.7	94.7
Named entity & terminology	82	**93.9**	**97.6**	**93.9**	**93.9**	**93.9**	89.0	**93.9**	**93.9**	**92.7**	89.0	**93.9**	90.2	90.2	**92.7**	89.0	**92.7**	81.7	80.5	91.3
Date	16	**100.0**	**100.0**	**100.0**	**93.8**	**100.0**	**100.0**	**100.0**	**100.0**	**93.8**	**93.8**	**93.8**	**93.8**	**93.8**	**100.0**	**100.0**	**87.5**	81.3	**87.5**	95.5
Domainspecific term	11	**90.9**	**90.9**	72.7	**100.0**	**90.9**	**100.0**	**90.9**	**90.9**	**100.0**	**90.9**	**100.0**	**81.8**	**90.9**	**100.0**	**81.8**	**90.9**	**81.8**	**90.9**	90.9
Location	19	94.7	94.7	94.7	94.7	94.7	94.7	94.7	94.7	94.7	94.7	94.7	89.5	94.7	89.5	94.7	94.7	84.2	89.5	93.3
Measuring unit	17	82.4	**100.0**	**94.1**	**88.2**	**94.1**	70.6	**88.2**	**88.2**	**94.1**	**94.1**	**94.1**	**94.1**	**88.2**	**94.1**	**94.1**	**100.0**	**94.1**	82.4	90.8
Proper name	19	**100.0**	**100.0**	**100.0**	**94.7**	**89.5**	84.2	**94.7**	**94.7**	84.2	73.7	**89.5**	**89.5**	84.2	84.2	73.7	**89.5**	68.4	57.9	86.3
Negation	15	100.0	100.0	100.0	93.3	93.3	100.0	93.3	93.3	100.0	100.0	93.3	93.3	86.7	100.0	100.0	93.3	93.3	93.3	95.9
Non-verbal agreement	68	**100.0**	**98.5**	**97.1**	95.6	95.6	92.6	92.6	92.6	92.6	89.7	91.2	92.6	88.2	89.7	92.6	88.2	88.2	89.7	92.6
Coreference	26	**100.0**	**96.2**	**96.2**	88.5	**92.3**	88.5	88.5	88.5	84.6	80.8	84.6	**92.3**	76.9	88.5	84.6	84.6	80.8	80.8	87.6
Genitive	15	**100.0**	**100.0**	**93.3**	**100.0**	**93.3**	**93.3**	**93.3**	**93.3**	**93.3**	**86.7**	**86.7**	80.0	**86.7**	**86.7**	**93.3**	73.3	80.0	**86.7**	90.0
Possession	27	100.0	100.0	100.0	100.0	100.0	96.3	96.3	96.3	100.0	100.0	100.0	100.0	100.0	92.6	100.0	100.0	100.0	100.0	99.0
Punctuation	37	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	78.4	78.4	91.9	81.1	78.4	81.1	86.5	75.7	78.4	78.4	78.4	70.3	86.5
Quotation marks	37	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	78.4	78.4	91.9	81.1	78.4	81.1	86.5	75.7	78.4	78.4	78.4	70.3	86.5
Subordination	161	**99.4**	**98.1**	**98.1**	**99.4**	95.7	**99.4**	**98.1**	**98.1**	**98.1**	**98.8**	**98.1**	**96.9**	**97.5**	93.8	**96.9**	94.4	92.5	96.3	97.2
Adverbial clause	14	100.0	100.0	100.0	100.0	92.9	100.0	100.0	100.0	92.9	100.0	100.0	92.9	92.9	92.9	100.0	92.9	92.9	85.7	96.4
Cleft sentence	16	**100.0**	**93.8**	**87.5**	**93.8**	**87.5**	**93.8**	**93.8**	**93.8**	**93.8**	**93.8**	**93.8**	**93.8**	**93.8**	**87.5**	**93.8**	**93.8**	81.3	**93.8**	92.4
Contact clause	24	**95.8**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.8**	**95.8**	**100.0**	**100.0**	**95.8**	**91.7**	**95.8**	**95.8**	**91.7**	**91.7**	87.5	**95.8**	96.3
Indirect speech	10	100.0	80.0	90.0	100.0	100.0	100.0	90.0	100.0	100.0	100.0	100.0	100.0	100.0	90.0	90.0	90.0	100.0	90.0	95.6
Infinitive clause	16	100.0	100.0	100.0	100.0	87.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.8	100.0	99.0
Object clause	15	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.3	93.3	100.0	93.3	100.0	100.0	98.9
Pseudo-cleft sentence	18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.4	100.0	100.0	100.0	100.0	100.0	88.9	100.0	100.0	94.4	94.4	98.5
Relative clause	36	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	**100.0**	**100.0**	**100.0**	**97.2**	**97.2**	**97.2**	**97.2**	**100.0**	**94.4**	**97.2**	91.7	91.7	**100.0**	97.8
Subject clause	12	100.0	100.0	100.0	100.0	91.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.5
Verb tense/aspect/mood	2366	98.6	97.9	97.3	96.9	96.1	97.4	**99.0**	**99.1**	**99.2**	97.4	98.4	96.7	97.3	90.7	98.6	94.8	95.2	94.7	97.0
Conditional	15	93.3	86.7	93.3	93.3	93.3	86.7	80.0	80.0	93.3	93.3	93.3	93.3	93.3	93.3	93.3	93.3	86.7	86.7	90.4
Ditransitive - conditional I progr.	57	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**98.2**	93.0	**98.2**	**96.5**	**100.0**	99.2
Ditransitive - conditional I simple	55	**96.4**	90.9	**96.4**	81.8	**100.0**	94.5	**100.0**	**100.0**	**98.2**	**98.2**	**96.4**	**96.4**	**98.2**	**96.4**	**96.4**	89.1	92.7	**96.4**	95.5
Ditransitive - conditional II progr.	14	**100.0**	**100.0**	**100.0**	**100.0**	**85.7**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**85.7**	**85.7**	**100.0**	**100.0**	78.6	**100.0**	96.4
Ditransitive - conditional II simple	15	100.0	100.0	93.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	86.7	100.0	100.0	100.0	93.3	100.0	98.5
Ditransitive - future I progr.	39	97.4	100.0	100.0	94.9	100.0	97.4	97.4	97.4	97.4	100.0	97.4	100.0	100.0	100.0	100.0	100.0	100.0	94.9	98.6
Ditransitive - future I simple	67	88.1	**100.0**	95.5	95.5	**100.0**	**97.0**	**100.0**	**100.0**	**100.0**	**100.0**	95.5	**100.0**	**98.5**	91.0	**97.0**	94.0	**97.0**	94.0	96.8
Ditransitive - future II progr.	54	**96.3**	**98.1**	**96.3**	94.4	**98.1**	**96.3**	**98.1**	**98.1**	**98.1**	88.9	**100.0**	70.4	**98.1**	33.3	92.6	88.9	66.7	88.9	89.0
Ditransitive - future II simple	44	88.6	**100.0**	**100.0**	90.9	**100.0**	90.9	90.9	90.9	**97.7**	**100.0**	**100.0**	**100.0**	**95.5**	65.9	**100.0**	77.3	93.2	**95.5**	93.2
Ditransitive - past perf. progr.	47	**95.7**	**97.9**	93.6	83.0	**100.0**	87.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	74.5	**100.0**	87.2	**95.7**	78.7	94.1
Ditransitive - past perf. simple	49	**98.0**	**98.0**	**100.0**	**95.9**	**100.0**	91.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.9**	**95.9**	**98.0**	93.9	81.6	97.2
Ditransitive - past progr.	30	**93.3**	76.7	90.0	**100.0**	**100.0**	**93.3**	**96.7**	**96.7**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**93.3**	**100.0**	**100.0**	96.7
Ditransitive - present perf. progr.	38	**100.0**	**100.0**	**100.0**	89.5	**100.0**	**100.0**	**97.4**	**97.4**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.4**	**94.7**	**100.0**	**94.7**	**100.0**	98.4
Ditransitive - present perf. simple	44	**100.0**	90.9	**95.5**	93.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.7**	93.2	93.2	**100.0**	**95.5**	**100.0**	97.7
Ditransitive - present progr.	38	**100.0**	**100.0**	**97.4**	92.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.4**	**97.4**	**100.0**	**97.4**	**94.7**	**100.0**	**97.4**	**94.7**	98.2
Ditransitive - simple past	53	100.0	98.1	98.1	96.2	100.0	98.1	98.1	98.1	100.0	100.0	100.0	98.1	100.0	100.0	100.0	100.0	98.1	100.0	99.1

categ	count	Faceb	VolcA	Onl-W	Onl-A	Huawe	Nemo	Onl-B	VolcG	uedin	P3AI	eTran	happy	nucle	Onl-Y	Manif	BUPT	ICL	Onl-G	avg
Ditransitive - simple present	36	100.0	100.0	100.0	97.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	97.2	100.0	100.0	100.0	100.0	99.7
Gerund	21	**100.0**	**95.2**	**95.2**	**100.0**	**100.0**	**95.2**	**95.2**	**100.0**	**95.2**	**100.0**	**95.2**	**100.0**	85.7	**90.5**	**100.0**	81.0	**95.2**	**95.2**	95.5
Imperative	9	100.0	100.0	100.0	88.9	100.0	100.0	100.0	100.0	100.0	100.0	88.9	100.0	88.9	100.0	88.9	77.8	88.9	88.9	95.1
Intransitive - conditional I progr.	24	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.8**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	87.5	**100.0**	99.1
Intransitive - conditional I simple	28	100.0	100.0	100.0	96.4	100.0	100.0	92.9	92.9	100.0	100.0	92.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98.6
Intransitive - future I progr.	27	**100.0**	**100.0**	**96.3**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	59.3	**100.0**	77.8	**100.0**	**96.3**	**100.0**	74.1	40.7	22.2	87.0
Intransitive - future I simple	46	100.0	95.7	97.8	100.0	100.0	97.8	97.8	97.8	100.0	100.0	100.0	100.0	97.8	100.0	100.0	97.8	100.0	97.8	98.9
Intransitive - future II progr.	10	**100.0**	**100.0**	**100.0**	**100.0**	**90.0**	**100.0**	**100.0**	**100.0**	**90.0**	20.0	**100.0**	50.0	**80.0**	30.0	**90.0**	0.0	20.0	50.0	73.3
Intransitive - future II simple	24	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	87.5	**100.0**	**100.0**	**95.8**	**100.0**	70.8	**100.0**	**91.7**	**100.0**	**100.0**	97.0
Intransitive - past perf. progr.	18	**94.4**	**88.9**	**94.4**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.4**	**100.0**	**100.0**	**100.0**	72.2	**100.0**	**88.9**	**100.0**	72.2	94.8
Intransitive - past perf. simple	30	**100.0**	**100.0**	**96.7**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	83.3	98.9
Intransitive - past progr.	11	100.0	100.0	100.0	100.0	81.8	100.0	90.9	90.9	100.0	100.0	100.0	90.9	100.0	81.8	100.0	100.0	100.0	100.0	96.5
Intransitive - present perf. simple	25	100.0	100.0	92.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.6
Intransitive - present progr.	50	**98.0**	**98.0**	**98.0**	**100.0**	**100.0**	**98.0**	**100.0**	**100.0**	**100.0**	**96.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**98.0**	94.0	98.9
Intransitive - simple past	30	**100.0**	**100.0**	**100.0**	86.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.7**	**100.0**	**100.0**	**100.0**	99.1
Intransitive - simple present	24	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.8	100.0	95.8	100.0	100.0	100.0	99.5
Modal	226	**100.0**	**99.6**	**99.6**	**100.0**	**100.0**	**100.0**	**99.6**	**99.6**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	96.9	**100.0**	**100.0**	**99.6**	**100.0**	99.7
Modal negated	213	**99.5**	**99.5**	98.6	98.6	93.9	**99.1**	**100.0**	**100.0**	**99.5**	**99.1**	**99.5**	**99.5**	**99.1**	**99.5**	**100.0**	**99.1**	**99.1**	95.8	98.9
Reflexive - conditional I progr.	25	**96.0**	**100.0**	**100.0**	**100.0**	84.0	**96.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.0**	**100.0**	**96.0**	88.0	**100.0**	**100.0**	**100.0**	**100.0**	97.6
Reflexive - conditional I simple	22	**95.5**	**100.0**	**95.5**	86.4	77.3	**90.9**	**100.0**	**100.0**	**95.5**	**100.0**	86.4	**95.5**	**95.5**	86.4	**100.0**	**95.5**	**100.0**	**100.0**	94.4
Reflexive - conditional II progr.	6	100.0	100.0	83.3	100.0	66.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	83.3	83.3	100.0	100.0	100.0	100.0	95.4
Reflexive - conditional II simple	23	**100.0**	**100.0**	**95.7**	**100.0**	73.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.7**	78.3	**100.0**	**100.0**	**100.0**	87.0	96.1
Reflexive - future I progr.	20	**100.0**	**100.0**	**100.0**	**100.0**	85.0	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**90.0**	**90.0**	**90.0**	**100.0**	**90.0**	**95.0**	**100.0**	96.4
Reflexive - future I simple	40	**100.0**	**97.5**	**100.0**	**100.0**	77.5	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	87.5	**100.0**	**100.0**	**100.0**	**100.0**	97.6
Reflexive - future II progr.	24	**100.0**	**95.8**	**100.0**	**100.0**	87.5	**100.0**	**100.0**	**100.0**	**100.0**	75.0	**95.8**	62.5	**95.8**	54.2	**95.8**	66.7	70.8	83.3	88.0
Reflexive - future II simple	30	**100.0**	**100.0**	90.0	**100.0**	83.3	**100.0**	**96.7**	**96.7**	**100.0**	**100.0**	**100.0**	**100.0**	90.0	63.3	**100.0**	**96.7**	**100.0**	**100.0**	95.4
Reflexive - past perf. progr.	28	**100.0**	**100.0**	**100.0**	**92.9**	82.1	85.7	**100.0**	**100.0**	**100.0**	**100.0**	89.3	**96.4**	**96.4**	71.4	**100.0**	85.7	**100.0**	**100.0**	94.4
Reflexive - past perf. simple	31	**100.0**	**100.0**	**100.0**	**100.0**	87.1	**96.8**	**100.0**	**100.0**	**100.0**	**100.0**	87.1	**100.0**	**100.0**	90.3	**100.0**	**96.8**	**100.0**	**96.8**	97.5
Reflexive - past progr.	4	100.0	50.0	50.0	100.0	75.0	50.0	75.0	75.0	100.0	100.0	75.0	50.0	75.0	50.0	100.0	75.0	100.0	50.0	75.0
Reflexive - present perf. progr.	23	**100.0**	**100.0**	**100.0**	**100.0**	82.6	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.7**	**100.0**	**100.0**	**100.0**	**100.0**	98.8
Reflexive - present perf. simple	30	**100.0**	**100.0**	**100.0**	**100.0**	90.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**93.3**	**100.0**	**100.0**	**100.0**	**100.0**	99.1
Reflexive - present progr.	25	**96.0**	**100.0**	**100.0**	**96.0**	84.0	**96.0**	**100.0**	**100.0**	**96.0**	**96.0**	88.0	**92.0**	**92.0**	**92.0**	**96.0**	**96.0**	**92.0**	**92.0**	94.7
Reflexive - simple past	32	**100.0**	**100.0**	**96.9**	**100.0**	84.4	**100.0**	**96.9**	**96.9**	**100.0**	**100.0**	90.6	**96.9**	87.5	84.4	**100.0**	90.6	**96.9**	**93.8**	95.3
Reflexive - simple present	25	**96.0**	**100.0**	**100.0**	**96.0**	80.0	**96.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.0**	**96.0**	88.0	**100.0**	**100.0**	**96.0**	**100.0**	96.9
Transitive - future II progr.	27	**100.0**	**96.3**	**96.3**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	44.4	**100.0**	40.7	**100.0**	14.8	**92.6**	51.9	51.9	77.8	81.5
Transitive - conditional I progr.	26	**100.0**	84.6	88.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	98.5
Transitive - conditional I simple	30	**100.0**	**100.0**	90.0	83.3	**100.0**	**100.0**	**100.0**	**100.0**	86.7	**96.7**	**93.3**	**93.3**	**93.3**	**100.0**	**100.0**	86.7	**100.0**	**93.3**	95.4
Transitive - conditional II progr.	28	**100.0**	89.3	**92.9**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.4**	**100.0**	**96.4**	**100.0**	98.6
Transitive - conditional II simple	29	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.6	100.0	100.0	100.0	96.6	100.0	99.6
Transitive - future I progr.	21	**100.0**	85.7	81.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.2**	**100.0**	**100.0**	**95.2**	**100.0**	**100.0**	**95.2**	**95.2**	**100.0**	97.1
Transitive - future I simple	41	**100.0**	**100.0**	**100.0**	**97.6**	**100.0**	85.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	90.2	**95.1**	**100.0**	92.7	**100.0**	**97.6**	97.7
Transitive - future II simple	35	**100.0**	**97.1**	**97.1**	**100.0**	**100.0**	**100.0**	**97.1**	**97.1**	**100.0**	**100.0**	**100.0**	**100.0**	91.4	82.9	**100.0**	88.6	**94.3**	**100.0**	97.0
Transitive - past perf. progr.	27	**100.0**	**92.6**	74.1	**100.0**	**100.0**	**92.6**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.3**	**96.3**	77.8	**100.0**	**96.3**	**100.0**	**92.6**	95.5

categ	count	Faceb	VolcA	Onl-W	Onl-A	Huawe	Nemo	Onl-B	VolcG	uedin	P3AI	eTran	happy	nucle	Onl-Y	Manif	BUPT	ICL	Onl-G	avg
Transitive - past perf. simple	30	100.0	100.0	96.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.7	100.0	96.7	99.4
Transitive - past progr.	4	25.0	100.0	100.0	25.0	25.0	25.0	100.0	100.0	25.0	25.0	50.0	25.0	25.0	100.0	25.0	25.0	25.0	75.0	50.0
Transitive - present perf. progr.	21	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.2	95.2	95.2	95.2	100.0	100.0	95.2	100.0	100.0	98.7
Transitive - present perf. simple	31	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.8	100.0	100.0	100.0	100.0	100.0	99.8
Transitive - present progr.	37	**100.0**	**97.3**	**97.3**	**100.0**	**100.0**	91.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.6**	**97.3**	**100.0**	**100.0**	**97.3**	**94.6**	98.3
Transitive - simple past	40	**100.0**	**100.0**	**100.0**	92.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**97.5**	**100.0**	92.5	98.8
Transitive - simple present	40	**100.0**	**100.0**	**97.5**	**100.0**	**97.5**	**100.0**	**97.5**	**97.5**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**97.5**	**100.0**	90.0	98.5
Verb valency	96	**90.6**	81.3	**85.4**	81.3	**84.4**	81.3	**83.3**	**83.3**	81.3	**83.3**	**84.4**	80.2	80.2	77.1	81.3	77.1	75.0	74.0	81.4
Case government	20	90.0	90.0	85.0	90.0	95.0	90.0	95.0	95.0	95.0	95.0	95.0	95.0	95.0	85.0	95.0	90.0	90.0	90.0	91.9
Catenative verb	20	**100.0**	**90.0**	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	**95.0**	**100.0**	**100.0**	**90.0**	**90.0**	85.0	96.7
Middle voice	19	68.4	63.2	63.2	52.6	42.1	36.8	47.4	47.4	47.4	42.1	52.6	42.1	36.8	42.1	42.1	36.8	31.6	31.6	45.9
Passive voice	17	100.0	94.1	88.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	88.2	100.0	100.0	100.0	94.1	98.0
Resultative	20	**95.0**	70.0	**95.0**	65.0	**85.0**	**80.0**	**75.0**	**75.0**	65.0	**80.0**	**75.0**	70.0	**75.0**	70.0	70.0	70.0	65.0	70.0	75.0
micro-average	3106	**97.4**	96.5	95.9	95.3	94.7	95.6	**96.9**	**96.9**	96.5	95.1	95.8	94.4	94.5	89.4	95.2	92.3	92.1	92.0	94.8
phen. macro-average	3106	**95.7**	94.6	93.9	93.3	91.7	93.0	**95.1**	**95.1**	93.8	91.8	93.1	90.8	90.8	86.8	91.7	88.3	88.2	89.1	92.1
categ. macro-average	3106	**94.5**	**93.6**	93.3	91.2	91.2	90.8	90.5	90.4	89.7	88.4	87.4	86.9	85.6	85.6	84.9	84.8	84.2	84.0	88.7

Table 8: Accuracies (%) of successful translations on a phenomenon-level granularity for English-German, organized in categories. Boldface indicates the best scoring system in each row, including all systems which are not significantly inferior than the best scoring system. Grey rows average the accuracies of the phenomena per category.

Pruning Neural Machine Translation for Speed Using Group Lasso

Maximiliana Behnke
University of Edinburgh
maximiliana.behnke@ed.ac.uk

Kenneth Heafield
University of Edinburgh
kheafiel@inf.ed.ac.uk

Abstract

Unlike most work on pruning neural networks, we make inference faster. Group lasso regularisation enables pruning entire rows, columns or blocks of parameters that result in a smaller dense network. Because the network is still dense, efficient matrix multiply routines are still used and only minimal software changes are required to support variable layer sizes. Moreover, pruning is applied during training so there is no separate pruning step. Experiments on top of English→German models, which already have state-of-the-art speed and size, show that two-thirds of feedforward connections can be removed with 0.2 BLEU loss. With 6 decoder layers, the pruned model is 34% faster; with 2 tied decoder layers, the pruned model is 14% faster. Pruning entire heads and feedforward connections in a 12-1 encoder-decoder architecture gains an additional 51% speed-up. These push the Pareto frontier with respect to the trade-off between time and quality compared to strong baselines. In the WMT 2021 Efficiency Task, our pruned and quantised models are 1.9–2.7× faster at the cost 0.9–1.7 BLEU in comparison to the unoptimised baselines. Across language pairs, we see similar sparsity patterns: an ascending or U-shaped distribution in encoder feedforward and attention layers and an ascending distribution in the decoder.

1 Introduction

Making transformer-based machine translation models (Vaswani et al., 2017) faster and smaller is a common requirement for server and mobile deployment. We focus on pruning methods that actually improve speed upon strong baselines. There is a variety of work on pruning individual parameters (See et al., 2016; Brix et al., 2020), structures like attention heads (Voita et al., 2019; Behnke and Heafield, 2020), and even whole layers (Sajjad et al., 2020). Unfortunately, much of the prior work on pruning does not report speed or makes inference slower: Brix et al. (2020) achieved no speed-up while Yao et al. (2019) report a 87.5% sparse model took 1.6× as long using cuSPARSE. However, Gale et al. (2020) point out that coefficient-sparse kernels like cuSPARSE are highly unoptimised. Even block-sparse kernels are 1.8× slower at 70% sparsity (Gray et al., 2017) though they did eventually achieve a 1.4× speed-up with "balanced pruning". We propose pruning entire rows or columns and even whole submatrices of a tensor, resulting in a smaller dense matrix. Because the inference problem remains dense, we sidestep the need for sparse kernels to improve speed.

We use group lasso (Yuan and Lin, 2006) regularisation, which encourages groups to diminish together, during the usual training procedure. Murray et al. (2019) used group lasso to prune feedforward layers in their submission to the Efficient Translation Task at WNGT 2019 (Kim et al., 2019a). Their submissions, which "eliminate more than 25% of the model's parameters while suffering a decrease of only 1.1 BLEU" were at best 6% faster than their baseline. When tuned for the same quality loss, our method reduces size by 66% and translates 52% faster. Moreover, their submissions were slower than higher-quality competitors by an order of magnitude, whereas our baselines are state-of-the-art.

Too much work (Gu et al., 2018; Lee et al., 2020; Wang et al., 2020b) on efficiency compares a baseline unoptimized system with their optimized system, which is smaller or faster in exchange for some reduction in BLEU. What these papers fail to prove is that their method works better than existing methods that also make models smaller or faster in exchange for some reduction in BLEU like knowledge distillation (Kim and Rush, 2016), quantisation, reducing the number of layers, prior work on pruning, or simply training a smaller model. In other words, is the trade-off offered by their method any better than the trade-offs already available, regardless of the type of method? Stacking the exist-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1074–1086
November 10–11, 2021.

ing methods produces a variety of data points with different speed and quality. The Pareto frontier is the set of data points that a practitioner would choose from: no other data point is simultaneously faster (or smaller) and of higher quality. We argue that a new method's empirical justification should advance the Pareto frontier. In this work, we build upon and compare to strong baselines to show the frontier advances.

To compare with the state-of-the-art in terms of speed and to investigate the impact this pruning makes on different languages, we build upon English→German, Spanish→English and Estonian→English student models trained with sequence-level knowledge distillation (Kim and Rush, 2016). We experiment with four architecture variations: a typical decoder with 6 layers and faster variations with shallow decoder of only 1–2 layers. We also include our experiments from the WMT 2021 Efficiency Shared Task.

Our key findings show that:

1. It is possible to prune entire nodes from feed-forward layers early during training, resulting in Pareto optimal architectures (quality vs speed). Similarly, pruning entire heads on the top of it results in even faster models.
2. Different language pairs exhibit similar structural sparsity patterns.
3. Pruning during training matches, and sometimes outperforms, retraining the pruned model from scratch.
4. Among the English→German Pareto optimal models, the notable examples include a model with a 6-layered decoder being 34% faster at the cost of 0.2 BLEU and a model with 12-1 encoder-decoder ratio gaining additional 51% speed-up costing 0.3 BLEU.
5. This type of pruning combined with quantisation gives a significant speed boost. Our models are 1.9–2.7× faster at the cost of 0.9–1.7 BLEU.

2 Related work

Extensive research to reduce workload, compress and speed-up neural machine translation models includes methods such as knowledge distillation (Kim and Rush, 2016), quantisation (Quinn and Ballesteros, 2018; Aji and Heafield, 2020), layer approximation (Kim et al., 2019b) and pruning. For the best results, they can be stacked together to train the efficient state-of-the-art model.

In their analysis, Dalvi et al. (2020) claim that 85% of transformer neurons are redundant across the network. Using transfer learning, they find the minimal set of neurons that achieve optimum performance given the task. However, that method requires a fully pretrained model to perform a brute-force search on it, making overall training time too long.

Pruning techniques are usually split into two groups: unstructured and structured. Unstructured removes individual coefficients. It is straightforward to apply and yields good quality results simultaneously, which makes it popular. Unstructured magnitude pruning, while successfully applied to NMT (See et al., 2016), often needs retraining to recover from quality damage. Moreover, it also requires an efficient matrix multiplication routine to get any speed-up besides size compression. The latest research on combining lottery ticket hypothesis with other methods (Brix et al., 2020) sparsified NMT models by 70 to 90% while losing between 0.6 to 3.3 BLEU points in quality. They used a sparse matrix representation for compression but did not report any speed gains.

On the other hand, structured pruning removes whole layers or groups of parameters, such as blocks (Narang et al., 2017), which makes it easier to optimise on hardware via a special block-sparse matrix multiply (Gray et al., 2017). We apply block sparsity, but the blocks are entire rows or columns so that the usual dense matrix multiply can be used with less overhead. Another line of work prunes entire attention heads from a model (Voita et al., 2019; Behnke and Heafield, 2020), which we also explore in our approach.

Yao et al. (2019) combine unstructured sparsity with a light structure that aims to balance parallel workloads. They introduce a specialised kernel for their structured sparsity. Our workloads are easier to balance because they retain density. The idea that different levels of coarseness can be combined may also extend to prune coefficients, rows, columns, and layers simultaneously in future work.

Wang et al. (2020c) parametrise weight matrices with low-rank factorisations and remove rank-1 components during training, which is said to better preserve linear transformation of uncompressed matrices. They report compressing a Transformer-XL language model by 90% with 1.6× speed-up during inference. Low-rank approximations preserve density, albeit at the cost of doing serial ma-

trix multiplications.

Fan et al. (2019) explored a structured dropout that allows users to prune models for inference. Unfortunately they fail short on NMT experiments. They call WMT14 en-de a 'competitive benchmark' which it has not been for many years. Most problematically, they use tokenised BLEU, which has been noted to be harmful and gives false 'boost' of multiple points on tokenised data. They do not specify the tokeniser or script they use either. Again, there do not include any report on speed or model sizes despite claiming to have much smaller models as a result.

Dodge et al. (2019) used group lasso to sparsify a variant of RNN for text classification, which is an easier task to learn than NMT. They have to train until convergence twice, which we avoid. They provide no speed or model size analysis, suggesting that there is no improvement or proper implementation.

Group lasso has also been previously used by Wuebker et al. (2018) to compress the delta between a base model and a domain adapted version of the model. They still have to run a full-size model in inference, so they have no overall speed gain. They also have to store the full base model; compression only refers to the delta. In contrast, our work makes the base model faster and smaller. The different goals also mean different groups: they focused on embeddings that update in domain adaptation while we focus on costly parts of the architecture.

Though we use the same algorithm of group lasso, our method differs in several ways from (Murray et al., 2019). We prune submatrices in addition to rows and columns, though experiments on just rows and columns show better performance than theirs. They pruned only feedforward layers; we see more speed-up from feedforward layers and additionally prune attention. Finally, we use the normal Adam optimiser (Kingma and Ba, 2014) instead of proximal gradient descent (Parikh and Boyd, 2014). Empirically, we find turning regularisation off after some training is important to quality. Overall, we achieve a much better trade-off between quality and speed/compression.

Most of the methods above need either tuning or retraining, often multiple times. They are usually treated as techniques to compress already existing models. Still, there are ongoing research efforts on training a reduced model from start to finish in one go. For example, Golub et al. (2018) pruned weights with the lowest total accumulated gradients and reduced the memory footprint to allow training much larger models than possible on available hardware. Some methods prune immediately after initialisation, in either unstructured (Lee et al., 2019) or structured (Wang et al., 2020a) way. Our method is orthogonal and is integrated into a training scheme instead.

Using regularisation to sparsify groups of parameters was introduced by Yuan and Lin (2006) and has been since then built upon in the machine learning field (Scardapane et al., 2017; Wen et al., 2016). In this paper, we use group lasso in its simplest form to achieve structural sparsity in transformer layers, focusing on inference speed of machine translation.

3 Methodology

To allow regularisation to remove parameters structurally, we need to define how we group parameters. Depending on which matrix it is, we treat parameters in its rows, columns or heads as one entity to be pruned together. Thus, we apply a group lasso over them. A bias term, if necessary, is treated as a part of regularised groups. We want such a sparsity pattern to emerge early into training so that there is no need to retrain or tune it later.

Group lasso regularisation

Given a matrix w split into non-overlapping groups of parameters G, the group lasso is defined as:

$$R(w) = \sum_{g=1}^{G} \gamma \left\| w_g \right\|_2 = \sum_{g=1}^{G} \gamma \sqrt{\sum_{j=1}^{|G_g|} \left(w_g^j \right)^2}. \quad (1)$$

This penalty term applies L_2 norm over the parameters in each group to force them to go towards 0 together, with L_1 on top of it to enforce overall sparsification. γ is a scalar that orthonormalises groups of different sizes (Simon and Tibshirani, 2012), scaling by the number of elements in a group $\sqrt{d_g}$. If regularising only rows and columns, all groups are of the same size. However, in later experiments, we also regularise whole attention heads alongside individual feedforward connections.

In the end, the penalty for each layer is added to the cost function and scaled by λ and averaged

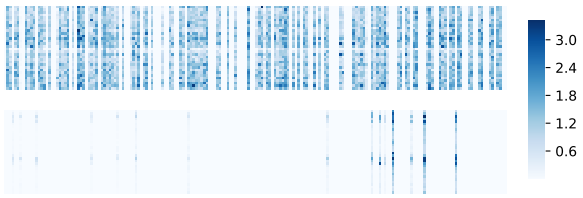


Figure 1: An example of block-sparse matrices in the first layer of decoder (top) and encoder (bottom) pruned by group lasso regularisation.

over words in a batch along with cross-entropy:

$$\frac{1}{|B|}(\sum_{x \in B} CE(x) + \lambda * \sum_{w \in W} R(w)) \,. \quad (2)$$

Initially we experimented with 8x8 blocks in group lasso. However, this approach removed entire rows and columns that correspond to pruning connections. Figure 1 shows an example of this effect on parameter matrices. When an entire row or column is zero, it can be deleted to form a smaller dense matrix. If an input connection is ignored and not used elsewhere, it can be removed from the upstream matrix. If an output connection is constant, the constant can be folded into downstream bias. These optimisations are typically discovered automatically by regularisation. We mainly focus on pruning feedforward layers, but later include experiments that prune attention heads as well.

4 Setup

4.1 Data & architectures

We concentrate on three language pairs: English→German, Spanish→English and Estonian→English. We use knowledge distillation (Kim and Rush, 2016) under teacher-student regime.

In English→German, we follow the Workshop on Neural Generation and Translation 2020 Efficiency shared task (WNGT2020) [1] under the WMT 2019 data condition (Barrault et al., 2019). As a teacher, we use a WMT 2019 system submitted by Microsoft to news translation task (Junczys-Dowmunt, 2019). It is an ensemble of four deep transformer-big models (Vaswani et al., 2017) with 12 layers in encoder and decoder, embedding size of 1024 and feedforward size 4096 and 8 attention heads. For Spanish→English and Estonian→English, we use teachers provided by the Bergamot,[2] which is an ensemble of two similar architectures but with 6 layers instead.

We start with strong student baselines, which are already very small and fast, closely following the latest trends set by WNGT2020.

Students for all language pairs have an embedding dimension of 256 and feedforward of 1536, based on "tiny" architecture from Kim et al. (2019a). The attention has 8 heads in each layer except for decoder self-attention, which is replaced by a faster SSRU (Simpler Simple Recurrent Unit) (Kim et al., 2019b). The models use a shared vocabulary of 32,000 subword units generated by SentencePiece (Kudo and Richardson, 2018) and translate using shortlists of top 50 words.

We tried different configurations of layers to investigate trade-offs between them and potential bottlenecks. We describe each architecture by layer number in encoder and decoder and whether the decoder layers are tied. Thus, we investigate the following architectures (chronologically): "6-2tied", "6-6", "12-1" and "6-2".

Other training hyperparameters were Marian defaults for training a transformer base model.[3] We used dynamic batching, filling a 10GB workspace on each of 4 GPUs, resulting in about 71,000 words per batch in a "6-2tied" student and about 46,000 words per batch in a "6-6" student. As is more effective in the teacher-student regime, we did not use dropout or label smoothing. We use the Adam optimiser (Kingma and Ba, 2014).

The English→German models were trained on 13M sentences of available parallel data, using the concatenated English-German WMT testsets from 2016–2018 as a development set. The Spanish→English students were trained on 242M sentences which included about 15M of mixed forward- and backtranslations. We used a WMT13 testset for development. Estonian→English students were trained on 132M sentences which included about 30M of mixed forward- and backtranslations, and WMT18/dev was used for development.

We trained and decoded all our models using the Marian NMT toolkit (Junczys-Dowmunt et al., 2018). We evaluate quality and speed on 1 CPU core. In order to expand beyond BLEU and incentivise others to do the same, we additionally

[1] https://sites.google.com/view/wngt20

[2] https://github.com/browsermt/students

[3] Available via `--task transformer-base`.

evaluate with chrF (Popović, 2015) and COMET[4] (Rei et al., 2020) as well. We use SacreBLEU (Post, 2018) for BLEU and chrF. Training progressed until BLEU stopped improving for 20 consecutive validations. The checkpoint with the highest BLEU score was selected.

4.2 Training regime

Our training regime for all of ours models has three phases:

1. Pretrain for 25k batches.
2. Train with a regulariser for 250k batches.
3. Remove rows/columns with a sum less than $1e-5$, collapse a model and then train without regularisation until convergence.

It is well known that initial transformer training is problematic and sensitive to model hyperparameters (Nguyen and Salazar, 2019; Aji et al., 2019; Liu et al., 2020). A transformer starts training with 1–2 BLEU and quickly jumps over to 15–30 or more within a short training period, then slows down. Thus, we start pruning after BLEU improvement slowed down to be less than 1 BLEU point in a single checkpoint. This way, we avoid any potential damage to a model during the critical initial period. In this case, we pretrain for 25k batches.

Next, we had to decide how long to regularise our model to achieve a good trade-off between quality and sparsity levels. We started with regularising a model until convergence. As shown in Fig. 2, most of the parameters are already pruned in the first half of the training. Since students require significantly more updates to train than standard models, we want to give a model enough time to sparsify and converge without any constraints. For this reason, we split an average training time into two halves: the first with pruning, the second without it with normal convergence. We found that switching the regulariser off at some point is less aggressive and allows a model to recover some of its lost quality. We chose 250k updates as a pivot as it is about halfway to when the model has begun stalling in Fig. 2.

After each step, we copy the latest checkpoint and start a fresh training round. Thus, all training hyperparameters are reset. We checked and found no additional advantage to our baselines by refreshing learning rate scheduling or Adam optimiser. We do so to avoid partially retraining the same settings during our development phase, but Brix et al. (2020) found it beneficial in their pruning scheme.

[4]We used the default 'wmt20-comet-da' metric model.

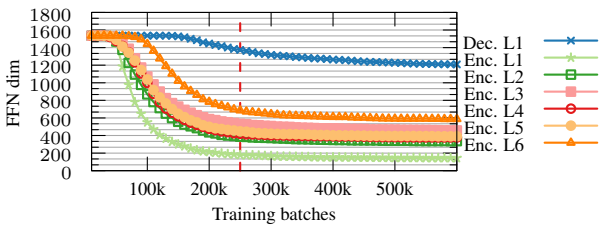


Figure 2: An example of pruning FFN layers in an English→German tied model ($\lambda = 0.5$). About "halfway" through, most parameters are already removed.

Reg. $\lambda \longrightarrow$	Base	0.3	0.4	0.5	0.7	1.0
BLEU	37.2	37.8	37.4	36.8	36.5	36.1
chrF	63.3	63.8	63.4	63.1	62.8	62.5
COMET	49.7	51.1	50.4	48.9	47.2	46.0
FFN sparsity	0%	45%	63%	73%	85%	92%
Size (MB)	61	51	47	45	43	41
WPS	2404	2613	2748	3067	3215	3420
Speed-up	1.00	1.09	1.14	1.28	1.34	1.42

(a) With pruned encoder + decoder.

Reg. $\lambda \longrightarrow$	Base	0.3	0.4	0.5	0.7	1.0
BLEU	37.2	37.6	37.3	36.8	36.8	36.4
chrF	63.3	63.4	63.4	63.1	62.9	62.8
COMET	49.7	50.4	49.8	49.8	48.3	47.8
FFN sparsity	0%	48%	64%	72%	79%	82%
Size (MB)	61	50	47	45	44	43
Words per sec	2404	2748	2916	2929	3054	3096
Speed-up	1.00	1.14	1.21	1.22	1.27	1.29

(b) With pruned encoder.

Table 1: The evaluation of English→German "6-2tied" students pruned using group lasso.

5 Experiments

5.1 Pruning "6-2tied" models (English→German)

We begin our experiments with the state-of-the-art English→German student with a tied decoder (Bogoychev et al., 2020). This is their fastest architecture and we want to investigate how much further it can be pushed in that regard. In terms what is a typical difference in inference speed, a tiny distilled model is usually at least 20× faster than its teacher (Germann et al., 2020).

We investigate two scenarios: pruning both the encoder and decoder (Tab. 1a) or pruning only the encoder (Tab. 1b). The models were trained with the regularisation term $\lambda \in \{0.3, 0.4, 0.5, 0.7, 1.0\}$.

Since there is only one layer's worth of decoder parameters, the regularisation is reluctant to re-

Reg. $\lambda \longrightarrow$		Base	0.3	0.4	0.5	0.7	1.0
BLEU	Pruned	37.2	37.8	37.4	36.8	36.5	36.1
	Reinit	-	37.1	36.5	36.5	36.1	35.3
chrF	Pruned	63.3	63.8	63.4	63.1	62.8	62.5
	Reinit	-	63.2	62.8	62.7	62.3	62.0
COMET	Pruned	49.7	51.1	50.4	48.9	47.2	46.0
	Reinit	-	48.8	47.3	47.3	45.7	42.7

Table 2: The average BLEU of English→German "6-2tied" students with pruned encoder and decoder (*Pruned*), compared to the same architecture trained from scratch (*Reinit*).

Reg. $\lambda \longrightarrow$	Base	0.1	0.15	0.2	0.3	0.4	0.5	1.0
Enc. L1	1536	821	453	259	100	56	35	12
Enc. L2	1536	932	506	327	166	93	65	22
Enc. L3	1536	1009	502	298	129	79	47	10
Enc. L4	1536	1213	663	384	147	70	41	11
Enc. L5	1536	1149	646	392	186	119	87	16
Enc. L6	1536	1366	919	610	322	208	148	43
Dec. L1	1536	542	231	121	43	15	8	1
Dec. L2	1536	836	435	259	108	50	27	4
Dec. L3	1536	448	227	129	49	26	16	3
Dec. L4	1536	1064	623	422	229	142	111	25
Dec. L5	1536	1528	1260	876	450	276	198	49
Dec. L6	1536	1536	1536	1536	1517	1216	835	178
BLEU	38.5	38.7	38.6	38.3	37.7	37.6	37.4	36.8
chrF	64.2	64.5	64.4	64.1	63.7	63.6	63.5	63.1
COMET	54.8	55.7	54.7	53.8	52.9	52.8	51.9	49.6
FFN sparsity	0%	31%	55%	69%	81%	87%	91%	98%
Size (MB)	83	72	63	58	54	52	50	48
WPS	1225	1377	1543	1639	1741	1827	1867	1976
Speed-up	1.00	1.12	1.26	1.34	1.42	1.49	1.52	1.61

Table 3: The evaluation of English→German "6-6" students pruned with group lasso on 1 CPU core, with the distribution of parameters left in each layer.

move any parameters from it (Tab. 1a). A similar effect was observed by Behnke and Heafield (2020).

Because only the encoder was pruned, the speed-up is relatively small. Still, we successfully prune from half up to two-thirds of feedforward parameters with ± 0.2 BLEU change with 9–14% faster inference. In the most extreme case, it gains 42% speed-up at the cost of 1.5 BLEU.

To investigate whether this pruning just found a new type of architecture structure, we reinitialise and retrain the smaller pruned models from scratch with reduced dimensions. As seen in Tab. 2, the same models achieve noticeably worse translation quality when trained from the get-go in comparison to careful pruning.

Next, we concentrate on pruning encoder only (Tab. 1b). The model with about 50% of feedforward parameters removed is 14% faster with no change to the overall quality. The most aggressive pruning removes almost all feedforward layers in the encoder at the loss of 1.2 BLEU.

Reg. $\lambda \longrightarrow$		Base	0.1	0.15	0.2	0.3	0.4	0.5	1.0
BLEU	Pruned	38.5	38.7	38.6	38.3	37.7	37.6	37.4	36.8
	Reinit	-	38.1	38.0	37.6	37.3	37.2	37.0	36.6
chrF	Pruned	64.2	64.5	64.4	64.1	63.7	63.6	63.5	63.1
	Reinit	-	64.0	63.9	63.7	63.6	63.4	63.3	63.0
COMET	Pruned	54.8	55.7	54.7	53.8	52.9	52.8	51.9	49.6
	Reinit	-	53.8	53.3	52.3	51.5	51.4	49.7	49.1

Table 4: The evaluation of English→German "6-6" students with pruned both encoder and decoder (*Pruned*), compared to the same architecture trained from scratch (*Reinit*).

Reg. $\lambda \rightarrow$	Base	0.1	0.15	0.2	0.3	0.4	0.5	1.0
BLEU	37.3	36.9	36.8	36.6	36.3	36.3	36.1	35.9
chrF	62.6	62.4	62.3	62.3	62.0	62.0	61.9	61.8
COMET	58.1	57.3	56.8	56.2	55.1	55.4	54.6	54.3
FFN sparsity	0%	48%	67%	77%	86%	91%	94%	98%
Size (MB)	83	59	66	55	52	50	49	48
WPS	1407	1655	1811	1891	2017	2071	2112	2204
Speed-up	1.00	1.18	1.29	1.34	1.43	1.47	1.50	1.57

Table 5: The evaluation of Spanish→English "6-6" students pruned with group lasso on 1 CPU core averaged over WMT12–13.

In both cases, only one-third of feedforward parameters is required to perform within a small margin of BLEU loss (-0.2 to -0.3). Removing more than that results in progressively worse quality.

5.2 Pruning "6-6" models (English→German, Spanish→English)

The models with "6-6" architecture were trained with $\lambda = \{0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 1.0\}$ pruning both encoder and decoder. The results are presented in Tab. 4. Additionally, we show the number of remaining rows/columns left in each layer, along with sparsity and inference speed-up.

The pruned models behave similarly to the smaller models in Tab. 1. The regularised models are of a better translation quality than the same architectures trained from scratch (Tab. 6). Similarly, it is possible to remove two-thirds of all feedforward parameters with -0.2 BLEU and +34% speed-up. Pruning more than that causes a notable step down in quality, which may not be worth aiming for since the "6-2tied" architectures outperform that loss. The sparsity pattern follows an ascending trend in both encoder and decoder layers.

We repeat the experiments but this time with Spanish→English using the same "6-6" architecture. The models were trained with regularisation $\lambda = \{0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 1.0\}$. The results are presented in Tab 5.

Reg. $\lambda \longrightarrow$		Base	0.1	0.15	0.2	0.3	0.4	0.5	1.0
BLEU	Pruned	37.3	36.9	36.8	36.6	36.3	36.3	36.1	35.9
	Reinit	-	37.0	36.7	36.5	36.3	36.2	36.2	35.8
chrF	Pruned	62.6	62.4	62.3	62.3	62.0	62.0	61.9	61.8
	Reinit	-	62.5	62.2	62.1	62.0	61.9	61.9	61.7
COMET	Pruned	58.1	57.3	56.8	56.2	55.1	55.4	54.6	54.3
	Reinit	-	57.3	56.7	55.9	55.1	54.6	54.5	53.0

Table 6: The evaluation of Spanish→English "6-6" students with pruned encoder and decoder (*Pruned*), compared to the same architecture trained from scratch (*Reinit*) averaged over WMT12–13.

Reg. $\lambda \longrightarrow$	Base	0.2	0.3	0.4	0.5	0.7
BLEU	38.2	37.9	37.3	37.0	37.0	36.6
chrF	63.9	63.6	63.2	62.9	62.9	62.5
COMET	49.5	49.6	48.2	46.5	45.5	44.6
Att. sparsity	0%	48%	56%	58%	57%	59%
FFN sparsity	0%	63%	76%	81%	84%	87%
Size (MB)	85	54	48	45	44	43
WPS	1930	2918	3029	3430	3446	3485
Speed-up	1.00	1.51	1.57	1.78	1.79	1.81

Table 7: The evaluation of English→German "12-1" students pruned with group lasso on 1 CPU core.

The only differences between German and Spanish experiments are the languages involved and the scale of the training data: Spanish students trained on a 19× larger corpus. Experiments of such scale are still widely unexplored in machine translation, raising the question of whether known methods are beneficial in real-life scenarios. Most pruning papers use English→German models under WMT14 constraints (Bojar et al.), which is only 4.5M sentences (See et al., 2016; Brix et al., 2020; Hsu et al., 2020), sometimes branching into different languages such as Russian or French in a similar scope (Voita et al., 2019; Kasai et al., 2020). For that reason, we sampled 13M from 242M sentences (the same amount as English→German) and repeated the experiments. In the end, we came to similar conclusions, meaning that the Spanish subpar results are not related to architecture or data size. The reinitialised models (Tab. 6) on full dataset achieve comparable quality to their pruned counterparts. We conclude that in some cases, structural pruning serves as an architecture search method to find Pareto optimal quality-speed trade-off.

5.3 Pruning "12-1" models (English→German)

Kasai et al. (2020) argues that shifting layers from decoder to encoder makes a model much faster at almost no cost in translation quality. Their experiments have shown that 12-1 encoder-decoder layer proportions perform as good as 6-6. Pruning an already reduced decoder may cause a bottleneck that damages quality too much. However, if we shift most of the workload into an encoder, we can focus on pruning it exclusively.

This time we prune attention layers as well. Pruning attention structurally is more tricky — you cannot remove individual connections easily due to how matrix multiplications perform their routine. The only option is to remove respective heads or an entire layer. To keep it simple, we regularise individual connections and remove an entire attention head if at least half of its connections are dead (its rows/columns $< 1e-5$). The results are in Tab. 7, with an extended version of it in the appendix.

In terms of quality and speed-up, it outperforms other models presented so far. This type of pruning was not aggressive on attention, preferring to prune feedforward layers instead, indicating that attention connections perform more critical work in a model. At the small cost of 0.3 BLEU, the model is 51% faster than the baseline.

		Baseline		rc+heads		rc+rc	
		FFN	Heads	FFN	Heads	FFN	Heads
Encoder 1		1536	8	579	0	210	7
Encoder 2		1536	8	793	1	552	1
Encoder 3		1536	8	959	0	712	3
Encoder 4		1536	8	913	0	459	6
Encoder 5		1536	8	1212	3	708	4
Encoder 6		1536	8	1523	2	1033	8
Decoder 1		1536	8	1536	2	1535	7
Decoder 2		1536	8	1536	7	1536	8
BLEU	Pruned	31.5		29.8		30.4	
	Reinit	-		28.5		30.3	
chrF	Pruned	58.4		57.0		57.6	
	Reinit			56.0		57.5	
COMET	Pruned	54.8		49.9		53.0	
	Reinit	-		46.8		50.7	
Time		21.57		15.16		18.9	
WPS		1414		2012		1614	
Speed-up		1.00		1.42		1.14	

Table 8: The WMT18 testset evaluation of Estonian→English "6-2" students pruned with group lasso on 1 CPU core with the same architectures trained from scratch (*Reinit*).

5.4 Pruning "6-2" models with head lasso (Estonian→English)

Finally, we train Estonian→English models, pruning both feedforward and attention layers across the whole model. We do not sweep parameters, choosing $\lambda = 0.3$. The results are in Tab. 8.

This time we try two options:

- regularising individual connections and then removing heads with more than half of con-

	BLEU		COMET		Sparsity		
	WMT20	WMT21	WMT20	WMT21	Att.	FFN	Speed (s)
12-1.tiny	36.1	27.6	48.2	41.9	0%	0%	19.2
+ head-lasso pruning	34.7	27.0	42.9	38.8	3%	75%	14.5
+ 8bit quantisation	33.9	26.2	38.8	33.6	3%	75%	9.3
+ 8bit finetuning	34.1	26.7	39.8	33.0	3%	75%	9.3
+ rowcol-lasso pruning	33.8	26.3	39.3	34.2	68%	73%	11.6
+ 8bit quantisation	32.9	25.6	33.7	28.7	68%	73%	6.9
+ 8bit finetuning	32.9	26.0	35.7	31.3	68%	73%	7.1
12-1.micro	35.4	27.6	46.2	40.2	0%	0%	17.1
+ head-lasso pruning	34.6	26.7	43.0	35.4	3%	72%	14.1
+ 8bit quantisation	33.4	26.0	36.7	31.2	3%	72%	9.2
+ 8bit finetuning	33.7	26.5	38.3	33.3	3%	72%	9.2
+ rowcol-lasso pruning	34.3	26.4	40.7	35.1	60%	59%	12.0
+ 8bit quantisation	32.7	25.5	34.2	29.1	60%	59%	7.5
+ 8bit finetuning	33.3	25.9	35.2	30.5	60%	59%	7.5

Table 9: 8-bit model performance. BLEU score is calculated from WMT20. Speed is measured on a single core CPU with a mini-batch of 32. We experimented with two types of pruning. Head pruning removes entire heads. Row and column pruning removes entire rows or columns of matrices, resulting in a smaller matrix.

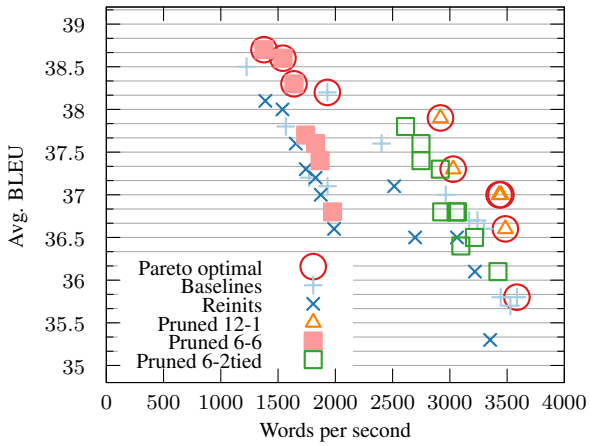


Figure 3: Pareto trade-off between average translation quality and average translation time for English→German students of different architectures.

nections inactive (rc+rc = rows/columns in FFN and attention both)

- regularising entire heads with group lasso (rc+heads = rows/columns in FFN, heads in attention)

Due to how the penalty is scaled with γ in Eq. 1, the regularisation of entire heads is much more aggressive towards them, removing some layers entirely, which we skip during inference. Both pruning methods perform within a -0.1 to -0.3 BLEU difference compared to the same architecture trained from scratch. However, despite only 0.1 BLEU difference, the same model loses 2.3 COMET points, further validating the fact that training from scratch is subpar. Those results show that there is a potential in regularising larger structures and even entire layers as a way of architecture searching. We leave the improvement of the method for future work.

6 Pareto trade-off (English→German)

In this section, we look at the Pareto trade-off between the translation quality and speed for all our English→German models (Fig. 3). To be fair in our comparison, we trained several simpler baselines with uniformly smaller feedforward dimensions set to $\{768, 384, 192, 96\}$. For"12-1" we additionally set heads per layer to 4 to roughly reflect sparsity percentages of pruned models.

We pit against each other the said baselines, the pruned models and their reinitialised counterparts. Naturally, the models with 6 decoder layers are slower but of a higher quality. However, it is better to switch to fewer decoder layers than to prune too far. Our experiments on "12-1" architecture show that its pruned models outperform all others (including all simpler baselines), being a leader in the Pareto frontier.

7 WMT2021 Efficiency Shared Task

To put our method to the final test, we participated in WMT2021 Efficiency Task[5] (Behnke et al., 2021). Under the task constraints, we trained, pruned and quantised 12–1.tiny and 12–1.micro architectures. We tried two pruning settings, following the directions set in Sect. 5.3 and 5.4: *rowcol-*

[5] http://www.statmt.org/wmt21/efficiency-task.html

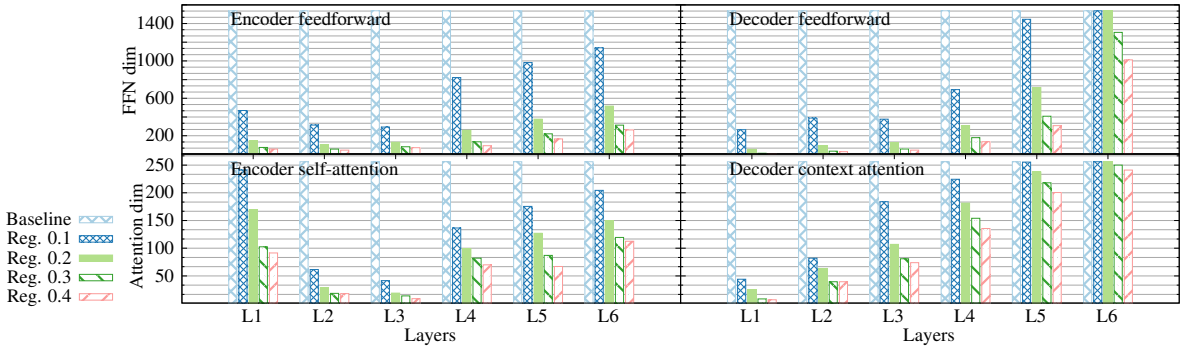


Figure 4: The distribution of feedforward and attention connections in Spanish→English "6-6" pruned students.

lasso and *head-lasso*. Both prune feedforward and attention layers in the encoder. *rowcol-lasso* regularised individual connections and removed an entire attention head if at least half of its connections are dead. *head-lasso* applied lasso to a whole head submatrix. Due to the scale of the task, we had no opportunity to grid-search for the best pruning hyperparameters, thus the experiments are as close to 'out-of-the-box' usage as they can be. We used $\lambda = 0.5$ for both methods. The models were pretrained for 50k updates and regularised for 150k, after which the models were sliced and trained until convergence. The results are presented in Tab. 9.

head-lasso left attention layers almost completely unpruned, focusing on removing connections from feedforward layers instead. *rowcol-lasso* was much more aggressive in both layers at the cost of quality. To further optimise the models, they were quantised to 8bit. However, we observe that the smaller a model is, the larger the quality drop after its quantisation. Additional finetuning allows us to recover at least partially from the quantisation damage. Evaluating on the latest testset WMT21, our pruned models are 1.2–1.7× faster at the cost of 0.6–1.3 BLEU. With quantisation, those models are 1.9–2.7× faster losing 0.9–1.7 BLEU in comparison to the unpruned and unquantised baselines.

8 Analysis

To analyse sparse architecture patterns, we experiment with Spanish→English models. We prune individual connections in all attention and feedforward layers. In Fig. 4, we present the distribution of remaining parameters for the baseline and the models regularised with $\lambda = \{0.1, 0.2, 0.3, 0.4\}$.

Since the decoder self-attention is replaced with SSRU (Kim et al., 2019b), we only show two "pairs" of parameters: encoder self-attention and decoder context attention, with their feedforward counterparts.

Both encoder layer types follow a similar sparsity pattern making a "U-shape", with the second and third ones being the most aggressively pruned. On the other hand, the decoder parameters are pruned less and less with each subsequent layer. This arrangement of parameters is identical to that exhibited by pruned attention heads in Behnke and Heafield (2020). In that paper, the attention in the encoder also prunes middle layers, and the context attention retains more heads in further layers. It strongly indicates that the decoder prefers to attend to itself first and confront context later.

The Estonian architectures, in which we pruned entire attention heads, exhibit a roughly similar structure. For us, this is a strong signal that structural pruning with its architecture search may have a broader generalisation.

9 Conclusions

This paper investigated the structural pruning of a transformer incorporated into a typical training routine. We focused on shredding nodes in feedforward layers and whole attention heads as training progresses. Our experiments on knowledge-distilled models with deep and shallow decoders have shown that this type of pruning leads to Pareto optimal architectures in quality and speed. Moreover, it converges in just one "pass" like a baseline since there is no need to repeat an entire or a part of the training. The resulting sparsity patterns are similar across different languages, with the first and middle layers being the most prioritised during pruning. On the other hand, our experiments on pruning both feedforward and attention layers reveal that some of them, such as the last context attention layer, distinctively avoid being pruned.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council (grant EP/L01503X/1), EPSRC Centre for Doctoral Training in Pervasive Parallelism at the University of Edinburgh, School of Informatics.

This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/P020259/1.

References

Alham Fikri Aji and Kenneth Heafield. 2020. Compressing neural machine translation models with 4-bit precision. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 35–42, Online. Association for Computational Linguistics.

Alham Fikri Aji, Kenneth Heafield, and Nikolay Bogoychev. 2019. Combining global sparse gradients with local gradients in distributed neural network training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3626–3631, Hong Kong, China. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Heafield Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap, and Roman Grundkiewicz. 2021. Efficient machine translation with model pruning and quantization. In *Proceedings of the Six Conference on Machine Translation*, Online. Association for Computational Linguistics.

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, Online. Association for Computational Linguistics.

Nikolay Bogoychev, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 machine translation efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. 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.

Christopher Brix, Parnia Bahar, and Hermann Ney. 2020. Successfully applying the stabilized lottery ticket hypothesis to the transformer architecture. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3909–3915, Online. Association for Computational Linguistics.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Online. Association for Computational Linguistics.

Jesse Dodge, Roy Schwartz, Hao Peng, and Noah A. Smith. 2019. RNN architecture learning with sparse regularization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1179–1184, Hong Kong, China. Association for Computational Linguistics.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *CoRR*, abs/1909.11556.

Trevor Gale, Matei Zaharia, Cliff Young, and Erich Elsen. 2020. Sparse gpu kernels for deep learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Ulrich Germann, Roman Grundkiewicz, Martin Popel, Radina Dobreva, Nikolay Bogoychev, and Kenneth Heafield. 2020. Speed-optimized, compact student models that distill knowledge from a larger teacher model: the UEDIN-CUNI submission to the WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 191–196, Online. Association for Computational Linguistics.

Maximilian Golub, Guy Lemieux, and Mieszko Lis. 2018. Dropback: Continuous pruning during training. *CoRR*, abs/1806.06949.

Scott Gray, Alec Radford, and Diederik P Kingma. 2017. Gpu kernels for block-sparse weights.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Yi-Te Hsu, Sarthak Garg, Yi-Hsiu Liao, and Ilya Chatsviorkin. 2020. Efficient inference for neural machine translation. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 48–53, Online. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. 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.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019a. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019b. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Jason Lee, Raphael Shu, and Kyunghyun Cho. 2020. Iterative refinement in the continuous space for non-autoregressive neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1006–1015, Online. Association for Computational Linguistics.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2019. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763, Online. Association for Computational Linguistics.

Kenton Murray, Brian DuSell, and David Chiang. 2019. Efficiency through auto-sizing: Notre Dame NLP's submission to the WNGT 2019 efficiency task. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 297–301, Hong Kong. Association for Computational Linguistics.

Sharan Narang, Eric Undersander, and Gregory F. Diamos. 2017. Block-sparse recurrent neural networks. *CoRR*, abs/1711.02782.

Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. *CoRR*, abs/1910.05895.

Neal Parikh and Stephen Boyd. 2014. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239.

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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Jerry Quinn and Miguel Ballesteros. 2018. Pieces of eight: 8-bit neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 114–120, New Orleans - Louisiana. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's bert: Smaller and faster transformer models.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. 2017. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of neural machine translation models via pruning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 291–301, Berlin, Germany. Association for Computational Linguistics.

Noah Simon and Robert Tibshirani. 2012. Standardization and the group lasso penalty. *Statistica Sinica*, 22(3):983.

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.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. 2020a. Picking winning tickets before training by preserving gradient flow.

Yong Wang, Longyue Wang, Victor Li, and Zhaopeng Tu. 2020b. On the sparsity of neural machine translation models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1060–1066, Online. Association for Computational Linguistics.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020c. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2082–2090, Red Hook, NY, USA. Curran Associates Inc.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 881–886, Brussels, Belgium. Association for Computational Linguistics.

Zhuliang Yao, Shijie Cao, Wencong Xiao, Chen Zhang, and Lanshun Nie. 2019. Balanced sparsity for efficient dnn inference on gpu. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:5676–5683.

Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 68:49–67.

10 Appendix

Reg. $\lambda \longrightarrow$	Base	0.3	0.4	0.5	0.7	1.0
Enc. L1	1536	330	179	113	52	29
Enc. L2	1536	633	387	266	135	57
Enc. L3	1536	882	533	351	175	82
Enc. L4	1536	738	420	269	137	66
Enc. L5	1536	720	447	309	179	100
Enc. L6	1536	1079	686	488	293	166
Dec. L1	1536	1534	1373	1072	626	339
BLEU	37.2	37.8	37.4	36.8	36.5	36.1
chrF	63.3	63.8	63.4	63.1	62.8	62.5
COMET	49.7	51.1	50.4	48.9	47.2	46.0
FFN sparsity	0%	45%	63%	73%	85%	92%
Size (MB)	61	51	47	45	43	41
WPS	2404	2613	2748	3067	3215	3420
Speed-up	1.00	1.09	1.14	1.28	1.34	1.42

(a) With pruned encoder + decoder.

Reg. $\lambda \longrightarrow$	Base	0.3	0.4	0.5	0.7	1.0
Enc. L1	1536	310	164	98	42	24
Enc. L2	1536	597	372	239	104	50
Enc. L3	1536	831	480	302	143	59
Enc. L4	1536	692	376	234	115	48
Enc. L5	1536	664	400	253	142	73
Enc. L6	1536	948	575	399	209	112
Dec. L1	1536	1536	1536	1536	1536	1536
BLEU	37.2	37.6	37.3	36.8	36.8	36.4
chrF	63.3	63.4	63.4	63.1	62.9	62.8
COMET	49.7	50.4	49.8	49.8	48.3	47.8
FFN sparsity	0%	48%	64%	72%	79%	82%
Size (MB)	61	50	47	45	44	43
Words per sec	2404	2748	2916	2929	3054	3096
Speed-up	1.00	1.14	1.21	1.22	1.27	1.29

(b) With pruned encoder.

Table 10: The evaluation of English→German "6-2tied" students pruned using group lasso on 1 CPU core.

Reg. $\lambda \rightarrow$	Base	0.1	0.15	0.2	0.3	0.4	0.5	1.0
Enc. L1	1536	563	258	137	52	30	24	12
Enc. L2	1536	454	236	156	73	47	31	15
Enc. L3	1536	421	221	135	73	47	37	19
Enc. L4	1536	799	368	197	96	52	34	16
Enc. L5	1536	999	565	350	189	114	83	32
Enc. L6	1536	1227	751	472	258	166	115	40
Dec. L1	1536	418	167	77	15	5	2	1
Dec. L2	1536	491	229	117	38	18	10	1
Dec. L3	1536	448	227	129	49	26	16	3
Dec. L4	1536	787	443	294	143	104	68	24
Dec. L5	1536	1475	1037	684	343	214	156	33
Dec. L6	1536	1536	1536	1533	1220	753	459	138
BLEU	37.3	36.9	36.8	36.6	36.3	36.3	36.1	35.9
chrF	62.6	62.4	62.3	62.3	62.0	62.0	61.9	61.8
COMET	58.1	57.3	56.8	56.2	55.1	55.4	54.6	54.3
FFN sparsity	0%	48%	67%	77%	86%	91%	94%	98%
Size (MB)	83	59	66	55	52	50	49	48
WPS	1407	1655	1811	1891	2017	2071	2112	2204
Speed-up	1.00	1.18	1.29	1.34	1.43	1.47	1.50	1.57

Table 11: The evaluation of Spanish→English 6–6 students pruned with group lasso on 1 CPU core.

Reg. $\lambda \longrightarrow$	Base	0.2	0.3	0.4	0.5	0.7
Enc. L1	1536	728	414	250	183	108
Enc. L2	1536	927	619	436	325	202
Enc. L3	1536	540	338	222	173	105
Enc. L4	1536	415	250	166	123	77
Enc. L5	1536	429	255	167	116	66
Enc. L6	1536	382	191	123	89	60
Enc. L7	1536	334	138	81	50	30
Enc. L8	1536	297	129	69	50	19
Enc. L9	1536	321	135	69	44	28
Enc. L10	1536	319	174	117	88	48
Enc. L11	1536	474	298	214	165	112
Enc. L12	1536	635	376	264	184	114
Dec. L1	1536	1536	1536	1536	1536	1536
Self att. L1	8	5	5	4	4	4
Self att. L2	8	3	2	2	2	2
Self att. L3	8	4	4	4	4	3
Self att. L4	8	4	3	3	3	3
Self att. L5	8	5	4	4	5	5
Self att. L6	8	4	4	4	3	3
Self att. L7	8	4	4	4	4	4
Self att. L8	8	3	1	1	1	1
Self att. L9	8	3	1	1	3	3
Self att. L10	8	3	2	1	1	1
Self att. L11	8	4	4	4	3	2
Self att. L12	8	4	4	4	4	4
Context att. L1	8	8	8	8	8	8
BLEU	38.2	37.9	37.3	37.0	37.0	36.6
chrF	63.9	63.6	63.2	62.9	62.9	62.5
COMET	49.5	49.6	48.2	46.5	45.5	44.6
Att. sparsity	0%	48%	56%	58%	57%	59%
FFN sparsity	0%	63%	76%	81%	84%	87%
Size (MB)	85	54	48	45	44	43
WPS	1930	2918	3029	3430	3446	3485
Speed-up	1	1.51	1.57	1.78	1.79	1.81

Table 12: The evaluation of English→German 12–1 students pruned with group lasso on 1 CPU core.

Phrase-level Active Learning for Neural Machine Translation

Junjie Hu*
University of Wisconsin-Madison
junjie.hu@wisc.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu

Abstract

Neural machine translation (NMT) is sensitive to domain shift. In this paper, we address this problem in an active learning setting where we can spend a given budget on translating in-domain data, and gradually fine-tune a pre-trained out-of-domain NMT model on the newly translated data. Existing active learning methods for NMT usually select sentences based on uncertainty scores, but these methods require costly translation of full sentences even when only one or two key phrases within the sentence are informative. To address this limitation, we re-examine previous work from the phrase-based machine translation (PBMT) era that selected not full sentences, but rather individual phrases. However, while incorporating these phrases into PBMT systems was relatively simple, it is less trivial for NMT systems, which need to be trained on full sequences to capture larger structural properties of sentences unique to the new domain. To overcome these hurdles, we propose to select *both* full sentences and individual phrases from unlabelled data in the new domain for routing to human translators. In a German-English translation task, our active learning approach achieves consistent improvements over uncertainty-based sentence selection methods, improving up to 1.2 BLEU score over strong active learning baselines.[1]

1 Introduction

Machine translation (MT) models are very sensitive to domain shift (Koehn and Knowles, 2017; Chu and Wang, 2018), and one typical way to address this problem is adding in-domain data to the MT training process (Luong and Manning, 2015; Chu et al., 2017). However, this data may not be available *a priori*, and hiring professional translators with knowledge of specific domains (such as medicine or law) is usually costly.

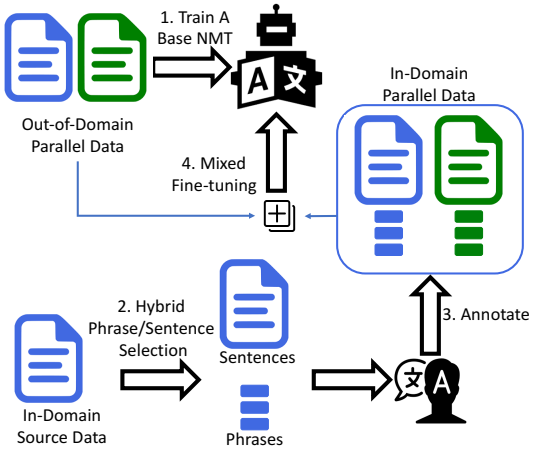


Figure 1: Overview of the active learning process

As a result, active learning approaches (Gangadharaiah et al., 2009; Haffari et al., 2009; Bloodgood and Callison-Burch, 2010) have been widely adopted to reduce the annotation cost by translating a smaller representative subset of the in-domain data, with the hope that models trained on this translated subset approximate those trained on a much larger labeled set. In general, active learning (AL) approaches iterate between two steps: *data selection/annotation*, and *model update*. With regards to data selection for machine translation, most existing works (Haffari et al., 2009; Peris and Casacuberta, 2018; Zeng et al., 2019) focus on selecting *sentences* that are most useful for training either phrase-based machine translation (PBMT) or neural machine translation (NMT) models.

However, even the most informative sentences inevitably involve segments that the MT system can already translate well, and asking the translator to also translate these segments is not cost-effective. There have been a few works used in conjunction with older PBMT models that ameliorate this problem through phrase-based selection techniques (Bloodgood and Callison-Burch, 2010; Daumé III and Jagarlamudi, 2011; Miura et al.,

*Work done at Carnegie Mellon University
[1]Code/data is released at https://github.com/JunjieHu/phrase-al-nmt.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1087–1099
November 10–11, 2021.

2016), which select only *individual phrases*, maximizing information gain. However, while these translated phrases can be easily integrated into PBMT by adding them to the existing phrase table, incorporating them into NMT models is less simple because NMT has no concept of a "phrase table" and must be trained on full sentences similar to those that must be translated.

In this paper, we propose a method for incorporating phrase-based active learning into NMT. Specifically, we first describe sentence-based and phrase-based selection strategies, then propose a hybrid strategy that combines both methods. We also describe several ways to incorporate this translated data into the training of NMT systems. We conduct experiments on German-English translation by adapting NMT models trained on WMT parallel data to the medicine and IT domains. Experimental results show that the hybrid selection strategy obtains more stable translation performance than either phrase-based or sentence-based selection strategy.

2 Problem Definition

In the setting of active learning for domain adaptation, we are given an out-of-domain labelled corpus $(x, y) \in \mathcal{L}$ and an in-domain unlabelled corpus $x \in \mathcal{U}$. We define a phrase as a contiguous sequence of words up to some length limit N, and denote a set of possible phrases in a sentence x by $\cup_{n\in[1,N]} n\text{-gram}(x)$, where we set $N = 4$ in all experiments below. To obtain translations of unlabelled data, we assume access to professional translators $\mathcal{O}(\cdot)$ who can translate source-side sentences $\mathcal{S}$ and/or phrases $\mathcal{P}$ selected from $\mathcal{U}$, i.e., $\mathcal{O}(x)\ \forall x \in \mathcal{S} \subset \mathcal{U}$, and $\mathcal{O}(p)\ \forall p \in \mathcal{P} \subset \mathcal{P}_{\mathcal{U}} = \cup_{x\in\mathcal{U}} \cup_{n\in[1,N]}\ n\text{-gram}(x)$. We assume that translating sentences or phrases requires cost $c(\cdot)$, and annotation must be performed within a fixed budget $B = \sum_{x\in\mathcal{S}} c(x) + \sum_{p\in\mathcal{P}} c(p)$. This active learning procedure consists of two main steps: selection/translation (§3) and fine-tuning (§4).

3 Active Selection Strategies

3.1 Sentence Selection Strategies

Existing sentence-based active learning methods usually define a sentence-level scoring function $\phi(x, \cdot)$, and select sentences with the top scores. Following Zeng et al. (2019), we categorize these methods into two classes: data-driven and model-driven methods. Data-driven methods only rely on the unlabeled data $\mathcal{U}$ and the labeled data $\mathcal{L}$, i.e., $\phi(x, \mathcal{U}, \mathcal{L})$, and usually score sentences based on the trade-off between the density and diversity of the selected sentences. In contrast, model-driven approaches usually estimate the prediction uncertainty of a source sentence given the current MT model θ, i.e., $\phi(x, \theta, \mathcal{U}, \mathcal{L})$, and select sentences with high uncertainty for training the model. Before getting to our proposed phrase-based strategies in §3.2 we highlight several existing sentence selection strategies.

Random Sampling: One easy strategy is randomly sampling sentences from the unlabeled data $\mathcal{U}$ for annotation. Although it is simple, this method is an unbiased approximation of the data distribution in $\mathcal{U}$. Therefore, this method remains a strong baseline in the active learning literature (Gangadharaiah et al., 2009; Miura et al., 2016; Zeng et al., 2019) if the annotation budget is sufficiently large.

Margin-based Ratio Score (MRS): Zhang et al. (2018) propose to measure the distance between sentence embeddings. This method takes each unlabeled sentence, estimates its distance in embedding space from the labeled sentences in the out-of-domain corpus, and iteratively selects sentences that are more distant from sentences in the labeled data. In our instantiation of this method, we leverage the pre-trained mBERT model (Devlin et al., 2019) to extract sentence representation $\mathbf{e}_x$ of a particular sentence x.[2] Instead of using a cosine similarity function, we measure a ratio-based score which is the ratio between the cosine similarity of $(\mathbf{e}_x, \mathbf{e}_{x'})$ and the average cosine similarity with their k nearest neighbors in Eq. (1), because the margin-based ratio score has been shown effective in sentence retrieval in (Artetxe and Schwenk, 2019).

$$\text{ratio}(\mathbf{e}_x, \mathbf{e}_{x'}) = \frac{\cos\left(\mathbf{e}_x, \mathbf{e}_{x'}\right)}{\sum\limits_{z\in \text{NN}_k(x)} \frac{\cos\left(\mathbf{e}_x, \mathbf{e}_z\right)}{2k} + \sum\limits_{z\in \text{NN}_k(x')} \frac{\cos\left(\mathbf{e}_{x'}, \mathbf{e}_z\right)}{2k}}, \quad (1)$$

where k is the number of nearest neighbors.

We then compute the distance between each in-domain sentence and its nearest out-of-domain

[2] We average the word representations from the 7th layer of the mBERT model as the sentence embedding, because the middle-layer representations have proven effective in cross-lingual retrieval tasks (Pires et al., 2019; Hu et al., 2020).

neighbor within a randomly sampled subset of labeled sentences $\mathcal{L}'$:

$$\phi(x, \cdot) = \text{dist}(x, \mathcal{L}') = 1 - \max_{x' \in \mathcal{L}'} \text{ratio}(\mathbf{e}_x, \mathbf{e}_{x'}).$$

We approximate the distance between x and out-of-domain corpus $\mathcal{L}$ using a subset $\mathcal{L}'$ for efficiency purposes, because the out-of-domain $\mathcal{L}$ is usually large. Next we use the distance $\text{dist}(x, \mathcal{L}')$ as our scoring function $\phi(x, \cdot)$, and select the unlabeled sentence with the largest distance from (subsampled) sentences in the out-of-domain corpus.

Round Trip Translation Likelihood (RTTL): One model-driven method is based on a method referred to as "round trip translation" (Haffari et al., 2009; Zeng et al., 2019). The labeled data $\mathcal{L}$ is used to train two MT models $\theta_{\text{src-tgt}}, \theta_{\text{tgt-src}}$ that translate between the source and target languages in two directions. Each unlabeled source sentence $x \in \mathcal{U}$ is first translated to $\hat{y}$ in the target language by $\theta_{\text{src-tgt}}$, and then $\hat{y}$ is translated to $\hat{x}$ by $\theta_{\text{tgt-src}}$. This method assumes that if this round-trip translation process fails to recover some of the content on the source side then this is an indication that the sentence may be difficult for the current model and is a good candidate for human annotation. Haffari et al. (2009) use a scoring function that computes the similarity between the original sentence x and $\hat{x}$ using the sentence-level BLEU score (Chen and Cherry, 2014), while Zeng et al. (2019) estimate the likelihood of the original source sentence x given $\hat{y}$ by the reverse MT model $\theta_{\text{tgt-src}}$.

$$\hat{y} \approx \underset{y}{\text{argmax}}\, P_{\theta_{\text{src-tgt}}}(y|x) \tag{2}$$

$$\phi(x, \cdot) = \log P_{\theta_{\text{tgt-src}}}(x|\hat{y}) \tag{3}$$

3.2 Phrase Selection Strategies

A few existing phrase-based active learning methods (Bloodgood and Callison-Burch, 2010; Miura et al., 2016) have been proposed to improve PBMT systems. These methods first determine the possible set of phrases in a sentence, select phrases to be translated according to a scoring metric, and incorporate these in the training of the PBMT system. In the following paragraphs, we introduce two phrase-based selection strategies, and discuss how to integrate this data into NMT in §4. Similar to the sentence selection strategies, we define a phrase-level scoring function $\phi(p, \cdot)$ and select phrases with the top scores.

n-gram Frequency (NGF) (Bloodgood and Callison-Burch, 2010): The most straightforward phrase selection strategy is to select the most frequent phrases in the unlabelled data that *do not* appear in the already labeled data. First we extract two sets of possible n-grams ($n \leq 4$) from sentences in $\mathcal{U}$ and $\mathcal{L}$, which are defined as $\mathcal{P}_\mathcal{U} = \cup_{x \in \mathcal{U}} \cup_{n \in [1,N]} n\text{-gram}(x)$, and $\mathcal{P}_\mathcal{L} = \cup_{(x,y) \in \mathcal{L}} \cup_{n \in [1,N]} n\text{-gram}(x)$. We then score each phrase as follows:

$$\phi(p, \cdot) = \begin{cases} occ(p, \mathcal{U}), & \text{if } p \in \mathcal{P}_\mathcal{U}, p \notin \mathcal{P}_\mathcal{L} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\text{occ}(p)$ counts the occurrences of p in $\mathcal{U}$. We then select the top frequent phrases until we use up the budget for annotating phrases.

Semi-Maximal Phrases (NGF-SMP): The two phrase sets $\mathcal{P}_\mathcal{U}, \mathcal{P}_\mathcal{L}$ extracted by the n-gram Frequency method contain many substrings that also occur in some longer strings. For example, p = "eines der" always co-occurs with the longer p' = "eines der besten" in the WMT14 German-English dataset. To identify the longer strings, Miura et al. (2016) proposed the following semi-order relation, which defines the relation between a phrase p' and its sub-string p satisfying the condition that p' occurs at least half the time of p in the corpus $\mathcal{U}$.

$$p \overset{*}{\preceq} p' \Leftrightarrow \exists \alpha, \beta : \ \alpha p \beta = p' \wedge \frac{\text{occ}(p, \mathcal{U})}{2} < \text{occ}(p', \mathcal{U}) \tag{5}$$

A phrase p is called a semi-maximal phrase if there does not exist a phrase p' in $\mathcal{U}$ such that $p \overset{*}{\preceq} p'$. Therefore, a compact subset of phrases $\mathcal{P}'_\mathcal{U}$ can be constructed by containing only semi-maximal phrases in the phrase set $\mathcal{P}_\mathcal{U}$ in $\mathcal{U}$:

$$\mathcal{P}'_\mathcal{U} = \{p | \nexists p' \in \mathcal{P}_\mathcal{U}, p \overset{*}{\preceq} p' \wedge p \in \mathcal{P}_\mathcal{U}\}. \tag{6}$$

By using semi-maximal phrases in $\mathcal{P}'_\mathcal{U}$ rather than all phrases in $\mathcal{P}_\mathcal{U}$, we remove a large number of phrases that are included in a longer phrase more than half the time, and reduce the redundancy of the selected phrases. Next we can select phrases similarly using Eq. (4) by replacing the original phrase set $\mathcal{P}_\mathcal{U}$ with the sub-set $\mathcal{P}'_\mathcal{U}$.

Notably, we select representative phrases by their occurrences instead of using a similarity function between phrase embeddings. Because it is easy to count the phrase occurrences by extract string-match while it is infeasible to do so for sentences.

As for sentence selection, measuring a similarity between sentence embeddings (e.g., MRS) provides an alternative way of matching sentences.

3.3 Hybrid Selection Strategy

Phrase-based selection has its benefits, such as efficient annotation of core vocabulary from the target domain. However, at the same time it lacks the ability to identify larger sentence structure that may nonetheless be unique to the target domain. Modeling this structure is particularly important for NMT (in constrast to PBMT), as NMT directly learns both lexical and syntactic transformations within the same model.

Because of this, we propose a simple yet novel hybrid selection strategy that leverages the benefits of both sentence-based and phrase-based selection strategies. Specifically, we allocate our budget in a way to annotate sentences with B_s words from our set of sentences and B_p words from our set of phrases. Depending on the specific sentence-based and phrase-based selection strategies chosen in the hybrid selection strategy, it is non-trivial to determine which selection strategy improves the in-domain translation performance more than the other one before actual finetuning. Therefore, in our implementation, we assume that we have no prior knowledge about which selection strategies will be most effective, and simply evenly distribute the annotation budget into the sentence-based and phrase-based strategies. We leave more sophisticated allocation strategies as future work, and we discuss some potential avenues briefly in §7.

4 Training with Sentences and Phrases

After data selection, we fine-tune the base NMT model on the newly translated data. This is essentially an extreme form of domain adaptation where we adapt a base NMT model trained on out-of-domain data to a new domain. Specifically, we adapt a strategy of *mixed fine-tuning* (Luong and Manning, 2015), which continues training a pre-trained out-of-domain model on both in-domain data and a certain amount of out-of-domain data to prevent overfitting to relatively small in-domain data. Compared to the standard domain adaptation setting where we have only a small number of in-domain sentences, our phrase-level active learning setting has the additional difficulty of having to use short translations of individual phrases. In the following, we describe both methods to choose which data to use in mixed fine-tuning, and how to incorporate phrasal translations.

4.1 Data Mixing

For data mixing, we sample a subset $\mathcal{L}_r$ of data directly from the labeled set $\mathcal{L}'$, and concatenate $\mathcal{L}_r$ with the newly annotated sentences $\mathcal{L}_s$ and phrases $\mathcal{L}_p$ for mixed fine-tuning (Line 8 in Algorithm 1). Specifically, we define a distribution function ψ over $\mathcal{L}'$, and either sample by $(x, y) \sim \psi$ or greedily take the most likely data by $(x, y) = \text{argmax}_{(x,y)\in\mathcal{L}'} \psi(x, y)$ iteratively for M times to obtain the subset $\mathcal{L}_r$ of M parallel data.

Random Sampling: The most simple way to select out-of-domain data is to randomly sample sentences from the out-of-domain corpus $\mathcal{L}'$, i.e., $(x, y) \sim \text{Uniform}(\mathcal{L}')$. Although it is simple, this has been popularly used in the literature of domain adaption for NMT (Chu and Wang, 2018).

Retrieve Similar Sentences: Recently, Aharoni and Goldberg (2020) showed that pre-trained language models implicitly learn sentence embeddings that cluster by domains, and proposed a data selection method that has proven more effective than methods based on the likelihood of an in-domain language model (Moore and Lewis, 2010). Since our base NMT model is pre-trained on out-of-domain corpus, we need to adapt the model to the domain of the unlabeled data. Instead of random sampling, we adopt the selection method in Aharoni and Goldberg (2020) to retrieve parallel sentences from $\mathcal{L}'$ that are close to the in-domain sentences in $\mathcal{U}$. To do so, we leverage the contextualized sentence representations, and measure the distance of a source sentence in $\mathcal{L}'$ w.r.t. the unlabeled corpus $\mathcal{U}$ by $\text{ratio}(x, \mathcal{U})$, $\forall x \in \mathcal{L}'$. Next, we iteratively retrieve labeled data from $\mathcal{L}'$ that have the smallest distance scores to their nearest neighbors, i.e., $(x, y) = \text{argmax}_{(x,y)\in\mathcal{L}'} \text{ratio}(x, \mathcal{U})$.

4.2 Incorporating Phrasal Translations

In addition to obtaining real parallel data from $\mathcal{L}'$ for mixed fine-tuning, we create synthetic parallel data $(\hat{x}, \hat{y})$ by incorporating phrasal translations into existing context from $\mathcal{L}'$. Specifically, for an unlabeled sentence $x \in \mathcal{U}$ containing a newly annotated phrase p_x, we retrieve the most similar sentence pair (x^*, y^*) from $\mathcal{L}'$ by

$$(x^*, y^*) = \underset{(x',y')\in\mathcal{L}'}{\text{argmax}} \text{ratio}(\mathbf{e}_x, \mathbf{e}_{x'}) \quad (7)$$

We then alter (x^*, y^*) with the newly annotated phrase pair (p_x, p_y) to create synthetic sentence pair $(\hat{x}, \hat{y})$. Similar to data mixing, we concatenate the set of synthetic data with the annotated sentences $\mathcal{L}_s$ and phrases $\mathcal{L}_p$ for mixed fine-tuning.

Switch Phrases: Inspired by existing data augmentation methods (Fadaee et al., 2017), we examine a data augmentation method that switches out phrases in the out-of-domain sentence pairs in $\mathcal{L}'$ by the newly annotated phrase pairs from $\mathcal{U}$. First, we define the following operation $\text{Switch}(x, p, i)$ that returns a new sentence by substituting the phrase at the i-th position in x^* by p_x.

$$\text{Switch}(x^*, p_x, i) = [x^*_{<i}; p_x; x^*_{\geq i+|p_x|}] \quad (8)$$

Next, we enumerate all possible positions in x^* for switching phrases, and then apply the in-domain language model trained on $\mathcal{U}$ to select the most probably synthetic sentence by

$$\hat{x} = \underset{\substack{x'=\text{Switch}(x^*,p_x,i) \\ \forall 0 \leq i < |x^*|-|p_x|, \\ p_x \in \cup_{n\in[1,N]} n\text{-gram}(x)}}{\text{argmax}} P_{\text{LM}}(x'), \quad (9)$$

where p_x is a phrase in the unlabeled sentence x. Notably, we use a 4-gram language model implemented in `KenLM`[3]. Since sentences are usually short (average length of 10-25 words), creating a synthetic sentence takes $O(|x^*||x|)$ scoring operations by the language model.

To synthesize the corresponding $\hat{y}$ from the retrieved target sentence y^*, we apply a word alignment model trained on $\mathcal{L}$ to find the index j for the translation of the replaced phrase $x^*_{i:i+|p_x|}$ in y^*, and substitute the phrase at the j-th position in y^* by p_y to obtain $\hat{y} = \text{Switch}(y^*, p_y, j)$.

Contextualized Phrases: The other idea is to augment the context of a newly annotated phrase pair (p_x, p_y), since a phrase p_x lacks larger sentence structure. Specifically, we define the contextualized operation that augments a phrase p_x in x by appending it to the retrieved sentence x^*.

$$\text{Contextualize}(x^*, p_x) = [x^*, p_x] \quad (10)$$

We then enumerate all annotated phrases in x, and apply an in-domain language model to find the most probable annotated phrase pair (p_x, p_y) that synthesizes $\hat{x}$. The corresponding $\hat{y}$ can be obtained by $\text{Contextualize}(y^*, p_y)$.

$$\hat{x} = \underset{\substack{x'=[x^*,p_x] \\ \forall p_x \in \cup_{n\in[1,N]} n\text{-gram}(x)}}{\text{argmax}} P_{\text{LM}}(x') \quad (11)$$

5 Experiments

5.1 Experimental Setting

We use the WMT14 German-English data as the out-of-domain labeled data for training our base NMT model, and take the source sentences of two parallel corpora in the medicine and IT domains (Koehn and Knowles, 2017) as the unlabeled data. More details can be found in Appendix B.1.

As our NMT model, we use a 6-layer 512-unit Transformer (Vaswani et al., 2017) implemented in `Fairseq`,[4] and use a subword vocabulary of 50,000 for both languages constructed by Byte Pair Encoding (Sennrich et al., 2016). We train the base model with Adam for 10 epochs with 4K warmup steps and a peak learning rate of 1e-3, and decay the learning rate based on the inverse square root of the number of update steps (Vaswani et al., 2017).

For active learning, we set our annotation budgets by number of words translated (following the prevailing translation market practice to charge for jobs by the word), and investigate the budgets from 2.5K words up to 40K words.[5] After data selection (§3), we obtain a set $\mathcal{L}_r$ of M parallel sentences (§4), and set the size $M = |\mathcal{L}_p|$ where $\mathcal{L}_p$ is selected by NGF-SMP. We then fix $\mathcal{L}_r$ for mixed fine-tuning in all experiments, and continue fine-tuning the base model on a mixture of the newly-translated data and $\mathcal{L}_r$ for 5 more epochs.

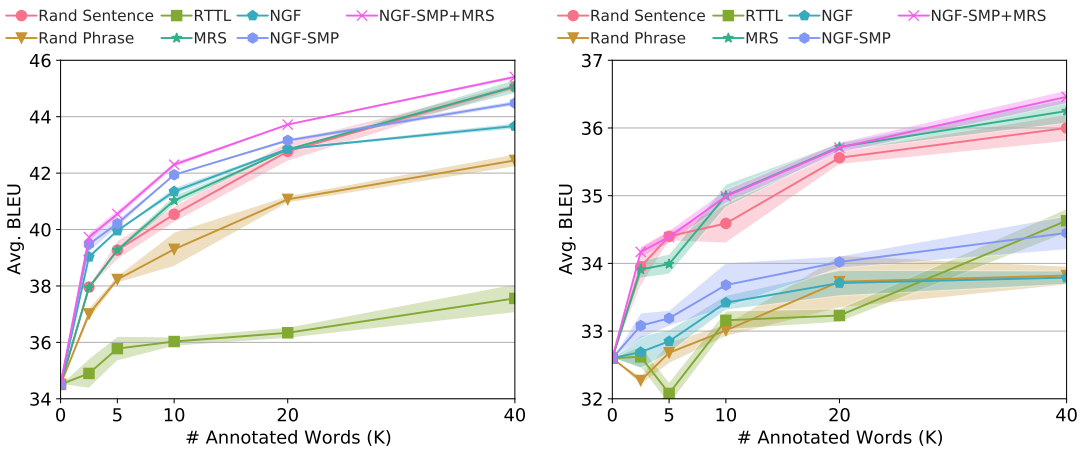
5.2 Word-level Translation Accuracy

Since our selection and mixed fine-tuning methods focus on leveraging phrasal translations for domain adaptation, we perform a fine-grained analysis on the word-level translation accuracy of the NMT systems due to the domain shift. A source word is defined as an unseen in-domain word when it never appears in the out-of-domain corpus. If phrase selection strategies select more in-domain words, we would expect a higher translation accuracy of such in-domain words by the adapted NMT systems using phrase selection. As a result, we compare the

[3] https://github.com/kpu/kenlm

[4] https://github.com/pytorch/fairseq

[5] At current market rates, this would cost from 491 to 7,092 USD for German-English translation by professional translators at https://translated.com/.



(a) WMT14→Medicine

(b) WMT14→IT

Figure 2: Average BLEU score over 3 runs for adapting a base NMT to the Medicine and IT domains.

translation accuracy of in-domain words by the NMT models using different selection strategies in Figure 3. As shown in the figure, NGF-SMP significantly improves the translation accuracy of the in-domain words with a small annotation budget. In contrast, MRS falls short of the other compared methods when the annotation budget is less than 80K words. Moreover, we find that the hybrid selection strategy of NGF-SMP and MRS can combine the merits of both methods, and obtain an even higher accuracy when the budget is greater than 40K annotated words. Qualitatively, the example in Table 1 shows the translations for a source sentence with all words appearing in the medical domain. The NMT model adapted by MRS translates the first half of the source sentence by picking the correct word "exercised", while the NMT model adapted by NGF-SMP generates the correction translation "somnolence" in the second half of the output. The NMT model using the hybrid of NGF-SMP and MRS strategies translates both words correctly (more examples in Appendix B.2).

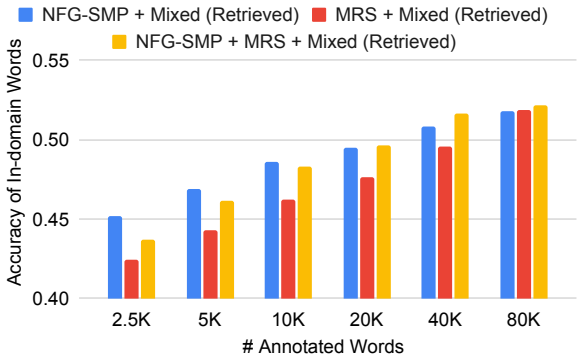


Figure 3: Translation accuracy of in-domain words in the test set from the medicine domain

5.3 How Does Each Selection Strategy Help?

We examine the question of which selection strategy (§3) best improves accuracy on in-domain test data. For mixed fine-tuning, in this section we use the retrieved out-of-domain parallel data for a fair comparison among all active selection strategies. Figure 2 shows the average BLEU score and the standard deviation of the adapted MT systems to two new domains over 3 independent runs.[6]

Comparing among sentence selection strategies in Figure 2, MRS performs slightly better than the random sentence selection baseline on adapting the NMT model to the IT domain with smaller standard deviation values, and performs comparably on adapting to the medicine domain. However, we observe that RTTL performs worst, and we conjecture that this is due to the usage of the base NMT models that are trained on the out-of-domain parallel data in both directions. The errors accumulated from the round trip translation process lead to an inaccurate estimation of the uncertainty score for a source sentence. Table 2 shows the top 5 sentences selected by RTTL. The selected sentences in the medicine domain are short phrases rather than complete sentences, and those selected in the IT domain contain duplicate phrases such as "bewerten mitâ".

[6]To obtain a stable result, we independently run the active learning procedure with different selection strategies 3 times, collect new translation data, and concatenate them with the same data retrieved from out-of-domain labeled data

	Output	S-BLEU
Source	Jedoch ist Vorsicht geboten, da Berichten zufolge Verwirrung und Somnolenz während der Behandlung auftreten können.	
Reference	However, caution should be exercised as confusion and somnolence have been reported.	
NGF-SMP	However, caution is required, as there are reports of confusion and somnolence during the treatment.	15.71
MRS	However, caution should be exercised, as confusion and drowsiness may occur during the treatment.	15.62
NGF-SMP+MRS	However, caution should be exercised as confusion and somnolence may occur during the treatment.	15.71

Table 1: Translations generated by NMT models using different selection strategies. The last column shows the sentence BLEU score of the translations. Translation errors are highlighted in red.

MED	Portugal Lundbeck Portugal Lda Quinta da Fonte Edifício D. Bronchitis Gastrointestinaltrakt : Neugebore 139 B.
IT	Eigenschaften des Stichwortes â % 1â bewerten mitâ Drei Sternenâ keine Speicherplatzinformation aufâ procfsâ bewerten mitâ Einem Sternâ neue und einzelne auswÃ Â hlen

Table 2: Top 5 sentences selected by RTTL

For phrase-based selection methods, NGF-SMP significantly outperforms the random phrase selection strategy. Further, NGF-SMP even outperforms sentence selection methods when the annotation budget is small (less than 20k words) for adaption to the medicine domain. As we increase the annotation budget to 40K annotated words, sentence selection strategies outperform phrase selection strategies. This indicates that if we keep training NMT systems on shorter phrase pairs when the annotation budget is sufficient, the NMT systems would be limited by lack of longer sentence structures. In Figure 2b, we also find that NMT models trained with phrasal translations fall short of those trained with sentence translations when adapting to the IT domain. It is hard to train the NMT systems to translate certain phrases correctly without the sentence context. For example, "Persönlichen Ordner" in the IT domain is translated to "home directory" rather than "personal folder" in the sentence "jedes Skript dieses Dialogs hat Schreib-Zugriff auf Ihren Persönlichen Ordner ".

Finally, the hybrid selection of NGF-SMP and MRS strategies outperforms the individual selection strategies over every budget in our set of budgets, i.e., 2.5K, 5K, 10K, 20K, 40K annotated words, improving the best phrase selection strategy NGF-SMP by 0.49 average BLEU points, and the best sentence selection strategy MRS by 1.11 average BLEU points in the medicine domain. Notably, the phrase-based selection strategy especially helps in the scenario where the context is not required to translate domain-specific words, for example, the name of a medicine or a disease in the medicine domain (See the first example in Appendix B.2). For the adaptation scenario that requires a longer context in some domains such as IT, the hybrid strategy can also significantly outperforms the best phrase-based strategy NGF-SMP by 1.2 average BLEU points, and the best sentence selection strategy MRS by 0.15 BLEU points. Overall, our hybrid selection strategy is effective to combine the merits of both sentence and phrase selection strategies in the domain adaptation setting.

5.4 How Representative Are the Selected Data?

If the selected data has a significant overlap of segments with the in-domain test data, we would expect a better adaptation performance of the NMT trained on the selected data. Therefore we investigate the n-gram overlap between the selected data and the test data when we annotate 5K words from the medicine corpus, and report the average BLEU score of the adapted NMT models trained on the selected data in Table 3. Interestingly, we find that there exists a high correlation ($\rho \approx 0.8$) between the n-gram overlap and the average BLEU score, which indicates that the n-gram overlap with the test set can be used as a good measure of whether the selected data is useful for improving the NMT model in the new domain. Compared to the random phrase selection, NGF-SMP selects phrases with a high overlap with the test data. We also observe that sentence selection strategies cover fewer phrases in the test data than phrase selection strategies. This also corroborates our assumption that asking translators to annotate phrases that the MT system can already translate well is not cost-effective to improve the in-domain translation performance.

Methods	uni-gram	bi-gram	tri-gram	4-gram	Avg. BLEU
OoD Data	79.33	32.65	7.30	1.10	34.51
+ Random Sentence	82.81	38.45	11.62	3.73	39.27
+ RTTL	80.70	35.76	9.85	3.04	35.78
+ MRS	82.74	38.83	12.01	4.05	39.27
+ Random Phrase	82.36	35.84	7.98	1.15	38.23
+ NGF	84.45	41.82	14.94	6.17	39.96
+ NGF-SMP	85.80	43.13	16.15	7.11	40.21
+ NGF-SMP + MRS	84.48	41.89	14.98	6.48	40.55
ID Training Data	98.58	87.30	67.61	52.11	57.59
Pearson Correlation	0.90	0.83	0.80	0.78	/

Table 3: Percentage of the n-gram in the test sentences that are covered by the selected data with 5K words, the out-of-domain training data and the in-domain training data. The last row shows the Pearson correlation coefficient between n-gram overlap and avg. BLEU score.

5.5 How Redundant Are the Selected Data?

To answer this question, we first define "in-domain words" as words that only appear in the in-domain test set but do not exist in the out-of-domain data. We report the statistics of the in-domain word types word counts in the selected data with 10K annotated words in Table 5. We find that phrase selection strategies select more unique in-domain word types and counts than the sentence selection strategies. This indicates that phrase selection strategies leverage the same amount of budget effectively to annotate more diverse in-domain words than sentence selection strategies.

5.6 How Do Phrasal Translations Help in Mixed Fine-tuning?

We further investigate the effect of mixed fine-tuning using the newly annotated in-domain data and sub-sampled out-of-domain data when comparing with fine-tuning only on the newly annotated data. Table 4 shows the average BLEU score and the standard deviation values over 3 independent runs. Compared to fine-tuning on only annotated data, adding randomly sampled sentence pairs from the out-of-domain data helps when the annotation budget is less than 5K annotated words, but hurts when we increase the budget. In contrast, adding sentences retrieved by the similarity in the sentence embedding space not only outperforms fine-tuning only on annotated data and mixed fine-tuning with randomly sampled sentences, but also achieves smaller standard deviation values. On the other hand, mixed fine-tuning on synthetic data by switching phrases performs slightly worse than the mixed fine-tuning on real retrieved data, but outperforms the fine-tuning without any out-of-domain data, especially when the annotation budget is small, e.g., 5K annotated words. Combining synthetic data by switching phrase and real retrieved data for mixed fine-tuning also improves the translation performance over the training only on synthetic data. However, the contextualized method performs worst among all mixed fine-tuning methods, which indicates that simply appending existing sentence context to phrasal translations might potentially introduce noise to the training data.

6 Related Work

Active Learning for Machine Translation Pioneering works on active learning for machine translation focus on selecting sentences that are most useful for training PBMT. This includes sentence selection strategies based on maximizing the percentage of unseen n-gram (Eck et al., 2005), n-gram frequency, lexical diversity (Haffari et al., 2009), or in-domain coverage (Ananthakrishnan et al., 2010). These sentence selection strategies have been used in active learning algorithms to deal with static data in the batch mode (Ananthakrishnan et al., 2010), or steaming data in the interactive setting (González-Rubio et al., 2012; Peris and Casacuberta, 2018; Lam et al., 2019).

For phrase-level annotations, there have been a few works applying phrase-based selection (Bloodgood and Callison-Burch, 2010; Miura et al., 2016) to PBMT. While the annotated phrases can be easily integrated by adding them with estimated translation probability to the existing phrase table in PBMT, it it less trivial to integrate these phrase-level annotations in NMT. Arthur et al. (2016) integrated the word-level translations to NMT by interpolating the probability of the NMT decoder with the estimated lexical probability. However, this approach requires a modification of the NMT model. Our paper investigates data-driven approaches that augment the training data by leveraging annotated phrases and existing parallel data.

Word/Phrase-based Data Augmentation The other line of research investigates data augmentation methods that leverage word or phrase translations to create synthetic parallel data for training MT models. This includes augmentation methods that replace a word in the existing parallel data with a low-frequency word sampled from the frequency distribution of the vocabulary (Xie et al., 2017) or from the probability of language models in both directions (Fadaee et al., 2017; Kobayashi, 2018). Wang et al. (2018) proposed an effective method

Out-of-domain Data				In-domain Data		2.5K	5K	10K	20K	40K
Sampled	Retrieved	Switched	Contextualized	NGF-SMP	MRS					
				✓		39.39 ± 0.14	39.22 ± 0.00	40.56 ± 0.02	41.19 ± 0.25	44.07 ± 0.33
					✓	37.94 ± 0.08	38.68 ± 0.54	40.62 ± 0.59	42.62 ± 0.03	45.00 ± 0.11
				✓	✓	38.94 ± 0.02	39.60 ± 0.09	41.34 ± 0.12	42.44 ± 0.15	44.90 ± 0.06
✓				✓	✓	39.46 ± 0.14	40.51 ± 0.23	40.62 ± 0.49	41.82 ± 0.26	43.78 ± 0.57
	✓			✓	✓	**39.73** ± 0.16	40.55 ± 0.14	**42.30** ± 0.10	**43.72** ± 0.04	**45.41** ± 0.08
		✓		✓	✓	38.93 ± 0.36	40.59 ± 0.17	41.82 ± 0.29	42.70 ± 0.37	45.33 ± 0.04
			✓	✓	✓	35.36 ± 0.38	37.85 ± 0.68	39.96 ± 0.35	42.83 ± 0.11	44.14 ± 0.15
	✓	✓		✓	✓	39.61 ± 0.06	**40.95** ± 0.06	42.19 ± 0.08	43.42 ± 0.17	45.06 ± 0.19
	✓		✓	✓	✓	37.88 ± 0.25	39.52 ± 0.32	41.17 ± 0.28	42.80 ± 0.21	44.28 ± 0.13

Table 4: Comparison between mixed fine-tuning methods. Bold indicates highest average BLEU by column.

Methods	IDWT	WT	$\frac{\text{IDWT}}{\text{WT}}$	IDWC	WC	$\frac{\text{IDWC}}{\text{WC}}$
Random Phrase	787	2206	35.68	860	5003	17.19
NGF	489	1053	46.44	889	5002	17.77
NGF-SMP	796	1492	53.35	1076	5001	21.52
Random Sentence	631	1984	31.80	712	5023	14.17
RTTL	592	1338	44.25	961	5023	19.13
MRS	647	2056	31.47	721	5023	14.35
NGF-SMP + MRS	667	1755	38.01	859	5035	17.06

Table 5: Statistics of the unique in-domain word types and word counts in the selected data with 10K annotated words.

that randomly replaces words in parallel sentences with other random words from the in-domain vocabulary. A more recent work on dictionary-based data augmentation (Peng et al., 2020) proposed to use an existing high-quality in-domain dictionary, and replaced a source word in the existing parallel data by the most similar word in the dictionary according to the cosine similarity metric in the embedding space. In contrast, we select noisy in-domain phrases using different phrase-based selection strategies (§3.2) to ensure the selection quality in an active learning process.

7 Discussion and Future Work

In this paper, we investigate ways to incorporating phrasal translations into training NMT for domain adaptation in the active learning setting. We find that phrasal translation is particularly useful in the adaptation scenario where longer sentence context is not necessarily required to translate in-domain words correctly. In contrast, NMT systems can benefit from learning sentence structure with sentence-based selection strategies. The hybrid selection strategies can combine the merits of both sentence-based and phrase-based selection strategies. Nonetheless, there are several future directions. (1) It is worth exploring how different annotation strategies may result in a difference in cost or time. (2) Although several findings could be generalized to other language pairs, testing our methods on morphologically rich languages is our next step. (3) Our current hybrid strategy simply allocates the annotation budget evenly without assuming any prior knowledge of the strategies and the translation performance. Techniques in multi-armed bandit problems (Gittins et al., 2011) can be used to learn a good allocation strategy.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Sankaranarayanan Ananthakrishnan, Rohit Prasad, David Stallard, and Prem Natarajan. 2010. A semi-supervised batch-mode active learning strategy for improved statistical machine translation. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 126–134, Uppsala, Sweden. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Michael Bloodgood and Chris Callison-Burch. 2010. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 854–864, Uppsala, Sweden. 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, Baltimore, Maryland, USA. 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, Vancouver, Canada. Association for Computational Linguistics.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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*, pages 407–412, Portland, Oregon, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Rashmi Gangadharaiah, Ralf D. Brown, and Jaime Carbonell. 2009. Active learning in example-based machine translation. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 227–230, Odense, Denmark. Northern European Association for Language Technology (NEALT).

John Gittins, Kevin Glazebrook, and Richard Weber. 2011. *Multi-armed bandit allocation indices*. John Wiley & Sons.

Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2012. Active learning for interactive machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 245–254, Avignon, France. Association for Computational Linguistics.

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*, pages 415–423, Boulder, Colorado. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the International Conference on Machine Learning 1*, pages 7449–7459.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. 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 452–457, New Orleans, Louisiana. 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.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2019. Interactive-predictive neural machine translation through reinforcement and imitation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 96–106, Dublin, Ireland. European Association for Machine Translation.

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.

Akiva Miura, Graham Neubig, Michael Paul, and Satoshi Nakamura. 2016. Selecting syntactic, non-redundant segments in active learning for machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 20–29, San Diego, California. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (short papers), Uppsala, Sweden. Association for Computational Linguistics.

Wei Peng, Chongxuan Huang, Tianhao Li, Yun Chen, and Qun Liu. 2020. Dictionary-based data augmentation for cross-domain neural machine translation. *arXiv preprint arXiv:2004.02577*.

Álvaro Peris and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. 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.

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.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. In *International Conference on Learning Representations (ICLR)*, Toulon, France.

Xiangkai Zeng, Sarthak Garg, Rajen Chatterjee, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Empirical evaluation of active learning techniques for neural MT. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 84–93, Hong Kong, China. Association for Computational Linguistics.

Pei Zhang, Xueying Xu, and Deyi Xiong. 2018. Active learning for neural machine translation. In *2018 International Conference on Asian Language Processing (IALP)*, pages 153–158. IEEE.

Appendix

A Pseudo code

Algorithm 1 shows the active learning procedure for machine translation, which consists of two main steps: selection/translation (§3) and fine-tuning (§4).

Algorithm 1 Active Learning for Domain Adaptation of Machine Translation

procedure ACTIVEADAPTATION($\mathcal{U}, \mathcal{L}, B$)
  Inputs: the unlabelled set $\mathcal{U}$, the labelled set $\mathcal{L}$, and a budget B.
  Train a MT model θ on $\mathcal{L}$.
  $\mathcal{S}, \mathcal{P} \leftarrow$ **SELECTION**($\mathcal{U}, \mathcal{L}, B$)
  Translate $\mathcal{S}$ by $\mathcal{L}_s = \{(x, \mathcal{O}(x)) | x \in \mathcal{S}\}$
  Translate $\mathcal{P}$ by $\mathcal{L}_p = \{(p, \mathcal{O}(p)) | p \in \mathcal{P}\}$
  $\mathcal{L}_r \leftarrow$ **Obtain parallel data from $\mathcal{L}$ (§4)**
  Fine-tune θ on $\mathcal{L}_s \cup \mathcal{L}_p \cup \mathcal{L}_r$
return θ

Algorithm 2 Hybrid Phrase/Sentence Selection

procedure SELECTION($\mathcal{U}, \mathcal{L}, B$)
  Inputs: the unlabelled set $\mathcal{U}$, the labelled set $\mathcal{L}$, and a budget B.
  Initialize $\mathcal{S} = \{\}$, $\mathcal{P} = \{\}$
  Allocate the budget: $B_s, B_p \leftarrow B$
  while $\sum_{x \in \mathcal{S}} c(x) < B_s$ **do**
    $x \leftarrow \text{argmax}_{x \in \mathcal{U}} \phi(x, \cdot)$
    $\mathcal{U} = \mathcal{U} \setminus \{x\}$
    $\mathcal{S} = \mathcal{S} \cup \{x\}$
  Construct $\mathcal{P}_\mathcal{U}, \mathcal{P}_\mathcal{L}$ by strategies (§3.2)
  while $\sum_{p \in \mathcal{P}} c(p) < B_p$ **do**
    $p \leftarrow \text{argmax}_{p \in \mathcal{P}_\mathcal{U}}$ **occ**$(p, \mathcal{U})$
    $\mathcal{P}_\mathcal{U} = \mathcal{P}_\mathcal{U} \setminus \{p\}$
    $\mathcal{P} = \mathcal{P} \cup \{p\}$
  return $\mathcal{S}, \mathcal{P}$

B Experiments

B.1 Experimental Details for Reproducibility

Dataset: As pointed out in Aharoni and Goldberg (2020), there is overlap between the training data and the test data in the original split of the two corpora provided by Koehn and Knowles (2017), so we follow them in removing the duplicated sentences in the in-domain data, and re-splitting two new test sets in order to prevent the model from memorizing the selected in-domain training data that could potentially be included in the test data. Table 6 shows the data statistics.

Data	Domain	Lang	#Sentences	#Words	Vocab	Avg Len
$\mathcal{L}$	WMT14	De	4.4M	108.0M	1.9M	24.4
		En		114.5M	955.3K	25.8
$\mathcal{U}$	Medicine	De	227.2K	3.8M	114.3K	16.8
	IT	De	190.6K	2.1M	114.6K	11.5

Table 6: Data statistics of the out-of-domain labeled data in WMT14 and the in-domain unlabeled data in the medicine and IT domains.

Model: As our NMT model, we use a 6-layer 512-unit Transformer (Vaswani et al., 2017) implemented in `Fairseq`,[7] and use a subword vocabulary of 5,000 for both languages constructed by Byte Pair Encoding (Sennrich et al., 2016). The model has 45M parameters.

Training: We train the base model with Adam for 10 epochs with 4K warmup steps and a peak learning rate of 1e-3, and decay the learning rate based on the inverse square root of the number of update steps (Vaswani et al., 2017). We save the last checkpoint as our base model, and continue fine-tuning the base model on a mixture of the newly-translated data and the retrieved out-of-domain data for 5 more epochs.

Training/Inference Time: We train each model on one NVIDIA RTX 2080Ti GPU for all our experiments. Training the base NMT model takes less than 1 days, and fine-tuning the base NMT model on selected data takes less than 4hours. The decoding of 2000 sentences can be finished within 5 minutes.

B.2 Qualitative Analysis

In the first example of Table 7, the NMT model adapted by NGF-SMP can predict most words correctly while the NMT model adapted by MRS generate a random sentence.

B.3 Do Phrasal Annotations Bias NMT?

Since phrasal annotations are short and do not contain complex sentence structure, we hypothesis that NMT systems trained on phrasal annotations would be biased towards generating shorter sentences or sentences in different grammatical order w.r.t. the reference sentence. To understand this question, we analyze the length ratio between the translation outputs and the reference sentences in Figure 4.

[7] https://github.com/pytorch/fairseq

	Output	S-BLEU
Source	Schwindel, Parästhesie, Geschmacksstörung	
Reference	Dizziness, paraesthesiae, taste disorder	
NGF-SMP	Dizziness, paraesthesia, taste disturbance	23.27
MRS	The room was very small and the bathroom was very small.	0.00
NGF-SMP+MRS	Dizziness, paraesthesia, taste disturbance	23.27
Source	Über Hospitalisierung oder Todesfälle in Verbindung mit Infektionen wurde berichtet.	
Reference	Hospitalisation or fatal outcomes associated with infections have been reported.	
NGF-SMP	There have been reports of Hospitalisation or death associated with infections.	29.79
MRS	Hospitals or deaths associated with infections have been reported.	54.63
NGF-SMP+MRS	There have been reports of Hospitalisation or fatality associated with infections.	29.79

Table 7: Translations generated by NMT models using different selection strategies. The last column shows the sentence BLEU score of the translations. Translation errors are highlighted in red.

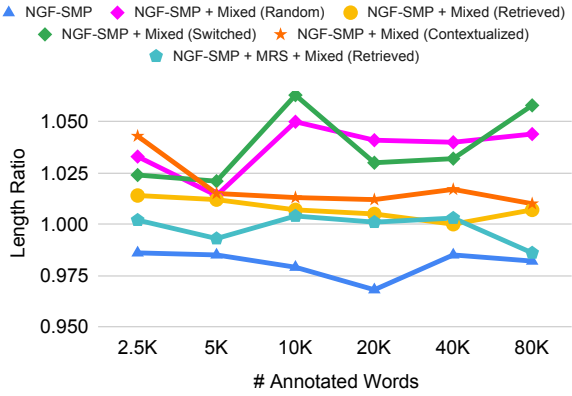


Figure 4: Length ratio between the NMT outputs and the reference sentences.

We find that the NMT model trained only on annotated phrases selected by NGF-SMP generates shorter sentences than reference sentences. In contrast, adding sentences randomly sampled from the labeled corpus $\mathcal{L}$ make the NMT model generate longer sentences than the reference sentences, while retrieving sentences from $\mathcal{L}$ that are similar to the sentences in $\mathcal{U}$ makes the model produces translation outputs with closed lengths as the reference sentences. Qualitatively, we also show the problem of generating sentences with different structures as the reference sentences in the third example in Table 1. In the third example, the NMT model trained with NGF-SMP produces a translation in an active voice, while the reference sentence uses a passive voice.

Learning Feature Weights using Reward Modeling for Denoising Parallel Corpora

Gaurav Kumar, Philipp Koehn, Sanjeev Khudanpur
Center for Language and Speech Processing
Johns Hopkins University
gkumar@cs.jhu.edu, {phi, khudanpur}@jhu.edu

Abstract

Large web-crawled corpora represent an excellent resource for improving the performance of Neural Machine Translation (NMT) systems across several language pairs. However, since these corpora are typically extremely noisy, their use is fairly limited. Current approaches to deal with this problem mainly focus on filtering using heuristics or single features such as language model scores or bilingual similarity. This work presents an alternative approach which learns weights for multiple sentence-level features. These feature weights which are optimized directly for the task of improving translation performance, are used to score and filter sentences in the noisy corpora more effectively. We provide results of applying this technique to building NMT systems using the Paracrawl corpus for Estonian-English and show that it beats strong single feature baselines and hand designed combinations. Additionally, we analyze the sensitivity of this method to different types of noise and explore if the learned weights generalize to other language pairs using the Maltese-English Paracrawl corpus.

1 Introduction

Large parallel corpora such as Paracrawl (Bañón et al., 2020) which have been crawled from online resources hold the potential to drastically improve performance of neural machine translation systems across both low and high resource language pairs. However, since these extraction efforts mostly rely on automatic language identification and document/sentence alignment methods, the resulting corpora are extremely noisy. The most frequent noise types encountered are sentence alignment errors, wrong language in source or target, and untranslated sentences. As outlined by Khayrallah and Koehn (2018), training algorithms for neural machine translation systems are particularly vulnerable to these noise types. As such, these web-crawled corpora have seen limited use in training large NMT systems.

This paper proposes a method for denoising and filtering noisy corpora which explores and searches over weighted combinations of features. During NMT training, we score sentences and create batches using random weight vectors. These batches are use to train the system and measure improvement over the validation set (reward). Finally, by modeling the weight-*reward* function, we learn the set of weights which maximize reward and are used to score and filter the noisy dataset. At a high level, this method (i) allows the use of multiple sentence level features, (ii) learns a set of interpolation weights for the features which directly maximize translation performance, (iii) requires no prior knowledge about which features are informative or even if they are mutually redundant, and (iv) trains within the NMT pipeline and does not require any special infrastructure.

We include experiments which apply this method to building NMT systems for the noisy Estonian-English Paracrawl dataset and show that it beats strong single feature filtering-baselines and hand-designed feature interpolation. Additionally, we analyze the robustness of this method in the presence of specific kinds of noise (Khayrallah and Koehn, 2018) via a controlled experiment on the Europarl datasets. Finally, we look at the impact of transferring the learned weights from one language pair (Estonian-English) to a noisy dataset of another language pair (Maltese-English Paracrawl).

We present related work in Section 2. Section 3 describes the procedure we use to model and search over the weight-feature-*reward* space to estimate feature weights which maximize translation performance. Our experiment design, datasets and features, appear in Section 4. Section 5 includes our primary results where we compare the performance of the proposed method to strong single feature filtering baselines and hand-design feature weights.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1100–1109
November 10–11, 2021.

We conclude in section 6 with an analysis of this method's performance at filtering specific kinds of noise and the application of learned weights to a different language pair.

2 Related Work

Existing efforts towards filtering and denoising noisy corpora focus on pre-filtering using hand-crafted rules and by using sentence pair scoring and filtering methods. Deterministic hand-crafted rules (Hangya and Fraser, 2018; Kurfalı and Östling, 2019) remove sentence pairs with extreme lengths, unusual sentence length ratios and exact source-target copies, and are extremely effective in removing most of the obvious automatic extraction errors. Automatic sentence pair scoring functions have been used successfully to filter noisy corpora as well. This includes the use of language models (Rossenbach et al., 2018), neural language models trained on trusted data (Junczys-Dowmunt, 2018) and lexical translation scores (González-Rubio, 2019). Chaudhary et al. (2019) propose the use of cross-lingual sentence embeddings for determining sentence pair quality while several efforts (Kurfalı and Östling, 2019; Soares and Costa-jussà, 2019; Bernier-Colborne and Lo, 2019) have focused on the use of monolingual word embeddings. Parcheta et al. (2019) use a machine translation system trained on clean data to translate the source sentences of the noisy corpus and evaluate the translation against the original target sentences using BLEU scores. Erdmann and Gwinnup (2019) and Sen et al. (2019) propose similar methods using METEOR scores and Levenshtein distance respectively. Rarrick et al. (2011), Venugopal et al. (2011) and Antonova and Misyurev (2011) present techniques for detecting machine translated sentence pairs in corpora. Tools such as LASER (Schwenk and Douze, 2017), BiCleaner (Sánchez-Cartagena et al., 2018) and Zipporah (Xu and Koehn, 2017) have been used (Chaudhary et al., 2019) for noisy corpus filtering. Curriculum learning has been used to obtain policies for data selection which can expose the model to noisy samples less often during training (Wang et al., 2018; Kumar et al., 2019). More recently, ElNokrashy et al. (2020) and Esplà Gomis et al. (2020) have used classifier based approaches to filtering noisy parallel data.

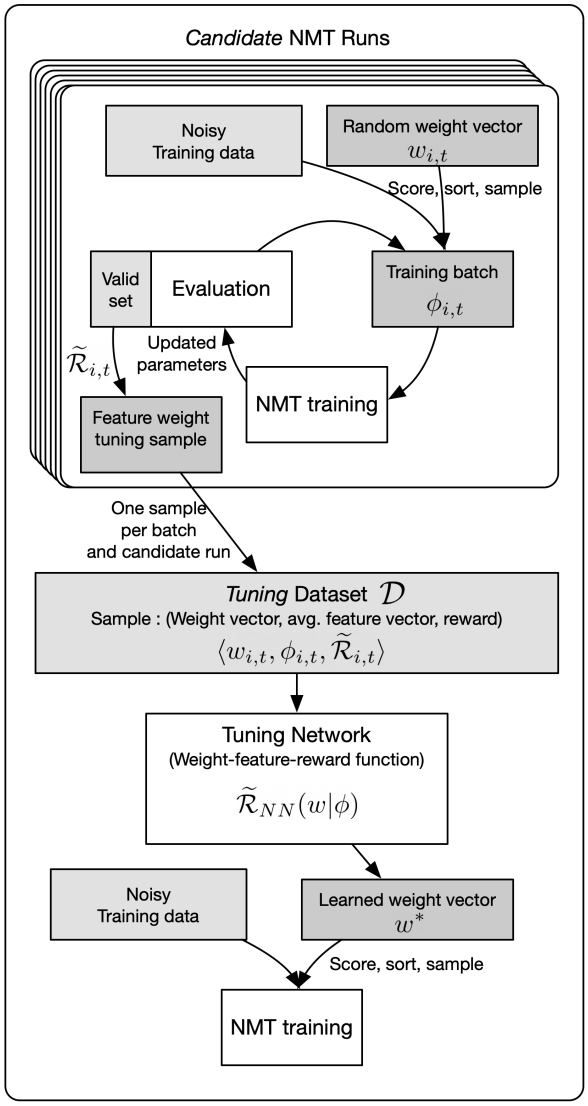


Figure 1: Overview of the proposed method for learning weights for sentence-level features to filter noisy parallel data and improve translation performance.

3 Method

The proposed method centres around finding weights for combining sentence-level features, which are then used to compute sentence-level scores and filter the noisy corpus. While the choice of features can be arbitrary, this method's performance will eventually depend on their quality, and we would ideally want them to be informative and decorrelated.

Figure 1 provides an overview of the proposed method. We first train a number of *candidate* neural machine translation (NMT) systems. During training for each candidate system, we repeatedly (i) generate a random weight vector, (ii) sample a batch of sentences from the noisy corpus based on sentence-level scores computed using this weight

vector, (iii) update NMT system parameters using this batch, and (iv) measure the improvement in translation quality on a validation set following this update. The weight vector w, the average feature vector ϕ of the batch, and the improvement $\mathcal{R}$ on the validation set (*reward*) are recorded for each batch t during the training of each candidate NMT system i, and $\langle w_{i,t}, \phi_{i,t}, \mathcal{R}_{i,t} \rangle$ becomes a sample in new data set $\mathcal{D}$, called the *tuning* data set[1], for learning feature weights to maximize reward. Hence, even though the parameters of the *candidate* systems are not used directly, they are used to gather noisy *candidate* evaluations of the latent weight-feature-reward function.

Once we have $\mathcal{D}$, we use a feed-forward network to learn the weight vector that maximizes the reward. The learned weight vector w^* is then used to compute sentence-level scores and filter the noisy data set. The *final* NMT system is trained using this *clean* data set.

Some subtleties in normalizing the observed rewards and learning weights are explained below.

3.1 Candidate NMT runs

Note from the bottom of Figure 1 that the learned weight vector w^* is used to sort all the sentences in the noisy training data, and the top-scoring ones are used for final NMT training. The purpose of the candidate NMT training runs is to generate the *tuning* data set $\mathcal{D}$ from which w^* is learned. Therefore, the setup for the candidate runs mimics typical NMT training, but for the following differences.

1. **Selecting batches**: For selecting sentences to constitute a batch, we first sample a random weight vector w of dimension $|\phi|$, the number of sentence-level features, uniformly[2] from $[-2.5, 2.5]^{|\phi|}$. Ideally, we would score *all* sentences in the noisy data set and then filter the top sentences to create a batch. However, this is prohibitively slow to do for every batch. Hence, we randomly sample twice the number of sentences required to constitute the batch, score them, and select the top half. For the i^{th} sentence, the score s_i is a dot product of its feature vectors with the weight vector:

$$s_i = \sum_{i=1}^{|\phi|} w_i \phi_i \tag{1}$$

   The selected sentences are removed from the training pool for this epoch. This method of batch selection ensures that the sampled weight vector determines which sentences are selected and that their average feature vector is significantly different from one obtained using unbiased/random selection.

2. **Reward computation**: The reward must represent how the choice of w (through the sentences selected to form the batch) impacts translation performance. This is approximated by computing the perplexity of a validation set following a parameter update with the selected batch. However, since perplexity naturally decays in standard NMT training, batches at the beginning of the training will naturally receive larger rewards, obscuring the impact of sentence selection. We mitigate this effect by using delta-perplexity, i.e. the change in perplexity of the validation set over a window of updates.

3. **Accumulating training samples**: For each batch t of candidate run i, we collect the random weight vector $w_{i,t}$, the batch feature vector $\phi_{i.t}$, defined as the average of the feature vectors of all sentences in the batch, and the reward $\mathcal{R}_{i,t}$. These triples are gathered from all batches during training, across all candidate training runs, to form the data set $\mathcal{D}$ for learning the feature weights.

3.2 Reward Normalization

As a further way to make the rewards time-invariant with respect to NMT training, the observed rewards $\mathcal{R}_{i.t}$ are normalized with respect to an expected reward estimated from a set of *baseline* NMT runs. Specifically, at each time step t, we compute the rewards $\mathcal{R}^b_{j,t}$ of $j = 1, \ldots, J$ concurrent training runs—whose batches selected in the standard manner—and, for each of the candidate NMT runs, we set

$$\widetilde{\mathcal{R}}_{i,t} = \mathcal{R}_{i,t} - \frac{1}{J} \sum_{j=1}^{J} \mathcal{R}^b_{j,t}, \tag{2}$$

[1] Not to be confused with the validation set which contains sentence pairs, this dataset is solely used to model the weight-reward function and contains no sentence identity beyond feature vectors.

[2] The range of the uniform distribution represents the plausible range of weights given the features.

where J is the number of baseline systems used.

Going forward, we do not need to track the identity of the update which led to a training sample, t, or the candidate system c_i which produced it.

3.3 Learning Feature Weights

The i^{th} sample $\langle w_i, \phi_i, \widetilde{\mathcal{R}}_i \rangle$ in $\mathcal{D}$ may be viewed as a (noisy) evaluation of an unknown function $\mathcal{R}(w|\phi)$. This function maps a vector w to final NMT quality, given a fixed sentence-level feature function ϕ and the stipulation that sentences are selected for training based on a weighted combination of their feature values using weights w. Furthermore, if we learn this function using $\mathcal{D}$, we may use the w^* that maximizes the learned function $\widetilde{\mathcal{R}}_{NN}(w|\phi)$ for our final denoising and NMT training. Specifically, we propose to use

$$\begin{aligned} w^* &= \arg\max_w \mathcal{R}(w|\phi) \\ &\approx \arg\max_w \widetilde{\mathcal{R}}_{NN}(w|\phi) \end{aligned} \tag{3}$$

We propose learning $\widetilde{\mathcal{R}}_{NN}(w|\phi)$ via a simple feed-forward neural network that maps the weights w_i to the observed reward $\widetilde{\mathcal{R}}_i$. We consider two ways of providing input to this neural network, one that uses only the w_i, and another that modulates w_i with batch quality, represented by ϕ_i.

1. **Weight-based**: We use a feed-forward network with the weight vectors w_i as input and learn to predict the observed reward $\widetilde{\mathcal{R}}_i$. Since the weight vectors interact directly with the feature vectors to determine which sentences are sampled to create a batch, we hypothesize that maximizing this weight-reward function will produce feature weights which will lead to better sentence sampling.

2. **Feature-based**: Since the *tuning* samples are noisy evaluations of the function $\mathcal{R}(w|\phi)$, we often encounter samples where weight vectors are close in weight space but have different rewards. To counter this problem, when using a feed-forward network to learn $\widetilde{\mathcal{R}}_{NN}(w|\phi)$, we scale the weight vector input w_i by the sum of the corresponding feature vector ϕ_i. This has the effect of keeping weight vectors which have similar feature vectors close in input space and moving apart those with significantly different feature vectors.

Once this neural network is learned from $\mathcal{D}$, we perform a grid search over its input space, as defined in Section 3.1, to find the maximizer of (3).

3.4 Re-sampling and training

The weight vector w^* learned from the previous section is used to score all sentences from the original noisy data set. We sort the sentences by these scores and sample the top candidates to form the *clean* training data set and use it to train a standard NMT system.

4 Experiment Setup

We use Fairseq (Ott et al., 2019) for our neural machine translation systems configured to be identical to the systems described in Ng et al. (2019). The feed-forward network used to tune weights has two 512-dimensional layers and is trained using standard SGD using a learning rate of 0.1. The grid search for the weights was done on the range $[-2.5, 2.5]$ with 5000 points uniformly distributed per dimension. The number of samples used for reward normalization was 3 and the window for computing the delta-perplexity reward was set to 3.

4.1 Corpora

We use the Paracrawl Benchmarks (Bañón et al., 2020) data set in Estonian-English for all our experiments. These consist of documents where sentences were aligned using Vecalign (Thompson and Koehn, 2019) and then de-duplicated so that each sentence pair only occurs once in the data set. The test and validation sets for our experiments in Estonian-English are *newstest2018* and *newsdev2018* respectively. Statistics of these corpora appear in Table 1.

	train	**valid**	**test**
Sentence Pairs	22.8m	2k	2k
Source Tokens	190m	29k	31k
Target Tokens	207m	38k	40k
Avg. Len (src)	9.8	14.5	15.3
Avg. Len (tgt)	10.7	19.1	20.1

Table 1: Statistics for the processed Estonian-Engligh (Es-En) Paracrawl data set and its corresponding validation and test sets. The training corpus was filtered using Vecalign scores; the raw corpus contains about 168m sentence pairs.

4.2 Features

We use five sentence-level features for all our filtering experiments. They are, (i) IBM Model 1 alignment scores (Brown et al., 1993), (ii and iii) source and target language model scores, (iv) dual conditional cross entropy (Junczys-Dowmunt, 2018) and (v) sentence length ratio. We experimented with aggregate features such as Zipporah (Xu and Koehn, 2017), BiCleaner (Sánchez-Cartagena et al., 2018) and bilingual features such as LASER (Schwenk and Douze, 2017) and these were used to replicate the baselines from Bañón et al. (2020) for our dataset. The IBM Model 1 scores were obtained using the Moses (Koehn et al., 2007) pipeline. The Estonian and English language models were trained on their respective NewsCrawl data sets[3]. The *clean* machine translation model for computing the conditional dual-cross entropy scores is trained on the *Europarl*v8 data set[4]. All features are gaussianized using the Yeo-Johnson power transformation and then normalized to have zero mean and unit variance.

5 Results

For our experiments, we scored all sentences in the noisy corpus, sorted and sampled the top parallel sentences to form subsets with 10, 15 and 20 million English words. These filtered data sets were used to train standard NMT systems and performance was evaluated on the test set described in the previous section. The results of these filtering experiments appear in Table 2.

First, we evaluate the efficacy of all the features we use for our interpolation task by filtering the data set on these features alone. Additionally, to include some strong baselines, we use three out-of-the-box, scoring features which provided strong results in the WMT 2020 parallel corpus filtering task[5] (Bañón et al., 2020; Chaudhary et al., 2019). These are BiCleaner, Zipporah and LASER. Of these, LASER provides the strongest filtering and translation results beating the other two by 0.3 to 0.9 BLEU points. Of the five features we use for our experiments, dual cross-entropy (Junczys-Dowmunt, 2018) is the strongest feature and matches the performance of LASER. Using source or target language model scores in isolation leads to the weakest translation performance while IBM Model 1 scores perform only slightly better than them. Surprisingly, the simple sentence length ratio feature beats all other features except dual cross-entropy by 1.4 to 1.6 BLEU points. This is a strong indicator of the type of noise in the data set and that bilingual features (even simple ones) perform better than monolingual features such as language model scores.

	10m	**15m**	**20m**
1-Feature Filtering Baselines			
Zipporah	20.4	21.3	21.3
BiCleaner	19.8	20.9	21.2
LASER	21.7	22.4	22.5
IBM Model 1	18.1	19.9	20.8
Target LM	17.6	19.5	20.4
Source LM	17.4	19.4	20.4
Dual Cross-Entropy	21.5	22.4	22.6
Sentence Length Ratio	19.7	20.2	21.2
Filtering using Feature Weights			
Uniform weight baseline	20.9	21.5	21.6
Weight based (14)	22.1	**23.1**	**23.5**
Feature based (15)	**22.4**	**23.1**	**23.6**

Table 2: BLEU scores for the Estonian-English NMT systems where the training data was filtered using single features or a learned weighted combination of features. Feature weights were learned using the proposed method. The number of candidate runs which produced the best results appear in parentheses.

Next, we look at interpolation of features using weights learned using the proposed method. As a baseline, we also include an experiment which filters based on a uniform interpolation of the five features we use. This baseline performs worse than the strongest single feature filtering experiments by 0.5 to 1 BLEU points. For both the weight-based and feature-based methods of learning interpolation weights for the features, a significant number of *candidate* runs are required before adequate performance is achieved. This is not surprising, since we are searching for an optimal weight vector in a fairly large weight space and we need a large number of samples before a good representation of the weight-reward function can be learned. Figure 2 shows the improvement in BLEU scores for the weight-based approach as data from more *candidate* runs in added to the *tuning* stage for learning weights and filtering the data set. The performance

[3] statmt.org/wmt18/translation-task.html

[4] statmt.org/europarl/

[5] statmt.org/wmt20/parallel-corpus-filtering.html

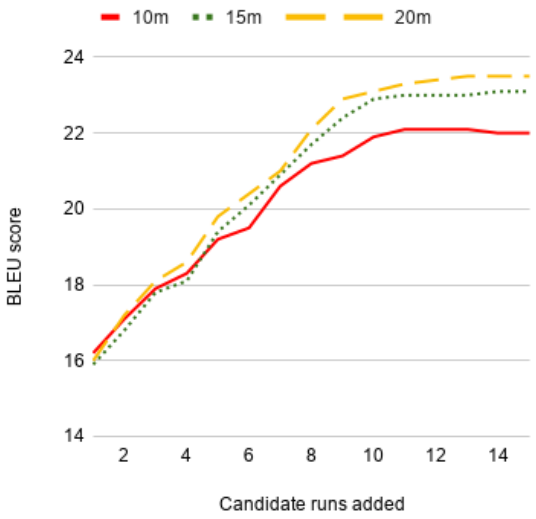


Figure 2: Improvement in BLEU scores of the final NMT system as data from additional 'candidate' training runs is added to the tuning stage to learn weights. Training data was filtered using the learned weights.

of the final NMT system steadily improves as more data from more systems is added and eventually converges.

Our strongest result was achieved with 14 *candidate* runs for the weight-based approach for the 10, 15 and 20m setting respectively. This beat the uniform weight baseline by 1.5 to 2 BLEU points and the strongest single feature (LASER) baseline by 1 BLEU point. The feature based approach performed slightly better with 15 candidate runs and beat the strongest single feature baseline (LASER) by 1.3 BLEU points.

6 Analysis

The following sections examine the learned weights, the effect of transferring them to noisy corpora of a different language pair and the method's performance when exposed to specific kinds of noise.

6.1 Learned Weights

Table 3 shows the weights learned using the *tuning* network, normalized to sum to one. Unsurprisingly, the strongest feature (dual cross-entropy) has the highest weight, with the sentence length ratio and IBM Model 1 (weak multi-lingual features) drawn for the next place while source and target LM have relatively low weights.

Feature	Weight	Feature
IBM Model 1	0.07	0.12
Source LM	0.03	0.02
Target LM	0.02	0.02
Dual xent	0.81	0.76
Sen. Length Ratio	0.07	0.08

Table 3: Feature weights learned post-tuning with the weight-based and the feature-based approaches. The weights have been normalized to sum to 1 (column).

6.2 Weight Transfer

Since the feature functions we use for our experiments are reasonably language-independent, a reasonable experiment is to see if the feature weights learned on one language-pair can be transferred to a noisy corpus of another another language pair. However, we hypothesize that unless the feature distributions (proxy for noise profile of the dataset) of the datasets are similar, this transfer will have limited success.

We test this hypothesis using the Maltese-English Paracrawl corpus. The training corpus contains 26.9 million sentence pairs and was sentence aligned using Vecalign and de-duplicated in a manner similar to our primary experiments. The validation and the test sets for these experiments are from the EUbookshop[6] dataset and contain 3k and 2.2k sentences respectively. The sentence level features were computed using the procedure described in section 4.2 and we use the DGT corpus[7] (about 1.6 million parallel sentences) to the train the clean translation models, the source and the target language models.

1-Feature Filtering Baselines	
Target LM	28.3
Source LM	27.1
Dual Cross-Entropy	**32.5**
Filtering using Transfer Weights	
Uniform weight baseline	30.5
Weight based	31.6
Feature based	31.3

Table 4: BLEU scores for the Maltese-English Paracrawl NMT systems where the training data was filtered using single features or a *transferred* (from Estonian-English) weighted combination of features.

[6] opus.nlpl.eu/EUbookshop.php
[7] data.europa.eu/euodp/en/data/dataset/dgt-translation-memory

The results of these experiments appear in Table 4. Even though filtering with the transferred weights beats the simpler single feature baselines, it fails to beat the strongest one, dual cross-entropy. It is worth noting that the reason filtering with the learned weights does this well is because the dual cross-entropy feature has the highest weight from our previous experiments. These experiments suggest that some form of feature distribution matching across corpora is required before weight transfer becomes viable.

6.3 Sensitivity to Noise Types

Inspired by Khayrallah and Koehn (2018), we look at how the most common noisy types in the Paracrawl data set affect the performance of the proposed method. For the purpose of these experiments, we use the *Europarl* v8 [8] Estonian-English data set. The training data set consists of about 651k parallel sentences, 11.2m source and 15.7m target tokens. We only use the feature-based method for this analysis and each experiment tunes weights based on 5 *candidate* runs.

We add synthetic noise to this data set by replacing 50% of the sentences in the data set to contain a specific kind of noise. The noise types we looked at and their perturbation methods are described below:

1. Misaligned sentences: Since parallel corpora extraction efforts use automated document and sentence alignment methods, noise includes source sentences which are not aligned to the correct target sentence. To emulate this, we randomly shuffle the source sentences of half the sentences in the clean data set.

2. Misordered words: A result of automatic or imperfect human translation, we add this noise to the clean data set by randomly shuffling the words within the source sentences.

3. Wrong language: This is a very common noise type in web-crawled corpora. We emulate it by performing lexical replacements (from Estonian to French).

4. Untranslated words: This other common noise type is added to our data set by copying the source sentence to the target.

[8] www.statmt.org/europarl

Noise Type	% Retained
Misaligned sentences	92
Misordered words	81
Wrong language	89
Untranslated words	78

Table 5: The portion of the clean sentences retained after perturbing 50% of the data set with specific noise types, learning feature weights and resampling the top 50% samples.

For each type of noise, we perform the following experiment: perturb 50% of the clean data with the chosen noise type, compute feature values for the sentences in the full data set, learn feature weights using the weight-based method described in section 3, filter out the top 50% of the data set and measure the percentage of *clean* (non-perturbed) sentences which were retained.[9] The results of this analysis appears in Table 5. The method performs significantly better than chance in all noise categories, but given our choice of features, it is better at filtering out misaligned sentences and sentences with tokens in the wrong language and is slightly less effective at dealing with misordered and untranslated words.

7 Future Work

The validation set based delta-perplexity is expensive to compute per update and replacing it with a more stable or time invariant reward (Wang et al., 2019) may help improve the performance of this method. Additionally, we plan to replace grid search with a more granular search procedure over the weight space with respect to the weight-feature-reward function. The tuning network can also be modified to include sentence-quality modulated loss functions (via feature values). An alternative to searching for feature weights is to instead search for the prototypical feature vector which maximizes translation performance and then use it to filter the closest sentence pairs from the noisy dataset. Finally, as discussed in Section 6.2, transferring learned weights has the potential to dramatically reduce the cost of applying to this method to new language pairs and may help with performance on low-resource language pairs where good feature weights cannot be learned.

[9] We note that the performance of this analysis depends on the chosen features. As an extreme example, if we perturb the source sentences and only consider a target-side feature (such as target language model scores), we will have no way of discriminating bad noisy samples from the clean ones.

8 Conclusion

We present a method for denoising and filtering noisy parallel data for improving the performance of neural machine translation systems. We learn interpolation weights for sentence-level features by modeling and searching over the weight-reward space. These are used to score and filter sentences in the noisy corpora. Our experiments with Estonian-English Paracrawl show gains of over a BLEU point over the strongest single feature filtering and uniform weight baselines. Analysis also shows that this method is effective at addressing the most common noise types in web-crawled corpora.

References

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, Portland, Oregon. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy. Association for Computational Linguistics.

Peter F. 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):263–311.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Muhammad N. ElNokrashy, Amr Hendy, Mohamed Abdelghaffar, Mohamed Afify, Ahmed Tawfik, and Hany Hassan Awadalla. 2020. Score combination for improved parallel corpus filtering for low resource conditions.

Grant Erdmann and Jeremy Gwinnup. 2019. Quality and coverage: The AFRL submission to the WMT19 parallel corpus filtering for low-resource conditions task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 267–270, Florence, Italy. Association for Computational Linguistics.

Miquel Esplà Gomis, Víctor M. Sánchez-Cartagena, Jaume Zaragoza-Bernabeu, and Felipe Sánchez-Martínez. 2020. Bicleaner at wmt 2020: Universitat d'alacant-prompsit's submission to the parallel corpus filtering shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 950–956, Online. Association for Computational Linguistics.

Jesús González-Rubio. 2019. Webinterpret submission to the WMT2019 shared task on parallel corpus filtering. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 271–276, Florence, Italy. Association for Computational Linguistics.

Viktor Hangya and Alexander Fraser. 2018. An unsupervised system for parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 882–887, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. 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, 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.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

Murathan Kurfalı and Robert Östling. 2019. Noisy parallel corpus filtering through projected word embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 277–281, Florence, Italy. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and

Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Zuzanna Parcheta, Germán Sanchis-Trilles, and Francisco Casacuberta. 2019. Filtering of noisy parallel corpora based on hypothesis generation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 282–288, Florence, Italy. Association for Computational Linguistics.

Spencer Rarrick, Chris Quirk, and Will Lewis. 2011. Mt detection in web-scraped parallel corpora. In *Proceedings of MT Summit XIII*. Asia-Pacific Association for Machine Translation.

Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani, and Hermann Ney. 2018. The RWTH Aachen University filtering system for the WMT 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 946–954, Belgium, Brussels. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. 2018. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Sukanta Sen, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Parallel corpus filtering based on fuzzy string matching. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 289–293, Florence, Italy. Association for Computational Linguistics.

Felipe Soares and Marta R. Costa-jussà. 2019. Unsupervised corpus filtering and mining. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Och, and Juri Ganitkevitch. 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.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Graham Neubig, and Jaime G. Carbonell. 2019. Optimizing data usage via differentiable rewards. *CoRR*, abs/1911.10088.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Monotonic Simultaneous Translation with Chunk-wise Reordering and Refinement

Hyojung Han[1*†], **Seokchan Ahn**[1*†], **Yoonjung Choi**[1], **Insoo Chung**[1†], **Sangha Kim**[1], **Kyunghyun Cho**[2]
[1]Samsung Research, Seoul, Republic of Korea
[2]New York University, New York, United States
hjhan@cs.umd.edu, seokchaa@uci.edu, yj0807.choi@samsung.com,
insoo.chung@tamu.edu, sangha01.kim@samsung.com, kyunghyun.cho@nyu.edu

Abstract

Recent work in simultaneous machine translation is often trained with conventional full sentence translation corpora, leading to either excessive latency or necessity to anticipate as-yet-unarrived words, when dealing with a language pair whose word orders significantly differ. This is unlike human simultaneous interpreters who produce largely monotonic translations at the expense of the grammaticality of a sentence being translated. In this paper, we thus propose an algorithm to reorder and refine the target side of a full sentence translation corpus, so that the words/phrases between the source and target sentences are aligned largely monotonically, using word alignment and non-autoregressive neural machine translation. We then train a widely used wait-k simultaneous translation model on this reordered-and-refined corpus. The proposed approach improves BLEU scores and resulting translations exhibit enhanced monotonicity with source sentences.

1 Introduction

Simultaneous interpretation is widely used in various scenarios such as cross-lingual communication between international speakers, international summits, and streaming translation of a live video. Simultaneous interpretation has a latency advantage over conventional full-sentence translation, i.e. offline translation, as it requires only partial sequence to start translating. However, as the source and target languages differ in word orders, there is a difficulty in simultaneous interpretation that does not exist in offline translation which translates only after the whole source sentence is received. For example, when dealing with language pairs that significantly differ in word order (e.g., between SOV language and SVO language), an interpreter may not receive sufficient information with partial sequence to start generating a translation that respects the natural order of the target language. One of the approaches to address this problem is to perform *anticipation*[1]. Note that the nature of anticipation relies on interpreters' assumptions and the anticipation may provide incorrect translations. Alternatively, human interpreters strategically resort to producing *monotonic translations* that follow the word order of the source sequence (Cai et al., 2020).

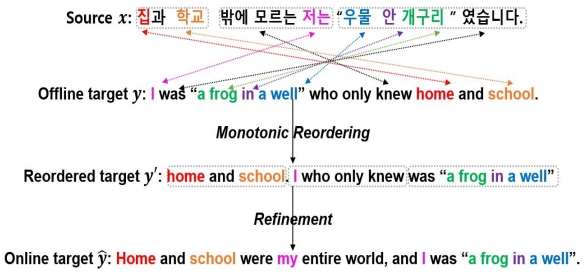


Figure 1: An example illustration of monotonic reordering and refinement for simultaneous translation

To illustrate the differences between the two strategies, the example in Figure 1 may be referred to. Of the two targets, offline target $\mathbf{y}$ respects the target language order and an online target $\hat{\mathbf{y}}$ roughly follows the source word ordering. Successful anticipation in Figure 1's case would be to predict the initial words in $\mathbf{y}$ (I was a "frog in a well") before receiving the full $\mathbf{x}$. This would pose difficulty even to professional translators as all the relevant information is in the latter part of the $\mathbf{x}$ (저는$_{I}$ /"우물$_{a\ well}$ / 안$_{in}$ / 개구리$_{a\ frog}$" 였습니다$_{was}$.). Bartłomiejczyk (2008) reports the success rate of human interpreters' anticipation attempts to be as low as 38.1% even though they make predictions based on pre-acquired domain knowledge. On the other hand, a monotonic approach would be to gen-

* Equal contribution
† Work done at Samsung Research

[1]A simultaneous interpretation strategy where the interpreter says information that is not yet said by the speaker.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1110–1123
November 10–11, 2021.

erate an $\hat{y}$ style translation - the grammaticality in the resulting sequence is sacrificed to translate only the received information.

A similar case applies to Simultaneous Machine Translation (SimulMT) models, which start translating before a whole sentence is given. Several studies (Ma et al., 2019; Arivazhagan et al., 2019; Ma et al., 2020) often utilize offline full-sentence translation corpora to train SimulMT models. Offline full-sentence parallel corpora are expected to follow the natural order of languages and mostly contain source to offline-target pairs. Naturally, when SimulMT models are trained on these corpora, the models inevitably learn to perform anticipation. Recent SimulMT studies are focused on reducing anticipation (Zhang et al., 2020a) or performing better anticipation (Zhang et al., 2020b). On the contrary, studies on enabling monotonic translation in SimulMT are scarcely available. Recently, Chen et al. (2020) suggest utilizing pseudo-references for monotonic translation.

In this paper, we propose a paraphrasing method to generate a monotonic parallel corpus to allow a monotonic interpretation strategy in SimulMT. Our method consists of two stages. The method first chunks source and target sequences into segments and monotonically *reorders* the target segments based on source-target word alignment information (Section 3.1). Then, the reordered targets are *refined* to enhance fluency and syntactic correctness (Section 3.2). To show the effectiveness of our method, we train wait-k models (Ma et al., 2019) on the resulting monotonic parallel corpus of reordering-and-refinement. Results show improvements in BLEU scores over baselines and models producing monotonic translations. Our main contributions are as follows:

- We propose a method to reorder and refine the target side in an offline corpus to build a monotonically aligned parallel corpus for SimulMT.
- We investigate the monotonicity in different language pairs, and show monotonicity can be improved after the reordering-and-refinement process.
- We train widely used wait-k models on generated monotonic parallel corpora in multiple language pairs. The results show improvements over baselines in both translation quality and monotonicity.

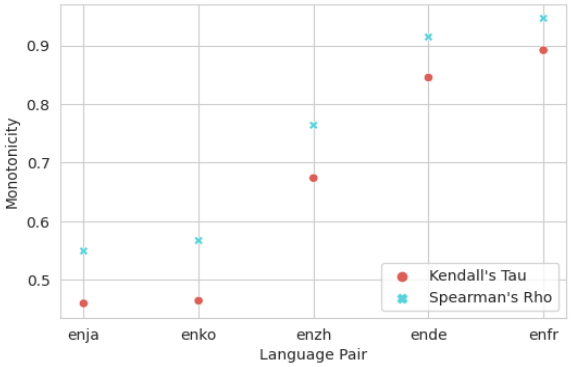


Figure 2: Monotonicity measured on offline trainsets. Utilized data is described in Section 4.1. As Kendall's τ and Spearsman's ρ show similar patterns, we only report Kendall's τ measurements in the rest of the paper.

2 Monotonicity Analysis

In this section, we analyze the degree of word order differences in multiple language pairs, i.e., the monotonicity in different language pairs. To measure the monotonicity, two rank correlation statistics are utilized: Kendall's τ and Spearman's ρ. The analyzed language pairs are: English-{Korean, Japanese, Chinese, German, French}.

According to Polinsky (2012), English is a head-initial language and Korean and Japanese are rigid head-final languages; Korean and Japanese are likely to exhibit extreme word order differences with English. German and Chinese are considered a mixture of head-final and head-initial languages; they are likely to have word differences with English, but not as severe as Korean or Japanese. French is also head-initial, so English and French pair is likely to have similar word order.

Figure 2 show monotonicity measurements between English and five different languages which vary in monotonicity: English-German and English-French pairs show high monotonicity, while English-Japanese and English-Korean pairs show low monotonicity.

Lower monotonicity in language pairs presents higher difficulties for SimulMT tasks. For example, wait-k algorithm only sees $k + t$ source tokens to generate a target token at step t which could lead to unwanted anticipation. To avoid such anticipation, as we mentioned in Section 1, human interpreters often provide a monotonic translation. In the same sense, we conjecture that promoting monotonicity in training corpora is beneficial for translation quality in SimulMT.

3 Monotonic Reordering and Refinement

In this section, we describe our proposed paraphrasing method of chunk-wise reordering and refinement to generate monotonic corpus for SimulMT. Given source $\mathbf{x} = \{x_1, x_2, \cdots, x_{|\mathbf{x}|}\}$ and offline full sentence target $\mathbf{y} = \{y_1, y_2, \cdots, y_{|\mathbf{y}|}\}$, an alignment $\mathbf{a}$ is defined as a set of position pairs of $\mathbf{x}$ and $\mathbf{y}$.

$$\mathbf{a} = \{(s, t) : s \in \{1, \cdots, |\mathbf{x}|\}, t \in \{1, \cdots, |\mathbf{y}|\}\}$$

First, in chunk-wise reordering phase, we generate source chunk set $\mathcal{C}^X$

$$\mathcal{C}^X = \{(x_{1:p_1}), (x_{p_1+1:p_2}), \cdots, (x_{p_{k-1}+1:p_k})\},$$

where $0 < p_1 < p_2 < \cdots < p_k = |\mathbf{x}|$ and reordered target chunk set $\mathcal{C}^Y$

$$\mathcal{C}^Y = \{(y'_{1:q_1}), (y'_{q_1+1:q_2}), \cdots, (y'_{q_{k-1}+1:q_k})\},$$

where $0 < q_1 < q_2 < \cdots < q_k = |\mathbf{y}|$, and $y'_i \in \mathbf{y}'$ is reordered target token from offline target $\mathbf{y}$. The elements of a reordered target chunk $\mathcal{C}^Y_i$ are corresponding target tokens of a source chunk $\mathcal{C}^X_i$ based on given alignment information $\mathbf{a}$. Also, we preserve the original target order within each $\mathcal{C}^Y_i$. For example, offline and reordered target in Figure 1 correspond to $\mathbf{y}$ and $\mathbf{y}'$ respectively, and both sequences are only different in token orders. The number of source chunks in one sentence is the same as the number of reordered target chunks ($|\mathcal{C}^X| = |\mathcal{C}^Y|$), while the number of tokens in $|\mathcal{C}^X_i|$ and $|\mathcal{C}^Y_i|$ could vary. We experiment two chunking methods; fixed-size chunking and alignment-aware adaptive size chunking.

Given chunked sets $\mathcal{C}^X$ and $\mathcal{C}^Y$, we refine reordered target tokens to generate more natural and fluent sentence with a Non-Autoregressive Translation (NAT) model. In the refinement algorithm, final paraphrased sentence $\hat{\mathbf{y}}$ is generated from reordered sequence $\mathbf{y}'$. Furthermore, we incorporate an Autoregressive Translation (AT) model into our refinement process. The more detailed steps for each phase will be explained in the following subsections.

3.1 Chunk-wise Reordering

3.1.1 Fixed-size Chunk Reordering

In the fixed-size chunk reordering method, we simply chunk a sequence of tokens into fixed size segments. The source chunk set $\mathcal{C}^X$ in this chunking method is as follows:

$$\mathcal{C}^X = \{(x_{1:K}), (x_{K+1:2K}), \cdots, (x_{[|\mathbf{x}|/K]:|\mathbf{x}|})\},$$

where $K \in [1, |\mathbf{x}|]$ is chunk size. If $k = 1$, $\mathcal{C}^X$ is identical with $\mathbf{x}$. We conduct subword operation such as sentencepiece or BPE after chunking process in order to avoid subword separation.

3.1.2 Alignment-Aware Chunk Reordering

In the alignment-aware chunking method, we segment a sentence adaptively by leveraging alignment information $\mathbf{a}$, as described in Algorithm 1. The left grid in Figure 3 presents the subword alignment between source and target sentence. We run aligner on subword over word because the alignment performance is consistently better Zenkel et al. (2020) when using GIZA++ (Och and Ney, 2003) , which we use in our experiments. Based on this alignment information, we initialize a list of chunks $\mathcal{C}$. As observable, there are some tokens which have no alignment information. To avoid omission, we assign the same alignment as the previous token; if a token is at the head, it follows the next token's alignment. To ensure subwords can be properly detokenized after reordering, we merge mid-splitted subwords. The middle grid in Figure 3 presents the result of these initialization steps. After initialization, we generate consistent chunks by merging all the inconsistent ones, following the definition of consistency in Zens et al. (2002). In a consistent chunk, tokens are only aligned to each other, not to tokens in other chunks. If any chunk in $\mathcal{C}$ has size smaller than a minimum size threshold δ, we merge a chunk pair that are adjacent in both source and target side and have the shortest target distance between them. If the distances are the same between multiple candidate pairs, we choose the pair of chunks that makes the smallest size after merging. We additionally merge the chunks adjacent to the merged one if they are arranged monotonically. Merging is repeated until all chunks meet the size requirements. An example of final result is the right grid in Figure 3. Phrase extraction method used in statistical machine translation Koehn (2004) also makes phrase level alignments from word alignments using heuristics like ours, but it tends to choose shorter phrases since the number of co-occurrences decrease drastically as the phrase size grows, which makes it difficult to generate larger chunks to prevent hurting grammatical correctness while reordering phase.

3.2 Refinement

Reordered target results from previous phase inevitably entail irregularities mainly for two rea-

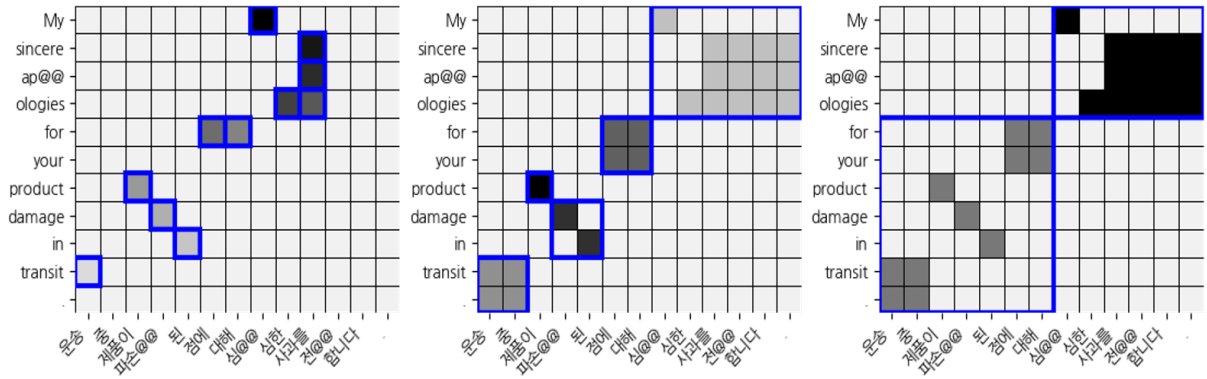


Figure 3: Example of alignment-aware reordering process. **Left**: Original alignments. **Center**: (Initialization) After filling align vacancy, merging mid-split subwords and enforcing the consistency requirement. **Right**: After merging all the chunks shorter than length thresholds.

Algorithm 1: Alignment-Aware reordering

Input: Source sentence $\mathbf{x}$ and target sentence $\mathbf{y}$
Output: Monotonically aligned chunks $\mathcal{C}'$

$\mathbf{a}$ = Alignment between $\mathbf{x}$ and $\mathbf{y}$
$\mathcal{C}$ = Initialize chunks $\mathcal{C}$
$\mathcal{C}$ = Merge all the inconsistent chunks in $\mathcal{C}$
while $|C_i^X| < \delta_{src}$ **or** $|C_i^Y| < \delta_{tgt}$ **for any** $\mathcal{C}_i$ **in** $\mathcal{C}$ **do**
  $\mathcal{C}_k$ = The smaller of the chunks adjacent to $\mathcal{C}_i$
  Merge $\mathcal{C}_i$ and $\mathcal{C}_k$
  Merge monotonic chunks adjacent to $\mathcal{C}_i$
end
$\mathcal{C}'$ = Reorder target side of $\mathcal{C}$ monotonically

Algorithm 2: Chunk-wise Refinement

Input: Source and target chunks $\mathcal{C}^X, \mathcal{C}^Y$
Output: Paraphrased target $\hat{\mathbf{y}}$

$i = 1$
$\hat{Y}_0 = []$
while $i \leq |\mathcal{C}^X|$ **do**
  $X_i = \mathcal{C}^X_{1:i}$ and $Y'_i = [\hat{Y}_{i-1}; \mathcal{C}^Y_i]$
  $\hat{Y}_i = \arg\max_Y \log p^{\mathcal{R}}(Y|X_i, Y'_i)$
  $i = i + 1$
end
Return: $\hat{Y}_{|\mathcal{C}^X|}$

sons. One could be broken connectivity of collocations in segmentation process. The other would be disfluently missing or containing words of endings and preposition as the position of chunk has been changed, thus requiring an addition of new words or clearing unnecessary words. In this part, we focus on refining aforementioned anormalities in order to enhance fluency, while preserving the monotonicity at the same time.

3.2.1 Refinement with NAT

We iteratively decode partial source $\mathcal{C}^X$ with pretrained translation model, given partial reordered target $\mathcal{C}^Y$ as a guidance in order to generate corresponding online target $\hat{Y}$. More specific process is explained in Algorithm 2. As the model refines given $[\hat{Y}_{i-1}; \mathcal{C}_i^Y]$, previous refined output $\hat{Y}_{i-1}$ could be altered as the model re-generates the entire sequence from scratch. Similarly in translation (Arivazhagan et al., 2020; Han et al., 2020b), we set an option of fixed or alterable prefix to force the model whether to generate same target prefix of $\hat{Y}_{i-1}$ or to allow the model to modify the prefix. As we limit the visibility of source information and iteratively generate target tokens with increasing source chunks, we expect the refinement model to generate monotonically aligned and paraphrased targets $\hat{\mathbf{y}}$ with enhanced fluency.

We use NAT architecture as the core refinement model $\mathcal{R}$. In NAT inference, the model's decoder is first given source features and fed an empty target sequence. Then the NAT decoder develops the empty sequence into a translation of the source sequence. This development is often iterative. Note that at every iteration step, the target sequence is refined - closer to the source sequence in meaning and become more fluent. This motivates us to utilize NAT architecture in our refinement process for monotonic-yet-disfluent sequences. In our approach, the NAT model starts refinement iteration with initialized tokens of previous output and

reordered target chunk Y_i', instead of an empty sequence. This target initialization act as a weak supervision to generate monotonically aligned target, which allow model to focus only on the fluency the reordered targets.

3.2.2 Incorporation of AT

Despite the aptness of NAT structure to our refinement phase, NAT model entails a performance degradation compared to AT model in the expense of speedup. Also, there exists repetition problem in NAT (Lee et al., 2018; Gu and Kong, 2021) which is generated in the process of multiple chunk-wise iterative refinement. In order to complement the aforementioned weaknesses of NAT decoding, we incorporate AT into our refinement process with NAT model.The final probability is computed jointly with the probability of AT and NAT model:

$$p^{\mathcal{R}}(Y|X_i, Y_i') \propto p^{AT}(Y|X_i)^{\alpha} \cdot p^{NAT}(Y|X_i, Y_i')^{(1-\alpha)}, \quad (1)$$

where $\alpha \in [0, 1]$ is hyper-parameter deciding the ratio between AT and NAT probability.

Size/L_{avg}	EnKo	EnJa	DeEn	EnZh
Train	3.4M/22	3.9M/12	4M/28	15.9M/27
Valid	800/19	4451/17	3000/25	4000/30
Test	1429/21	1194/17	2169/25	4000/30
AlignAw	3.1M	1.7M	2.2M	6.5M
er	0.85	1.15	0.98	1.12

Table 1: Data statistics and average of En token length L_{avg} of used corpus. AlignAw denotes number of pairs processed with alignment aware refinement. er denotes emission rates used in wait-k decoding. Token lengths and er are measured base on subwords counts.

4 Experiments

4.1 Dataset

In this section, we describe the utilized datasets. Detailed statistics are presented in Table 1. Utilized EnKo trainset and devset are created using in-house translation corpora while test scores are reported on IWSLT17 (Cettolo et al., 2017) EnKo testset. The DeEn trainset of WMT15 translation task (Bojar et al., 2015) is utilized. *newstest2013* is utilized as devset and *newstest2015* is used as testset. The EnJa trainset and validset are respectively the combination trainsets and validsets of KFTT (Neubig, 2011), JESC (Pryzant et al., 2018), TED (Cettolo et al., 2012). The trainset and validset are used as preprocessed and provided by the MTNT authors[2] (Michel and Neubig, 2018). Only the TED portion of testsets is used. For EnZh training UN Corpus v1.0 (Ziemski et al., 2016) is used. Trainset, devset, and testset follow the original splits. Monotonicity of EnFr in Figure 2 is measured on the WMT14 (Bojar et al., 2014) trainset. Additional details regarding utilized tokenization and vocabulary training are listed in Appendix A.

4.2 Metric

All the BLEU scores are cased-BLEU measured using sacreBLEU (Post, 2018). Test scores are measured using models that report best BLEU on their respective devsets. All references and translations of each Korean, Japanese, and Chinese languages are tokenized prior to BLEU evaluation. Tokenizers utilized are mecab-ko[3], KyTea [4], and jieba for Korean, Japanese and Chinese respectively. We report detokenized BLEU on DeEn results. To measure monotoniticy, we use Kendal's τ rank correlation coefficient.

4.3 Implementation Details

The default setting for NMT and SimulMT models follow the base configuration of transformer (Vaswani et al., 2017). SimulMT models are trained using wait-k algorithm, where $k \in \{4, 6, 8, 10, 12\}$, with uni-directional encoder similarly to Han et al. (2020a). The base NMT and SimulMT models are trained up to 300k train steps on a single GPU - each step is performed on a batch of approximately 12288 tokens. For refinement, we utilize NAT models of Levenshtein transformer architecture (Gu et al., 2019) with maximum iteration of 1. The NAT models are trained using sequence-level knowledge distillation (Kim and Rush, 2016) - the references of trainset pairs are replaced with beam search results ($beam = 5$) of NMT teachers. The NAT models follow base configurations and teacher NMT models follow big configuration. Both types of models are trained up to 300k steps on 8 GPUs. In each training step, a 8192 tokens batch is used per GPU. Additional implementation details can be found in the Appendix B.

[2] https://www.cs.cmu.edu/~pmichel1/mtnt/

[3] https://github.com/hephaex/mecab-ko

[4] http://www.phontron.com/kytea/

Wait-k BLEU	k=4	k=6	k=8	k=10	k=12
Offline	10.9	12.1	12.6	12.8	13.4
Fixed + NAT	11.2	12.3	**13.5**	**13.5**	**13.7**
AlignAw + NAT	11.4	**12.8**	12.7	12.9	13.1
AlignAw + NAT + AT ($\alpha = 0.25$)	**11.7**	12.6	12.4	12.7	13.2
AlignAw + NAT + AT ($\alpha = 0.50$)	10.7	12.4	13.0	13.0	13.2
Wait-k Kendal's τ	k=4	k=6	k=8	k=10	k=12
Offline	0.65526	0.60792	0.58440	0.55169	0.53701
Fixed + NAT	0.71822	0.66811	0.63154	0.61244	0.59478
AlignAw + NAT	0.71903	0.69101	0.67149	0.65985	0.64043
AlignAw + NAT + AT ($\alpha = 0.25$)	**0.73215**	0.69386	0.69524	0.67593	0.63335
AlignAw + NAT + AT ($\alpha = 0.50$)	0.73055	0.70106	0.67674	0.65559	0.64042

Table 2: BLEU scores and monotonicity measurements of EnKo wait-k models trained on offline and reordered-and-refined corpora. Note that monotonicity is measured between the model translations and testset references.

4.4 Corpus Generation and Training

We demonstrate the effectiveness of our reordering-and-refinement method by training wait-k models on the resulting datasets. The wait-k models are trained on the combination of the monotonically aligned training pairs and offline trainset. AlignAw + NAT + AT denotes monotonically aligned corpora using alignment-aware reordering and refinement using joint probability of NAT and AT models. And Offline refers to the offline full-sentence corpus.

4.4.1 Reordering

Fixed: For fixed-size reordering, we experiment with chunk sizes $K \in \{4, 6, 8, 10, 12\}$. In wait-$k$ training, k and K are matched. All fixed-size reordered-and-refined corpora have the same size as corresponding offline corpus.

AlignAw: For each corpus, we generate four variations of alignment-aware reordering with source and target minimum chunk size of $2, 3$. Alignment-aware reordering is not applicable on the already-monotonic cases and the sentence pairs which are locally non-monotonic inside a chunk and globally monotonic among chunks within a single pair - typically, the reordering method is applicable to 20% to 50% of offline corpus.

We gather unique pairs from the created four variations to generate the final reordered pairs. The statistics of reordered set for each translation direction is in Table 1. The resulting pairs are refined and combined with corresponding offline corpus to train wait-k models. Here, same set of reordered pairs are utilized for all k settings.

seq-rep-n	Offline	NAT	NAT + AT
1-gram	0.036	0.076	0.072
2-gram	0.008	0.022	0.018
3-gram	0.003	0.009	0.006
4-gram	0.001	0.004	0.002

Table 3: N-gram repetition rate measured on offline and reordered-and-refined EnKo corpora. $\alpha = 0.25$ is set to for NAT + AT

4.4.2 Refinement

NAT: NAT models are utilized to refine the reordered pairs. Both the fixed prefix and alterable prefix refinement is performed and combined. BertScore (Zhang et al., 2020c) is measured and used to discard refinement results that show below average scores. The size of the resulting set is the same as the corresponding offline corpus.

NAT + AT: NAT and AT models can both be utilized to jointly compute token probability in refinement (Section 3.2.2). The AT models utilized are the baseline wait-k models trained on offline corpora. We experiment with $\alpha \in \{0.25, 0.5\}$. The examples of reordered-and-refined sequences can be found in Appendix E.

5 Results and Analysis

5.1 Experimental Results on EnKo

Table 2 shows BLEU scores and Kendal's τs of wait-k models trained using original offline corpus and variations of reordered-and-refined corpus. We observe that the models trained on monotonically

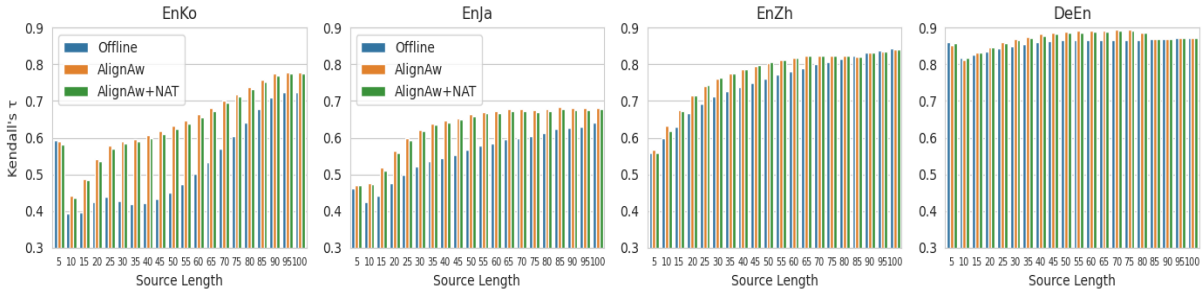


Figure 4: Monotonicity of offline pairs and pairs processed with reordering-and-refinement. AlignAw indicate that targets are alignment-aware reordering (Section 3.1.2, and AlignAw + NAT show monotonicity after NAT refinement (Section 3.2.1) is applied to AlignAw pairs.)

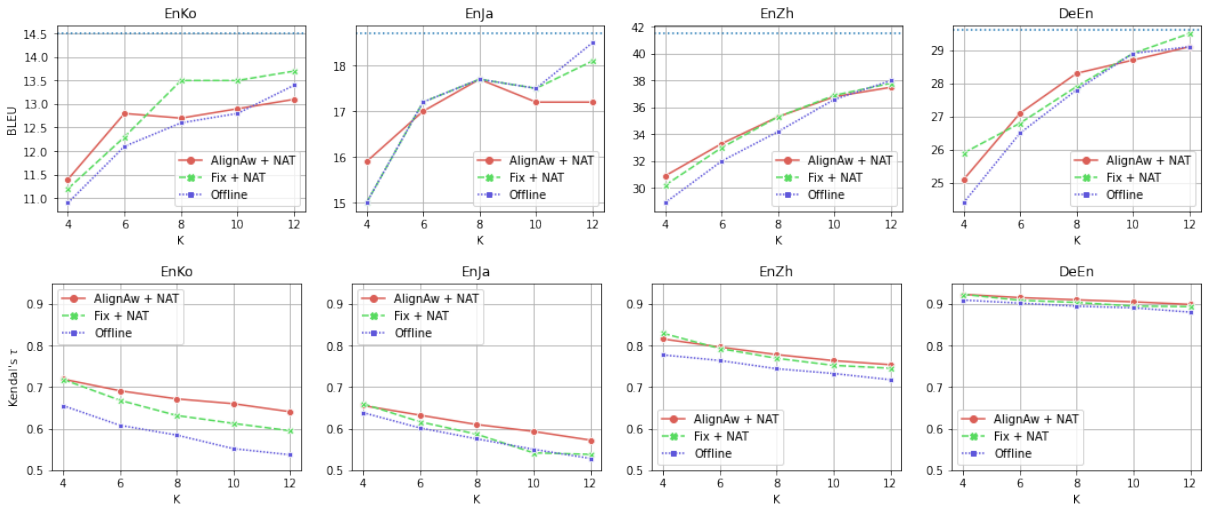


Figure 5: BLEU scores and monotonicity measurements presented by wait-k models trained on offline translation corpora and variations of reordered-and-refined corpora.

reordered-and-refined corpora show higher BLEU scores and monotonicity.

Reordering: Of the variations, corpora including AlignAw chunking process generally show better BLEU scores over Fixed + NAT when $k \leq 6$. This could be the benefit of the semantically plausible way to split sentences provided by AlignAw chunking. On the other hand, models trained with Fixed + NAT corpora show higher BLEU when $k \geq 8$.

Refinement: Experiments on utilizing AT probabilities show degraded BLEU scores in $k \in 6, 8, 10$. On the contrary, the models trained on AlignAw + NAT + AT corpora show enhanced monotonicity. The α value may be adjusted to make trade-off between promoting monotonicity in translation or enhancing translation quality in terms of BLEU.

Repetition Reduction with AT: Following (Welleck et al., 2020), we report n-gram repetition rate, seq-rep-n, on each generated corpus in Table 3. We observe from seq-rep-n in all of the tested n values, that employing AT models in refinement help alleviating the repetition problem of posed by NAT models.

5.2 Language Pairs Comparison

Figure 4 shows the difference in monotonicity between different language pairs: EnKo, EnJa, EnZh, and DeEn. It is observable in Figure 4 that the overall monotonicity in EnKo and EnJa pairs is enhanced after paraphrasing, while monotonicity scores of DeEn remain almost the same, only showing slight improvement. The extent of monotonicity enhancement in EnZh is between that of EnKo/Ja and DeEn. In all language pairs, the enhancements are generally lower in long or very short sequences. In the case of long sequence pairs, a pair may contain multiple sequences and be already aligned at the sequence level, thus resulting in marginal monotonicity enhancement. In the case of shorter length sequences, the whole sentence

may be merged into a single chunk, less benefiting from our process. After the reordered sets are refined, monotonicity marginally decreases. This is expected as forcibly aligned tokens are refined to augment the fluency in the resulting sentence.

To present the effectiveness of generated monotonic corpus in different language pairs, we train wait-k models on EnKo, EnJa, EnZh, and DeEn, and report BLEU and Kendal's τ of the models in Figure 5. The horizontal dotted line presents the BLEU of unidirectional offline model. The important observation we can find is that the monotonicity increment of wait-k model in Figure 5 is proportional to that of generated monotonic corpus in Figure 4, suggesting that promoting monotonicity in training corpus is beneficial for SimulMT models to generate monotonic output, especially in language pairs with differing word orders.

Within two paraphrasing methods, Fixed + NAT and AlignAw + NAT , we can see that the monotonicity of Fixed + NAT is always in between that of Offline and AlignAw + NAT in Figure 5, and the gap increases as the k value get higher.

While our methods are effective in EnKo, the performance of suggested method is similar or lower than that of baseline in EnJa. We presume that the ineffectiveness in EnJa is due to its short average sentence length with highest emission rate, as shown in Table 1. A short sentence often cannot preserve semantic properties while being split into chunks and reordered. For example, Fixed-length reordering always chunks all sentences ignores such feature and only increases disfluency in the chunked result. Also, even though AlignAw enforces the consistency requirement on the chunks, adjustment of such requirement like changing minimum chunk size may be required considering the high emission rate.

Based on the highest performance at $k = 8$, there is about 8% BLEU improvement over Offline in EnKo whereas there is about 3% improvement in EnZh and about 2% improvement in DeEn. It is roughly proportional to the monotonicity improvements shown in Figure 4.

5.3 Evaluation on Online References

We test wait-k models on our in-house EnKo online and offline testsets of 150 lines. We choose EnKo because the impact of reordering-and-refinement is the greatest in that pair. The source sentences of both testsets are identical. The online references

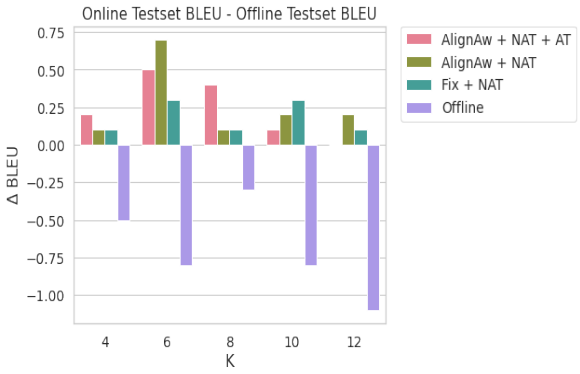


Figure 6: Differences of BLEU score between online and offline of EnKo wait-k models.

are constructed by a professional interpreter under a simulated simultaneous interpretation scenario and the offline references are constructed by the same interpreter assuming a typical translation scenario. In construction of online references interpreter was encouraged to perform monotonic interpretation rather than anticipation. BLEU scores are computed with both online and offline references for each trained model. Figure 6 plot the subtraction of BLEU scores on offline references from BLEU scores on online references. It is noticeable that the wait-k models trained on offline corpus have negative value while all the models trained on generated corpus present positive values, which implies the effectiveness of our approach. Overall, the substantial differences at $k = 6$ may suggest that the chunk size utilized by human interpreter has comparable value.

6 Related Work

Due to word order differences between languages, SimulMT training often face situations where anticipation is required. Note that word order difference is observed to be problematic even for human interpreters (Al-Rubai'i, 2004; Tohyama and Matsubara, 2006). Chen et al. (2020) suggest using pseudo-references which involve utilizing wait-k inference output to limit "future anticipation" in training. Zhang et al. (2020b) utilize NMT teachers to implicitly embed future information in their SimulMT students for better anticipation performance. Zhang et al. (2020a) study adaptive policy to tackle this problem - authors suggest an adaptive SimulMT policy that dictate READ/WRITE actions based on whether "meaningful units" are fully formed with consumed input tokens.

Related work in the broader SimulMT and para-

phrasing domain is presented in Appendix F.

7 Conclusion

Most of SimulMT models are trained on offline translation corpora, which could lead to limitation in translation quality and achievable latency, especially in non-monotonic language pairs. To address this problem, we propose a reordering-and-refinement algorithm to generate monotonically aligned online target with NAT model. We then train widely used wait-k SimulMT models on this newly generated corpus. Resulting models show BLEU score improvement and significant enhancement on monotonicity in multiple language pairs.

References

Alya Al-Rubai'i. 2004. The effect of word order differences on english-into-arabic simultaneous interpreters' performance. *Babel*, 50:246–266.

Ashkan Alinejad, Maryam Siahbani, and Anoop Sarkar. 2018. Prediction improves simultaneous neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3022–3027, Brussels, Belgium. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Magdalena Bartłomiejczyk. 2008. *Anticipation: A controversial interpreting strategy*, pages 117–126.

Ondřej 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, 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.

Ozan Caglayan, Julia Ive, Veneta Haralampieva, Pranava Madhyastha, Loïc Barrault, and Lucia Specia. 2020. Simultaneous machine translation with visual context. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2350–2361, Online. Association for Computational Linguistics.

Zhongxi Cai, Koichiro Ryu, and Shigeki Matsubara. 2020. What affects the word order of target language in simultaneous interpretation. In *2020 International Conference on Asian Language Processing (IALP)*, pages 135–140.

M. Cettolo, Marcello Federico, L. Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Rakesh Chada. 2020. Simultaneous paraphrasing and translation by fine-tuning transformer models. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 198–203, Online. Association for Computational Linguistics.

Junkun Chen, Renjie Zheng, Atsuhito Kita, Mingbo Ma, and Liang Huang. 2020. Improving simultaneous translation with pseudo references. *arXiv preprint arXiv:2010.11247*.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation 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 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. 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.

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*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, Online. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pages 11181–11191.

Hou Jeung Han, Mohd Abbas Zaidi, Sathish Reddy Indurthi, Nikhil Kumar Lakumarapu, Beomseok Lee, and Sangha Kim. 2020a. End-to-end simultaneous translation system for IWSLT2020 using modality agnostic meta-learning. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 62–68, Online. Association for Computational Linguistics.

Hyojung Han, Sathish Indurthi, Mohd Abbas Zaidi, Nikhil Kumar Lakumarapu, Beomseok Lee, Sangha Kim, Chanwoo Kim, and Inchul Hwang. 2020b. Faster re-translation using non-autoregressive model for simultaneous neural machine translation. *arXiv preprint arXiv:2012.14681*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. Simulated multiple reference training improves low-resource machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–89, Online. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Machine Translation: From Real Users to Research*, pages 115–124, Berlin, Heidelberg. Springer Berlin Heidelberg.

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, page 177–180, USA. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020. Monotonic multihead attention. In *International Conference on Learning Representations*.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

M. Polinsky. 2012. *Headedness, again*, pages 348–359. UCLA Department of Linguistics, Los Angeles.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. 2018. JESC: Japanese-English Subtitle Corpus. *Language Resources and Evaluation Conference (LREC)*.

Hitomi Tohyama and Shigeki Matsubara. 2006. Collection of simultaneous interpreting patterns by using bilingual spoken monologue corpus. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. 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.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. In *KI 2002: Advances in Artificial Intelligence*, pages 18–32, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020a. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289, Online. Association for Computational Linguistics.

Shaolei Zhang, Yang Feng, and Liangyou Li. 2020b. Future-guided incremental transformer for simultaneous translation. *arXiv preprint arXiv:2012.12465*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020c. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020a. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy. Association for Computational Linguistics.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, and Liang Huang. 2020b. Opportunistic decoding with timely correction for simultaneous translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 437–442, Online. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0.

A Dataset Details

All utilized texts regarding English-to-Korean and German-to-English directions are first tokenized with Moses (Koehn et al., 2007), then per-language BPE vocabularies are learned on the moses-tokenized trainset. The sizes of the vocabularies are 29k BPE English vocabulary and 44k BPE Korean vocabulary for English-to-Korean and 16k BPE German vocabulary and 16k BPE English vocabulary for German-to-English. The English-to-Japanese texts are first moses-tokenized. And KyTea[5] is applied to additionally tokenize Japanese texts. Separate English and Japanese vocabulary of size 32K is trained on tokenized training data using Sentencepiece (Kudo and Richardson, 2018). For English-to-Chinese training, no moses-tokenization is applied and Chinese sentences are tokenized using Jieba[6]. Separate English and Chinese vocabulary of size 32K is trained on training pairs using sentencepiece. The in-house EnKo data consists mainly of the AIHub EnKo offline translation corpus[7], news domain translation data, in-house proprietary patent data, and translated dialogue data of general domain.

B Additional Implementation Details

Our implementation is based on fairseq (Ott et al., 2019), and all GPUs used are V100s. The alignment information used in reordering process is extracted with GIZA++ (Och and Ney, 2003). The alignment information used to evaluate monotonicity is extracted using fast-align (Dyer et al., 2013). The alignments are measured in subword level. Embedding weights are separately learned for source and target languages, while transposed target language embedding weights also works as linear projection layers at the top of transformer decoders.

C Reporting AlignAw + NAT Scores

The AlignAw + NAT wait-k models are trained on different variations of AlignAw + NAT corpora - AlignAw + NAT corpora generated with prefixes fixed (**b0**), and with alterable prefixes (**b1**), and combination of **b0** and **b1** filtered using BertScore (**b0b1**). The reported AlignAw + NAT testset BLEU scores are of the wait-k models that show highest BLEU score on validset regardless of the dataset variations.

[5] http://www.phontron.com/kytea
[6] https://github.com/fxsjy/jieba
[7] https://aihub.or.kr

EnKo Wait-k	k=4	k=6	k=8	k=10
Offline	10.9/18	12.1/12	12.6/7	12.8/4
Pseudo Refs	11.2/19	11.6/13	12.2/9	13.3/5
Fixed + NAT	11.2/12	12.3/8	13.5/5	13.7/3
AlignAw + NAT	11.4/9	12.8/6	12.7/3	12.9/2

Table 4: BLEU scores/k-AR% of EnKo wait-k models.

D Comparison with Test Time wait-k Refs

In recent work, Chen et al. (2020) propose a method of pseudo-references generated with test time wait-k decoding. We apply their method in EnKo to create pseudo references for $k \in \{4, 6, 8, 10\}$ and train wait-k model. The results are presented in Table 4. Similar to our monotonicity metric, this work also suggest k-anticipation rate (k-AR) as a metric of parallel corpora. We also measure and report our generated corpus with this metric. Compare to Offline and Pseudo Refs , we see that our AlignAw + NAT corpus significantly decrease k-AR and the models trained with AlignAw + NAT also show enhanced BLEU score in general.

E Examples of Paraphrased Targets

Figure 7 presents an example sentence of English to Korean in whole pipeline process. We first represent the source sentence and its two different target sentences, online and offline translation. As results of the reordering phase, for each method (i.e., fixed chunking and AlignAw chunking), we provide only one case: $k = 8$ in fixed chunking and $\delta_{src} = 2$ and $\delta_{tgt} = 2$ in AlignAw chunking. Figure 8 shows the final grid of AlignAw chunking in this example. We conduct the MOS evaluation with the result of refinement phase. MOS is the average of human-evaluated score by professional interpreters. In this evaluation, AlignAW + NAT shows the best performance than others. Moreover, we present the inference outputs of SimulMT models which are trained on generated monotonic corpus. In this case, results of our methods are better than the result of offline model. We also provide DeEn example in Figure 9.

F More Related Work

Simultaneous Translation: A fixed policy is used in (Dalvi et al., 2018) and (Ma et al., 2019) which train SimulMT models according to the predefined policy. In particular, the Wait-k strategy

proposed by (Ma et al., 2019) waits for k sub-words and alternates READ/WRITE based on the emission rate. Due to the deterministic feature of this schedule, the model can be easily implemented and trained. On the downside, anticipation from missing contents often fails to predict correct target tokens, and a fixed schedule could impede the model from speeding up or slowing down flexibly for source inputs. There are several works of SimulMT with many variants of the Wait-k approach. For example, (Caglayan et al., 2020) explores whether additional visual context can complement missing source information. Furthermore, in (Zheng et al., 2020b), the opportunistic decoding technique is introduced which allows partial (certain length of suffix) corrections in a timely fashion. Finally, (Zheng et al., 2020a) extended the wait-k to an adaptive one by composing a set of fixed policies heuristically.

Upon the proposal by (Cho and Esipova, 2016), various adaptive policies have been suggested by several works including (Gu et al., 2017; Zheng et al., 2019a,b; Arivazhagan et al., 2019; Ma et al., 2020). SimulMT proposed by (Cho and Esipova, 2016) use greedy decoding with heuristic waiting criteria to decide whether the model should read or emit, while (Gu et al., 2017) utilize a pre-trained model with a reinforcement learning agent that maximizes quality and minimizes latency. Advancing this work, (Alinejad et al., 2018) proposes to add a new action PREDICT that anticipate future source words. Recently, (Arivazhagan et al., 2019) use hard attention to schedule the policy and introduced new differentiable average lagging metrics which can be integrated into training losses, and (Ma et al., 2020) incorporate this work into the multi-headed Transformer model. Furthermore, (Zhang et al., 2020a) proposes an adaptive policy which learns to segment source input considering possible target output. Other researches including (Zheng et al., 2019a) use separately trained oracles in the supervision of extracted action sequence.

Paraphrase: Translation is one of more common approaches for paraphrase generation. Mallinson et al. (2017) explore pivoting (translating a source sequence to a pivot language, then to a target language) to generate paraphrases and assess correlation between original and paraphrased sentences. Back-translation has also been explored for paraphrase generation (Wieting et al., 2017; Iyyer et al., 2018). Other techniques, such as translating with oversampling strategy have also bee studied (Chada, 2020).

On the other hand, various NMT research employ paraphrased data to overcome data limitation. Edunov et al. (2018) show that source-paraphrased corpus generated with back-translation can significantly improve BLEU scores in NMT tasks. Similarly, Khayrallah et al. (2020) directly implements paraphrasers in NMT training to improve translation quality.

Src-Tgt Pair			
Source		So *I* want to talk about that magic that happens when we actually represent something .	
Online		그래서 *제가* 얘기하고_싶은 것은 마@@ 법에 대한 이야기@@ 에요 . 우리가 실제로 어떤 것을 표현할 때 일어나는 그 마@@ 법 말입니다 .	
Offline		그래서 *나는* 우리가 실제로 어떤 것을 표현할 때 일어나는 마@@ 법에 대해 이야기@@_하려고 한다 .	
Reordering			
Fix	k=6	So *I* want to talk about ‖ that magic that happens when we ‖ actually represent something .	
	k=6	그래서 *나는* 대해_이야기@@ 하려고 ||| 우리가 것을 때 일어나는 마@@ ||| 실제로 것을 표현할 .	
AlignAw	src min 2	So *I* ‖ want to talk about ‖ that magic that happens when we actually represent something .	
	tgt min 2	그래서 *나는* ||| 대해 이야기@@_하려고 한다 . ||| 우리가 실제로 어떤 것을 표현할 때 일어나는 마@@ 법에	
Refinement			MOS(5/5)
Fix + NAT (K=6)	fixed prefix	그래서 *나는* 이 것에 대해 이야기@@ 하고자 할 때 일어나는 마@@ 법에 대해 말하고 싶습니다 . 우리가	2
	alterable	그래서 *나는* 그 마@@ 법에 대해 ~~그 마@@ 법에 대해~~ 이야기@@ 하고자 합니다 . 우리가 실제로 무언가를 나타@@ 낼 때 말이죠 .	2.75
AlignAw + NAT	fixed prefix	그래서 *나는* 이것에 대해 이야기@@ 하려고 한다_. 우리가 실제로 어떤 것을 표현할 때 일어나는 그 마@@ 법에 대해 말입니다	3.5
	alterable	그래서 *나는* 이것에 대해 이야기@@ 하려고 한다 . 우리가 실제로 어떤 것을 표현할 때 일어나는 마@@ 법에 대해 말입니다	4
SimulMT Wait-6			
Offline		그래서 *나는* ~~아 일아~~ ~~일어@@ 나면~~ 실제로 우리가 무언@@ 가를 대표@@ 할 때 일어나는 마@@ 법에 대해 이야기@@ 하고 싶습니다 .	2.5
Fix + NAT		그래서 *나는* 그 마@@ 법에 대해 이야기@@ 하고 싶습니다 . 우리가 실제로 무언@@ 가를 대표@@ 할 때 일어나는 마@@ 법에 대해 말하고 싶습니다 .	3.5
AlignAw + NAT		그래서 *저는* 그 마@@ 법에 대해 이야기@@ 하고 싶습니다 . 그래서 저는 실제로 어떤 것을 대변@@ 할 때 일어나는 것입니다 .	4
AlignAw + NAT + AT		그래서 *저는* 그 마@@ 법에 대해 이야기@@ 하고 싶습니다 . 우리가 실제로 무언@@ 가를 나타@@ 낼 때 일어나는 마@@ 법에 대해 말하고 싶습니다 .	4.5

Figure 7: Example sentence of EnKo in whole pipeline process from inputs to SimulMT results.

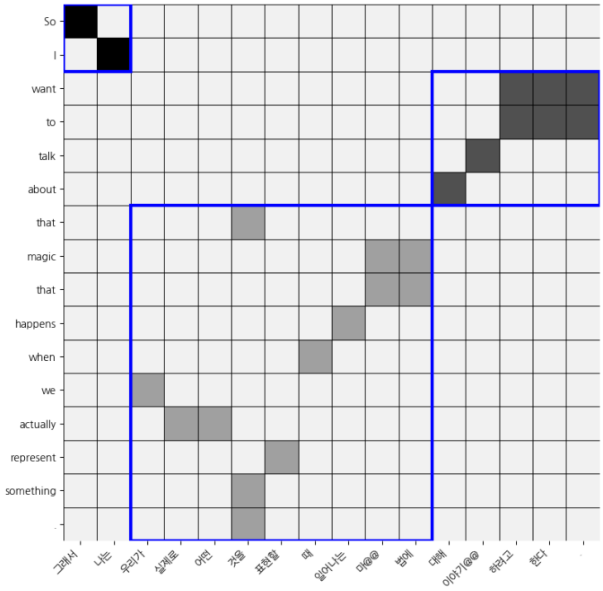


Figure 8: Alignment-aware result of the EnKo pipeline example in Figure 7

Src-Tgt Pair		
Source	Wie bei den meisten Kreb@@ ser@@ krank@@ ungen ist der genaue Auslö@@ ser meist schwer zu bestimmen .	
Offline	With most can@@ c@@ ers , it is hard to know the exact cause .	
SimulMT Wait-6		MOS(5/5)
Offline	As with most can@@ c@@ ers , the exact ~~extent of the action~~ is usually difficult to determine .	2
Fix + NAT	As with most can@@ c@@ ers , the exact ~~exact~~ trigger for the disease is usually difficult to determine .	4
AlignAw + NAT	As with most can@@ c@@ ers , the exact ~~spread~~ of the disease is usually difficult to determine .	3

Figure 9: Example outputs of DeEn wait-k models trained on reordered-and-refined corpora

Simultaneous Neural Machine Translation with Constituent Label Prediction

Yasumasa Kano[1] **Katsuhito Sudoh**[1,2] **Satoshi Nakamura**[1,2]
[1]Nara Institute of Science and Technology (NAIST), Japan
[2]Center for Advanced Intelligence Project (AIP), RIKEN, Japan
{kano.yasumasa.kw4, sudoh, s-nakamura}@is.naist.jp

Abstract

Simultaneous translation is a task in which translation begins before the speaker has finished speaking, so it is important to decide when to start the translation process. However, deciding whether to read more input words or start to translate is difficult for language pairs with different word orders such as English and Japanese. Motivated by the concept of pre-reordering, we propose a couple of simple decision rules using the label of the next constituent predicted by incremental constituent label prediction. In experiments on English-to-Japanese simultaneous translation, the proposed method outperformed baselines in the quality-latency trade-off.

1 Introduction

Simultaneous machine translation is a task in which the machine starts outputting a translation before reading the entire input sentence. This task is more difficult than full-sentence translation because it translates the initial part of a sentence without the context of the latter part. This involves a trade-off between delay and quality of the translation; using a longer context should improve translation quality at the cost of a longer delay, and vice versa. In practice, we should control the latency so that it's not too large, but we may also need to allow a long latency depending on the situation.

Most of the recent simultaneous translation models (Ma et al., 2019; Arivazhagan et al., 2019; Raffel et al., 2017; Arivazhagan et al., 2019; Ma et al., 2020b; Dalvi et al., 2018; Gu et al., 2017; Alinejad et al., 2018; Cho and Esipova, 2016; Zheng et al., 2020, 2019; Zhang et al., 2020) are based on neural machine translation (NMT), although earlier studies were based on statistical machine translation (Rangarajan Sridhar et al., 2013; Grissom II et al., 2014; Oda et al., 2014, 2015). In simultaneous NMT, there are two major approaches: those in which a latency hyperparameter is given before the training and those in which it is given at the time of inference.

The former approach requires training a model individually for each pre-defined latency setting, while the latter approach uses a single model for different latency conditions. Most human simultaneous interpreters would not need such long training to slightly adjust latency, while it takes much more time to learn other languages to develop their translation skill. Therefore, the latter approach is closer to the learning process of human simultaneous interpreters.

wait-k (Ma et al., 2019) is a simple simultaneous NMT method of the former approach that waits k tokens before starting to translate. It also has variants within the latter approach called test-time wait-k, in which k is determined at the inference time. wait-k had better performance than test-time wait-k in that study's experiments.

There is another method in the latter approach that uses Meaningful Unit (Zhang et al., 2020). In this model, chunk-based incremental decoding is done at inference time by segmentation with a boundary predictor. This model outperformed baselines of the former approach. They refined their basic boundary predictor to deal with sentence pairs in which full-sentence translation needs long-distance reordering. However, its training process is very complicated: It first generates monotonic translations, fine-tunes the NMT model with them, then generates an oracle boundary with the model, and finally fine-tunes a boundary-prediction model based on BERT (Devlin et al., 2019).

Simultaneous translation is still difficult for language pairs such as English-Japanese, which often require long-distance reordering. To tackle the reordering problem, we propose an input-segmentation method for simultaneous translation, using a couple of simple rules and incremental prediction of the label of a syntactic constituent coming immediately after the input existing so far. This

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1124–1134

Source sentence	I **bought** *a pen*.
Monotonic translation	watashi wa **katta** *pen wo*.
Full-sentence translation	watashi wa *pen wo* **katta**.

Table 1: Translation from English (SVO) to Japanese (SOV)

Boundary prediction	I / **bought** *a pen*.
Simultaneous translation	watashi wa / *pen wo* **katta**.

Table 2: Example of English-to-Japanese translation using proposed method with segment-boundary prediction

is not dependent on the trained NMT. Therefore, once we create it, it is reusable for other models.

Our proposed method is inspired by Head Finalization (Isozaki et al., 2010). Head Finalization reorders words of the source sentence before translating from an SVO (Subject-Verb-Object) language to an SOV language in full-sentence statistical machine translation. This method moves a syntactic head into a later position so that the word order of the source language (e.g., English) becomes similar to that of the target language (e.g., Japanese). This enables us to monotonically translate from English, which is a typical SVO language, to Japanese, a typical SOV language.

Recent NMT models like Transformer (Vaswani et al., 2017) works well on reordering in general, so this kind of pre-reordering is not usually used. However, simultaneous translation monotonically reads input words one by one, and therefore the difference in word order remains a problem. As shown in Table 1, monotonic translation often becomes unnatural compared to full-sentence translation. The part "bought a pen" should be translated to *pen wo katta* by reversing the word order. Therefore, after reading the word "bought," it is important to wait for future words without starting to translate it. In this case, "I" is the last word that does not require reordering. This word is regarded as a segment boundary to start a partial translation.

Table 2 shows an example of our proposed segmentation. Suppose we predict the next constituent label as a verb phrase (VP) after reading an input word "I." This shows the possibility that the next words should be reordered, so the "I" becomes the boundary. Once detecting the boundary, NMT model starts to translate "I" into "watashi wa." After that, the model restarts to read the remaining input words, then translates "bought a pen" into "pen wo katta." The total output of simultaneous translation based on the proposed segmentation is the same as that of full-translation in this simple example. By ensuring that the Verb and its Object of the source sentence are included in a single segment, it is possible to output translation while maintaining the SOV-like structure of the target language.

In experiments on English-to-Japanese simultaneous translation, the proposed method outperformed baselines in the quality-latency trade-off.

2 Related work

In statistical machine translation, there are several approaches to finding boundaries of segments for simultaneous translation. Oda et al. (2014) proposed a method to choose segment boundaries that maximize the BLEU score. Rangarajan Sridhar et al. (2013) proposed segmentation strategies based on lexical cues.

In NMT, there have been many studies on simultaneous translation. The amount of latency is decided either before training or at inference time. wait-k (Ma et al., 2019) is the simplest variant using fixed latency: It simply waits for k tokens before starting translation (Ma et al., 2019). The latency policy can be learned from a parallel corpus together with an NMT model. MILk (Arivazhagan et al., 2019) and other approaches (Raffel et al., 2017; Ma et al., 2020b) used a latency-augmented loss function in training to balance latency and accuracy.

In contrast, the latency policy can be learned with a pre-trained NMT model, such as test-time wait-k (Ma et al., 2019) and STATIC-RW (Dalvi et al., 2018). These have fixed policies that wait for the fixed number of tokens before translation, but there are other models that learn a more flexible policy for a given pre-trained NMT model. Some studies use reinforcement learning to learn an adaptive READ/WRITE policy (Grissom II et al., 2014; Satija and Pineau, 2016; Gu et al., 2017; Alinejad et al., 2018). Training by reinforcement learning can be unstable depending on the condition. One

method that does not use reinforcement learning is wait-if-* (Cho and Esipova, 2016), which translates and segments jointly to maximize the translation quality. Zheng et al. (2020) extended wait-k to an adaptive policy by adaptively choosing the strategy at inference. There is another method that generates oracle READ/WRITE actions by a pre-trained NMT model and predicts actions using a neural network model (Zheng et al., 2019). Meaningful Unit (Zhang et al., 2020) works along the same lines and has outperformed baselines such as MILk and wait-k.

With respect to the use of syntactic clues for simultaneous translation, Oda et al. (2015) proposed a method to incrementally parse an incomplete sentence by predicting unseen syntactic constituents on the right and left side of each segment. They concatenated the predicted constituents and the words in a segment and then input the result into tree2string translation. They decided to wait for more tokens or output the translation depending on where the constituents appear in the translation result.

Our proposed method is based on chunk-based simultaneous translation using chunk boundary detection with simple rules on next-constituent labels. It basically segments an input before a verb phrase. This is much simpler and easier to implement than the work by Zhang et al. (2020) and Oda et al. (2015).

3 Proposed Method

Figure 1 shows a step-by-step example of our proposed method described in this section.

3.1 Standard Simultaneous Translation

A standard NMT for full sentences is represented by the following equation:

$$p_{full}(Y|X) = \prod_{t=1}^{|Y|} P(y_t|X, y_{<t}), \quad (1)$$

where $X = x_1, x_2, ..., x_n$ is an input sentence consisting of n tokens and $Y = y_1, y_2, ..., y_m$ is a predicted target language sentence consisting of m tokens.

A simultaneous NMT uses only a prefix of the input to predict a target language token:

$$p_{simul}(Y|X) = \prod_{t=1}^{|Y|} P(y_t|x_{g(t)}, y_{<t}), \quad (2)$$

where $g(t)$ is a monotonic non-decreasing function representing the number of read source tokens to output the tth target token.

3.2 Chunk-based Simultaneous Translation

We use chunk-based incremental decoding for our simultaneous translation model and a full-sentence NMT model trained in a standard manner. However, at the time of inference, we translate the current prefix upon chunk segmentation while keeping the previously translated output unchanged.

Suppose we have already translated input chunks $\boldsymbol{X}^{i-1} = X_1, X_2, ..., X_{i-1}$ into an output prefix also represented by chunks: $\widetilde{\boldsymbol{Y}}^{i-1} = \widetilde{Y}_1, \widetilde{Y}_2, ..., \widetilde{Y}_{i-1}$, while translating the next input chunk X_i into $\widetilde{Y}_i$. We restart the translation from the beginning using all of the available input chunks X_1^i. This is similar to an approach called *re-translation* that generates translations from scratch for every new input word (Niehues et al., 2016; Arivazhagan et al., 2020), but we apply forced decoding to $\widetilde{\boldsymbol{Y}}^{i-1}$ in the output prefix. The probability of the prefix $\widetilde{\boldsymbol{Y}}^{i}$ can be denoted as follows:

$$p_{prefix}(\widetilde{\boldsymbol{Y}}^i|\boldsymbol{X}^i) = p_{full}(\widetilde{\boldsymbol{Y}}^{i-1}|\boldsymbol{X}^i) \times p_{chunk}(\widetilde{Y}_i|\boldsymbol{X}^i, \widetilde{\boldsymbol{Y}}^{i-1}). \quad (3)$$

The first term is calculated in the same way as the standard full-sentence NMT in Eq. (1) through forced decoding, and the second term is decomposed as follows, letting $\widetilde{Y}_i = y_1^i, y_2^i, ..., y_{|\widetilde{Y}_i|}^i$:

$$p_{chunk}(\widetilde{Y}_i|\boldsymbol{X}^i, \widetilde{\boldsymbol{Y}}^{i-1}) = \prod_{t=1}^{|\widetilde{Y}_i|} P(y_t^i|\boldsymbol{X}^i, \widetilde{\boldsymbol{Y}}^{i-1}, y_{<t}^i). \quad (4)$$

This can be more efficient than an incremental Transformer (Ma et al., 2019) that refreshes the encoder for every input word, since our chunk-based translation refreshes the encoder for every input *chunk*, which usually consists of multiple words.

3.3 Chunk Segmentation

We use constituent labels for our rule-based chunk segmentation as follows.

3.3.1 Incremental Constituent Label Prediction

We predict the label of a syntactic constituent coming after a sentence prefix at the current time-step. We call this process *Incremental Constituent Label Prediction* (ICLP). Here, we define this *next*

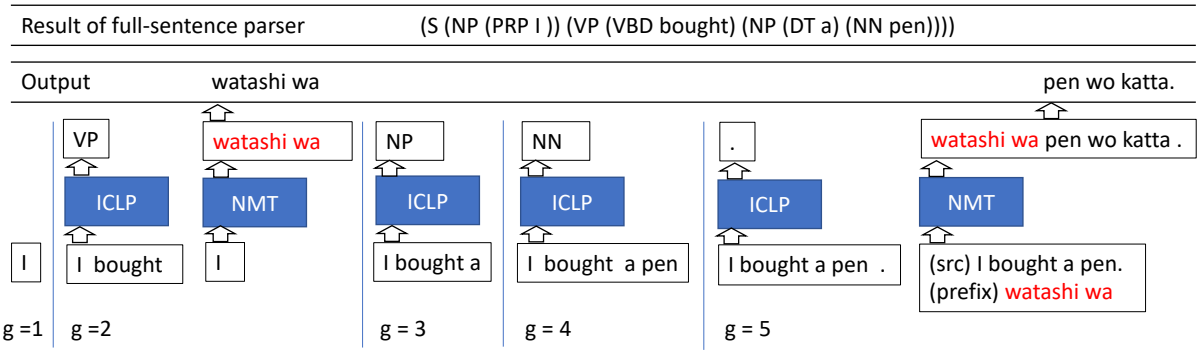


Figure 1: One look-ahead ICLP gives constituent labels. When a boundary is detected based on the label and rules, NMT starts to translate the source subsequence. The previous translation, which is red in the figure, is used as prefix words for the next translation. EOS is omitted for simplicity in the figure.

segmentation	You	/ can	save	time	by	/ doing	this	.
constituent label		VP	VP	NP	PP	S	NP	.
syntax tree	(S (NP (PRP You)) (VP (MD can) (VP (VB save) (NP (NN time)) (PP (IN by) (S (VP (VBG doing) (NP (DT this))))))) (. .))							

Table 3: Example result of one look-ahead ICLP with a minimum segment size of one.

constituent as the one coming next to the sentence prefix in pre-order tree traversal. However, this label prediction is not easy without observations on the next constituent. In this work, we allow one look-ahead, where we read one more word and predict the label of the constituent starting from that word. This causes an additional delay by one word but improves the prediction accuracy. Suppose we have an input sequence $W = [w_1, w_2, ..., w_{|W|}]$. The one look-ahead ICLP predicts the constituent label c_i upon the observation of w_i, as follows:

$$c_i = \underset{c' \in C}{\operatorname{argmax}}\, p(c' | w_{\leq i}), \quad (5)$$

where C is a set of constituent labels. Only a prefix word subsequence is fed into the ICLP, so previous label predictions do not affect later ones.

We can train the ICLP model as a multi-class classifier using a set of training instances in the form of prefix-label pairs. One sentence generates several instances for training data: (w_1, c_1), (w_1, w_2, c_2), (w_1, w_2, w_3, c_3), $(w_1, w_2, w_3, w_4, c_4)$, and so on. We implemented the ICLP model in two different ways using LSTM (Hochreiter and Schmidhuber, 1997) and BERT (Devlin et al., 2019).

3.3.2 Segmentation Rules

Table 3 shows an example of a result by the one look-ahead ICLP. We use one basic and two supplemental rules for chunk segmentation as follows.

- Segment the input coming just before constituents labeled `S` and `VP`.
- If the previous label is `S` or `VP`, do not segment the input.
- If the chunk is shorter than the minimum length, do not segment the input.

In incremental translation from *Subject-Verb-Object* to *Subject-Object-Verb*, the subject can be translated before observing the verb coming next, but the verb should be translated after observing the object. Therefore, the chunk boundary should be between the subject and verb, not between verb and object. To achieve this, we employ a simple rule to segment a chunk just before `VP`. We also include `S` in the rule just as with `VP` because `S` (simple declarative clause) often appears in the form of a unary branch "(`S` (`VP` ...))" as shown in Table 3.

However, in cases such as "can save" in the example, `VP` occurs again immediately after the segmentation before "can." The basic rule suggests segmentation before "save," but it does not seem appropriate. Therefore, we introduce the minimum segment size to avoid such over-segmentation as a hyperparameter to control the accuracy-latency trade-off. If the hyperparameter is larger than one, the chunk segmentation after "You" in the example does not occur.

4 Experimental setup

4.1 Dataset and preprocessing

We conducted experiments on English-Japanese (En-Ja) translation. We also tried English-German (En-De) translation to investigate the difference in language pairs.

For En-Ja, the model was trained on 17.9 M sentence pairs from WMT2020 and fine-tuned on 223 K sentence pairs from IWSLT2017. We used 5312 sentence pairs for the development set from dev2010, tst2011, tst2012, and tst2013 of IWSLT. We evaluated the model on 1442 sentence pairs from dev2021 of IWSLT.

For En-De, the model was trained on 4.5 M sentence pairs from WMT2014 and fine-tuned on 206 K sentence pairs from IWSLT2017. We used 5589 sentence pairs for the development set from dev2010, tst2011, tst2012, and tst2013 of IWSLT. We evaluated the model on 1,080 sentence pairs from tst2015 of IWSLT.

We tokenized English and German sentences with `tokenizer.perl` in Moses (Koehn et al., 2007) and Japanese sentences with MeCab (Kudo, 2005). For each language pair, we used subwords based on Byte Pair Encoding (BPE) (Sennrich et al., 2016) with a shared vocabulary of 16 K entries. To develop the subword vocabulary, we used all of the in-domain training sentences (IWSLT) and one million out-of-domain sentences (WMT).

We trained the ICLP models using Penn Treebank 3 (Marcus et al., 1993) for training, excluding a randomly selected one percent of sentences reserved for the development set. We used NAIST-NTT TED Talk Treebank (Neubig et al., 2014) for the evaluation set. The number of training, development, and test instances (e.g., the number of labels to be predicted) were 2.8 M, 27.9 K, and 21.9 K, respectively. Note that multiple ICLP instances are induced from what a single parse tree generates.

4.2 Model settings

We compared the following four models. All of them were based on the Transformer-base (Vaswani et al., 2017).

wait-k
: The range of `k` is [2, 4, 6,..., 30].

Meaningful Unit
: The hyperparameter is `p`, which is the threshold of the probability of a boundary. The ranges of `p` are [0.5, 0.1, 0.15,..., 0.95], [0.99, 0.991, 0.992,..., 0.999], and [0.9991, 0.9992,..., 0.9999]. Monotonic translation of Meaningful Unit was generated from the fine-tuning dataset by the fine-tuned NMT model. We used their refined Meaningful Unit method, which improved the translation quality at low latency (Zhang et al., 2020)[1]. They used a two look-ahead boundary predictor in their experiments. We additionally tried a one look-ahead predictor because it is not certain how many future words should be used for the predictor.

Fixed-size segmentation
: This simply segments an input with a fixed length specified by a hyperparameter `f`, which means the boundary comes every `f` subwords or words. The range of `f` is [2, 4, 6,..., 30] for words and [4, 8, 12,..., 60] for subwords.

ICLP
: The hyperparameter is `m`, which means the minimum number of words in one segment. The range of `m` is [1, 2, 3, . . . , 29].

We controlled hyperparameters to adapt to a wide range of latency. The hyperparameter is given both in the training and at the inference time for wait-k, but it is given only at the inference time for other models. Therefore, we trained the wait-k model for each `k` while in other approaches a single NMT model is commonly used.

We used fairseq (Ott et al., 2019) to implement these models and basically followed the official baseline for IWSLT 2021[2,3] to set the hyperparameters. We saved checkpoints every 5000 updates for pre-training and every 200 updates for fine-tuning. Other hyperparameters were the same for pre-training and fine-tuning. We stopped training early with patience 4. The max-tokens for the mini batch size was 4096, and weights were updated

[1] Zhang et al. (2020) removed the monotonic translations with a lower score than full-sentence translation. However, it is rare for a monotonic translation to have a higher score than full-sentence translation. Consequently, few sentences remained in our setting. Therefore, we improved the translation quality by preventing over-translation instead of removing it. Once the same words are output four times continuously or the target length becomes four times longer than the source length, we expand the source prefix.

[2] https://github.com/pytorch/fairseq/blob/master/examples/simultaneous_translation/docs/enja-waitk.md

[3] https://github.com/pytorch/fairseq/issues/346

every 4 mini batches. We set the learning rate to 0.0007 and trained the model on a single GPU. The last three models used the same NMT model. We used beam search within chunks in a standard way and chose 1-best hypotheses at the end of chunk translation. The beam size was four for the chunk-based and full-sentence models. We used greedy decoding for wait-k.

We implemented two types of ICLP models as mentioned earlier. For the LSTM-based ICLP, we used two-layered unidirectional LSTMs to encode an input sentence with a fully connected layer for the constituent label prediction. The numbers of dimensions for embedding and hidden states are 512. We tokenized English sentences using `tokenizer.perl` in Moses and Byte Pair Encoding (Sennrich et al., 2016) with a vocabulary of 16 K entries. For the BERT-based ICLP, we used a BERT-based classifier with an additional fully connected layer over the `[CLS]` token, implemented using Huggingface transformers (Wolf et al., 2020) with a pre-trained model `bert-base-uncased` and the corresponding subword tokenizer. For both models, the input was a subword sequence, so the constituent label prediction was made upon the observation of an *end-of-word* subword. The following training conditions were commonly applied to both models: learning rate of 5e-5, training batch size of 512 instances, checkpoints saved at the end of every epoch, and early stopping with the patience of three epochs.

4.3 Evaluation

We used SimulEval (Ma et al., 2020a) to evaluate the quality and latency of simultaneous translation. BLEU (Papineni et al., 2002) was used to evaluate quality. We used Average Lagging (AL) (Ma et al., 2019) to evaluate the latency. AL is widely used and defined by the following equation:

$$AL_g(X, Y) = \frac{1}{\tau_g(|X|)} \sum_{t=1}^{\tau_g(|X|)} g(t) - \frac{t-1}{\gamma}. \quad (6)$$

$\tau_g(|X|)$ is the decoding step when the source sentence finishes. It counts latency up to the $\tau_g(|X|)$ th target token predicted just after reading the final source token. γ is defined as $|Y|/|X|$. When the source length $|X|$ equals target length $|Y|$, AL of wait-k equals its k. In this experiment, the latency was calculated on character level for En-Ja, and word level for En-De.

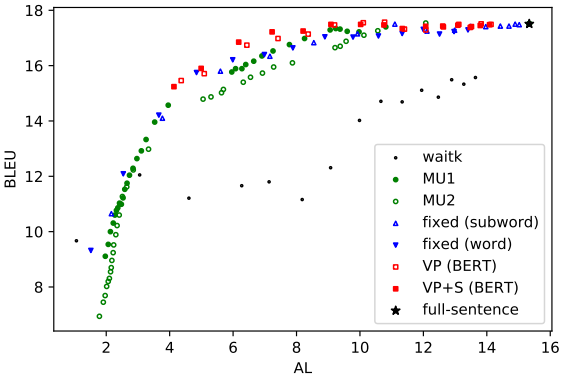


Figure 2: Scatter plot of BLEU and AL (En-Ja). MU1 and MU2 correspond to Meaningful Unit with one and two look-ahead respectively.

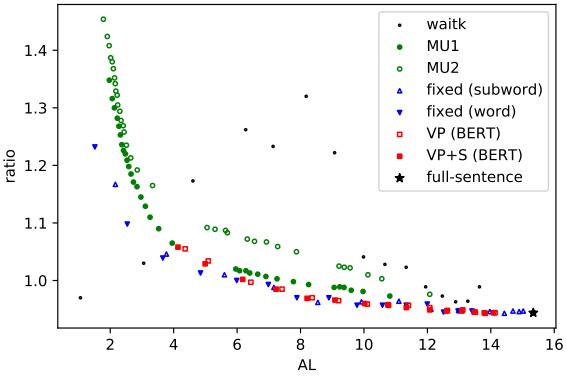


Figure 3: Scatter plot of length ratio and AL (En-Ja)

4.4 Results

We illustrate the results of English-Japanese translation in Figure 2. Our proposed method outperformed baselines in a wide range of AL. Most of the points of the proposed method appear to the upper-left of the other methods, thus showing the best performance. We compared the use of segmentation rules based on `VP` and `VP+S`. The points shifted to the left by adding `S` as boundary because it increased the number of boundaries and decreased latency. Although we tried the different look-ahead lengths of one and two for the boundary predictor of Meaningful Unit, our proposed model outperformed both of these models in a wide range of latency.

The difference between wait-k and the models using the full-sentence translation model was large in the quality-latency trade-off. Surprisingly, the fixed-size segmentation was also effective. When the segment size was fixed, it did not make a large

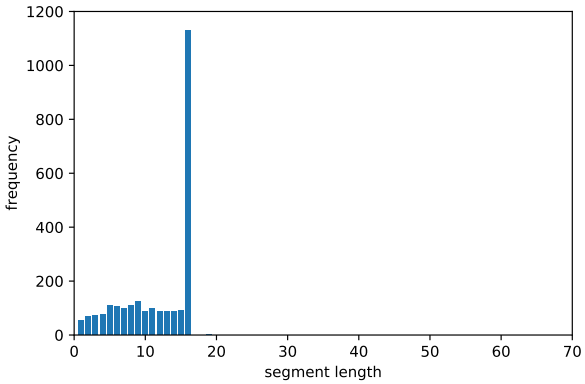


Figure 4: Segment length distribution of fixed-size segmentation with 16 subwords for AL 7.16 (En-Ja test)

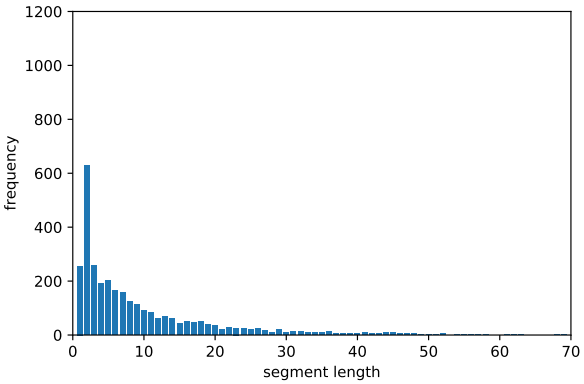


Figure 5: Segment length distribution of one look-ahead Meaningful Unit for AL 7.26 (En-Ja test)

Label	AL	BLEU
Fixed (16 subwords)	7.16	16.34
1 look-ahead MU	7.26	16.53
1 look-ahead ICLP (VP+S)	7.23	17.22

Table 4: BLEU results for AL close to 7

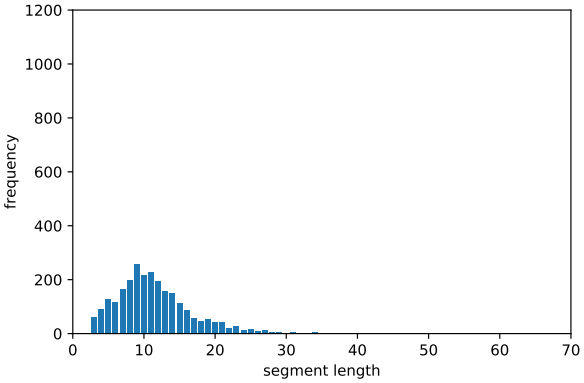


Figure 6: Segment length distribution of one look-ahead ICLP model dividing with label `S` and `VP` for AL 7.23 (En-Ja test)

difference in the result, regardless of whether the unit was a subword or a word.

5 Analysis

5.1 Length ratio

Figure 3 shows the length ratios of translation hypotheses and references with different latency parameters. Too large a ratio decreases the BLEU score and makes the content delivery difficult both in text (subtitles) and speech (text-to-speech).

The length ratio of wait-k was unstable compared to other models because it was trained individually for each k.

Except for wait-k, the length ratios were large in the range of small latency, probably due to the condition mismatch between training and inference. These NMT models were trained on full sentences, but they were used to translate short segments in the inference. Therefore, they tend to output longer segment translations than expected. Their ratios gradually decrease as AL increases and the length of segments becomes closer to the length of full sentences.

5.2 Segment length distribution

Figures 4,5 and 6 show the distributions of source segment length in the En-Ja test set for which AL is close to 7.2. Table 4 shows their corresponding AL and BLEU of each model. The length was calculated as the number of subwords in a segment, and the previous segment was concatenated to the next segment when the previous segment has no translation output.

Segmentation with fixed size 16 has some segments shorter than size 16 because the sentence length is not always a multiple of 16.

Compared with ICLP model, Meaningful Unit has wider distribution, and the most segments consist of two subwords. These short segments have less context information and can output longer segment translation than expected. This would be one of the reason why our proposed method outperformed Meaningful Unit.

5.3 Controlling latency

In Figures 7 and 8, each plot is labeled by the corresponding value of the hyperparameter of inference. It is difficult to control latency for Meaningful Unit as shown in the figure. BLEU scores of hyperpa-

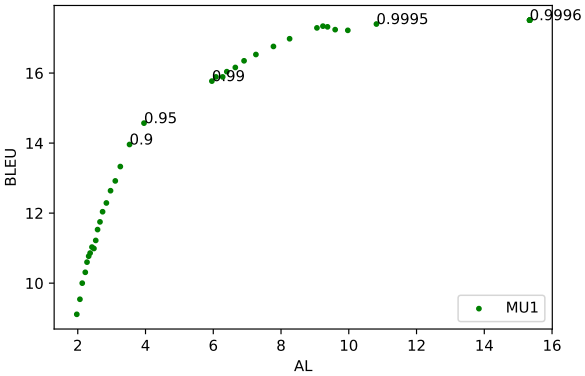


Figure 7: BLEU and AL with different chunk segmentation thresholds for Meaningful Unit

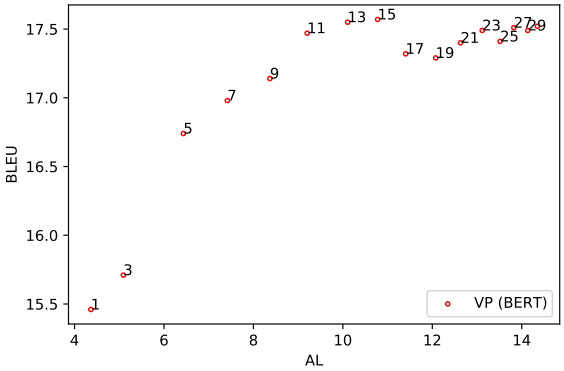


Figure 8: BLEU and AL with different chunk size thresholds for the proposed method

Label	Precision	Recall	F1
NP	0.90	0.94	0.92
VP	0.89	0.97	0.93
NN	0.95	0.97	0.96
,	0.98	1.00	0.99
PP	0.85	0.93	0.89
S	0.87	0.52	0.65

Table 5: Results of label prediction (BERT)

Label	Precision	Recall	F1
NP	0.85	0.89	0.87
VP	0.91	0.94	0.92
NN	0.93	0.92	0.92
,	0.98	1.00	0.99
PP	0.78	0.94	0.86
S	0.84	0.52	0.64

Table 6: Results of label prediction (LSTM)

Label	Precision	Recall	F1
NP	0.62	0.85	0.72
VP	0.75	0.80	0.78
NN	0.60	0.78	0.68
,	0.41	0.34	0.37
PP	0.50	0.47	0.48
S	0.77	0.62	0.69

Table 7: Results of label prediction (BERT) without look-ahead

rameters from 0.9996 to 0.9999 were also the same as that of a full-sentence translation model.

In contrast, our proposed method can easily control latency because it uses the minimum chunk length as an intuitive hyperparameter to adjust it.

5.4 How many words to wait

Compared with the fixed-size segmentation model, our proposed model and Meaningful Unit have a disadvantage in AL, which is caused by the look-ahead approach. Despite this disadvantage, our proposed approach outperformed the fixed-size segmentation in a wide range of AL. This means the benefit of looking at the future words and finding a better boundary outweighed the above disadvantage.

5.5 Performance of ICLP

Tables 5 and 6 show the results in precision and recall of the one look-ahead ICLP models. The LSTM-based ICLP was better in precision, but the BERT-based ICLP was better in recall for `VP`. Figure 9 compares them in the downstream simultaneous translation. The lines connected by dots nearly overlapped, so there was no large difference in BLEU score. LSTM is more efficient than BERT in incremental processes, so it is suitable for practical usage.

Table 7 shows the results by the ICLP model without one look-ahead approach. Compared with Table 5, the scores are much lower. One look-ahead approach was important to improve its performance.

5.6 En-De translation

We conducted additional experiments in En-De translation to investigate the performance in a different language. German is another language with different word order from English especially in verbs and also suffers from the reordering problem. Figure 10 shows the results. This is almost the opposite of the results of the En-Ja translation. The

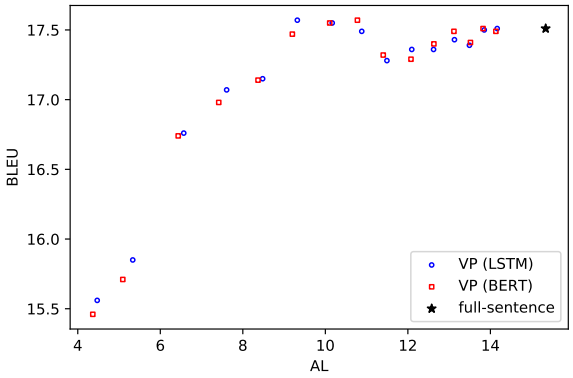


Figure 9: Comparison between the use of LSTM- and BERT-based ICLP

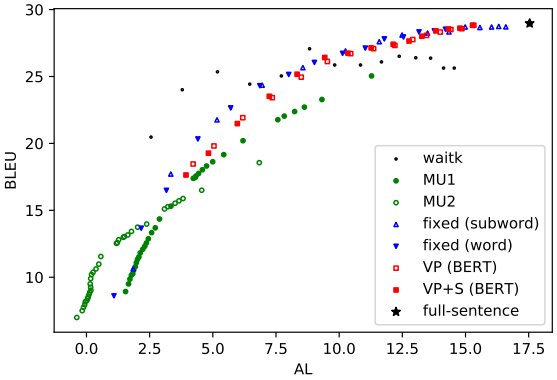


Figure 10: Scatter plot of BLEU and AL (En-De)

proposed boundary decision rules used for En-Ja translation were not so effective for En-De translation, so we need to find other rules to detect boundaries in En-De translation.

6 Conclusion

We proposed a novel segmentation method for simultaneous translation that uses simple rules and ICLP. Our proposed method is simple, and it outperformed the baselines in the quality-latency trade-off in En-Ja translation. On the other hand, the proposed method did not work effectively in En-De translation due to the smaller word order differences than those in En-Ja translation.

In future work, we expect to extract segmentation rules automatically and apply these rules to other language pairs as well.

7 Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP21H03500.

References

Ashkan Alinejad, Maryam Siahbani, and Anoop Sarkar. 2018. Prediction improves simultaneous neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3022–3027, Brussels, Belgium. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation 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 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1352, Doha, Qatar. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the*

Association for Computational Linguistics: Volume 1, Long Papers, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Sepp Hochreiter and J¨urgen Schmidhuber. 1997. *Long short-term memory*. Neural Computation.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head finalization: A simple reordering rule for SOV languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251, Uppsala, Sweden. 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 Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer. http://mecab.sourceforge.net/.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Xutai Ma, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020b. Monotonic multihead attention. In *ICLR 2020*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Graham Neubig, Katsuhito Sudoh, Yusuke Oda, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2014. The NAIST-NTT TED talk treebank. In *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, USA.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic transcription for low-latency speech translation. In *Interspeech 2016*, pages 2513–2517.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 551–556, Baltimore, Maryland. Association for Computational Linguistics.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Syntax-based simultaneous translation through prediction of unseen syntactic constituents. 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 198–207, Beijing, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. 2017. Online and lineartime attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2837–2846.

Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. 2013. Segmentation strategies for streaming speech translation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 230–238, Atlanta, Georgia. Association for Computational Linguistics.

Harsh Satija and Joelle Pineau. 2016. Simultaneous machine translation using deep reinforcement learning. In *Workshops of International Conference on Machine Learning*, page 110–119.

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 Uszko-reit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and andIllia Polosukhin. 2017. Attention is all you need. *CoRR*, page Vol.abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289, Online. Association for Computational Linguistics.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

Contrastive Learning for Context-aware Neural Machine Translation Using Coreference Information

Yongkeun Hwang[1], Hyungu Yun[1], Kyomin Jung[1,2]
[1]Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2]Automation and Systems Research Institute, Seoul National University, Seoul, Korea
{wangcho2k, youaredead, kjung}@snu.ac.kr

Abstract

Context-aware neural machine translation (NMT) incorporates contextual information of surrounding texts, that can improve the translation quality of document-level machine translation. Many existing works on context-aware NMT have focused on developing new model architectures for incorporating additional contexts and have shown some promising results. However, most existing works rely on cross-entropy loss, resulting in limited use of contextual information. In this paper, we propose CorefCL, a novel data augmentation and contrastive learning scheme based on coreference between the source and contextual sentences. By corrupting automatically detected coreference mentions in the contextual sentence, CorefCL can train the model to be sensitive to coreference inconsistency. We experimented with our method on common context-aware NMT models and two document-level translation tasks. In the experiments, our method consistently improved BLEU of compared models on English-German and English-Korean tasks. We also show that our method significantly improves coreference resolution in the English-German contrastive test suite.

1 Introduction

Neural machine translation (NMT) has achieved impressive performances on translation quality, due to the introduction of novel deep neural network (DNN) architectures such as encoder-decoder model (Cho et al., 2014; Sutskever et al., 2014), and self-attentional networks like Transformer (Vaswani et al., 2017). The state-of-the-art NMT systems are now even comparable with human translators in sentence-level performance.

However, there are a number of issues on document-level translation (Läubli et al., 2018). These include pronoun resolution across sentences (Guillou et al., 2018), which needs cross-sentential contexts. To incorporate such document-level contextual information, several methods for context-aware NMT have been recently proposed. Many of the works have focused on introducing new model architectures like multi-encoder models (Voita et al., 2018) for encompassing contextual texts of the source language. These works have shown significant improvement in addressing discourse phenomena such as anaphora resolution mentioned above, as well as moderate improvements in overall translation quality (Lopes et al., 2020).

Despite some promising results, most of the existing works have trained the model by minimizing cross-entropy loss, making the model rather exploit contextual information implicitly such as a form of regularization (Kim et al., 2019; Li et al., 2020). Data augmentation for context-aware NMT is also an important issue, despite that recent works have focused on back-translation (Huo et al., 2020).

In this paper, we propose a Coreference-based Contrastive Learning for context-aware NMT (CorefCL), a novel data augmentation and contrastive learning scheme leveraging coreference information. Cross-sentential coreference between the source and target sentence can be a good source of training signal for context-aware NMT since it occurs when one or more expressions refer to the same entity, thus reflects dependencies between the source and contextual sentences.

CorefCL starts by conducting automatic annotation of coreference between the source and contextual sentences. Then, the referred mentions on contextual sentences are corrupted by removing and/or replacing tokens to generate contrastive examples. With those contrastive examples, we introduce a contrastive learning scheme equipped with a max-margin loss which encourages the model to discriminate between the original examples and the contrastive ones. By doing so, CorefCL makes the model more sensitive to cross-sentential contextual information.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 1135–1144
November 10–11, 2021.

We experimented with CorefCL on three English-German corpora and one English-Korean document-level corpus, including WMT, IWSLT TED talk, and OpenSubtitles'18 English-German subtitles translation task, and a web-crawled English-Korean subtitles translation. In all translation tasks, CorefCL consistently improves overall BLEU over baseline models without CorefCL. On experiments with three common context-aware model settings, we show that improvements by CorefCL are also model-agnostic. Finally, we show that the proposed method significantly improved the performance on ContraPro (Müller et al., 2018), an English-German contrastive coreference benchmark.

2 Related Works

2.1 Context-aware NMT

Context-aware machine translation has been vigorously studied to exploit the crucial context information in surrounding sentences. Recent works have shown that contextual information can help the model to generate not only more consistent but also more accurate translation (Smith, 2017; Voita et al., 2018; Müller et al., 2018; Kim et al., 2019).

In particular, Voita et al. (2018) introduced a context-aware Transformer model which is able to induce anaphora relations, Miculicich et al. (2018) showed that a model using cross-sentential contextual information significantly outperforms in document-level translation tasks, and Yun et al. (2020) insisted that context-aware models record the best performance especially in spoken language translation tasks where mandatory information tend to be sparse over multiple sentences.

The simplest method for context-aware machine translation is to concatenate all surrounding sentences and treat the concatenated sequence as a single sentence (Tiedemann and Scherrer, 2017). Although the concatenation strategy boosted Transformer architectures in multiple tasks (Tiedemann and Scherrer, 2017; Voita et al., 2018; Yun et al., 2020), it lagged behind efficiency as the Transformer architecture has limited long-range dependency (Tang et al., 2018).

To improve the efficiency, an additional encoder module is introduced to encode only the context sentences (Libovický and Helcl, 2017; Jean et al., 2017; Voita et al., 2018). Additionally, hierarchical structures also have been introduced because the context sentences do not have the same significance as the input sentences (Miculicich et al., 2018; Yun et al., 2020).

2.2 Coreference and NMT

The difference in coreference expressions among languages (Zinsmeister et al., 2017; Lapshinova-Koltunski et al., 2020) gives MT systems a challenge on pronoun translation (Bawden et al., 2018). Several recent works have attempted to incorporate coreference information (Ohtani et al., 2019). The closest work to ours is (Stojanovski and Fraser, 2018) which also adds noise on creating a coreference-augmented dataset, while we do not add oracle coreference information directly to the training data.

2.3 Data augmentation for NMT

One of the most common methods for data augmentation in NMT is back-translation that generates pseudo-parallel data from monolingual corpora using intermediate NMT models (Sennrich et al., 2016a). Generally, back-translation is conducted at sentence-level, however, several works have proposed document-level back-translation (Sugiyama and Yoshinaga, 2019; Huo et al., 2020).

On the other hand, sentence corruption by removing or replacing word(s) has also been widely used for improving model performance and robustness (Lample et al., 2018; Voita et al., 2019). Inspired by these works, we choose sentence corruption for contrastive learning.

2.4 Contrastive Learning

Contrastive learning is to learn a representation by contrasting positive and negative (contrastive) examples. It has succeed in various machine learning fields including computer vision (Chen et al., 2020) and natural language processing (Mikolov et al., 2013; Wu et al., 2020; Lee et al., 2021).

Recently, several approaches to contrastive learning for NMT have also been studied. Yang et al. (2019) proposed strategies for generating word-omitted contrastive examples and leveraging contrastive learning for reducing word omission errors in NMT. Pan et al. (2021) applied contrastive learning for multilingual MT and employed data augmentation for obtaining both the positive and negative training examples.

While these works have been conducted in sentence-level NMT settings, we focus on extending contrastive learning in context-aware NMT.

3 Context-aware NMT models

In this section, we briefly overview context-aware NMT methods and describe our baseline models which are also commonly adopted in recent works.

Generally, a sentence-level (context-agnostic) NMT model takes an input sentence in a source language and returns an output sentence in a target language. On the other hand, a context-aware NMT model is designed to handle surrounding contextual sentences of source and/or target sentences. We focus on leveraging the contextual sentences of the source language.

Throughout this work, we consider the Transformer (Vaswani et al., 2017) as a base model architecture by following the majority of the recent works on context-aware NMT. Transformer consists of a stack of self-attentional layers in which a self-attention module is followed by a feed-forward module for each layer. Here we list four Transformer-based configurations that we used in the experiments:

- **sent-level**: As a baseline, we have experimented with the basic Transformer model which does not use any contextual sentences.
- **concat**: This is a straightforward approach to incorporate contextual sentences without modifying the Transformer model (Tiedemann and Scherrer, 2017). This concatenates all contextual sentences and an input sentence with special tokens between sentences.
- **multi-enc**: This has an extra encoder for encoding contextual sentences separately. We follow the model introduced in (Voita et al., 2018) which obtain a hidden representation of contextual sentences by weight-shared Transformer encoder. The model combines the encoded source and context representations using a source-to-context attention mechanism and a gated summation.
- **multi-enc-hier**: To represent multiple contextual sentences effectively, hierarchical encoders for contextual sentences have been proposed (Miculicich et al., 2018; Yun et al., 2020). In this configuration, the context representation is calculated in token-level first, then finally processed in sentence-level. We experimented with the model of (Yun et al., 2020) in this paper.

All the model structures are described in Figure 1.

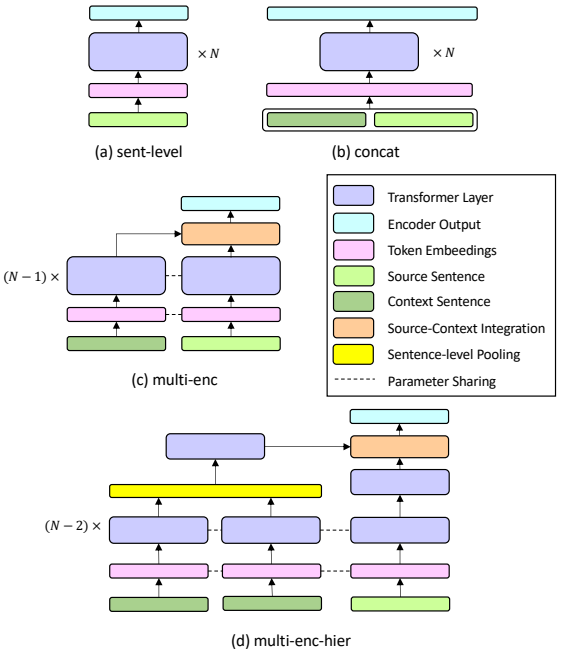


Figure 1: The structure of compared context-aware NMT models.

4 Our Method: CorefCL

In this section, we explain the main idea of CorefCL, a data augmentation and contrastive learning scheme leveraging coreference between the source and contextual sentences.

4.1 Data Augmentation Using Coreference

Generally, constrastive learning encourages a model to discriminate ground-truth and contrastive (negative) examples. In existing works, a number of approaches have been studied for obtaining contrastive examples:

- Corrupting the sentence by randomly removing or replacing one or more tokens in the sentence. (Yang et al., 2019)
- Choosing an irrelevant example in the batch or dataset. (Pan et al., 2021)
- Perturbations on representation space. Usually output vector of encoder or decoder is used. (Lee et al., 2021)

CorefCL basically takes a similar approach to the first one, by the sentence corruption. However, unlike previous works that modify the source sentence, CorefCL modifies the contextual sentences to form contrastive examples. Specifically, we corrupt cross-sentential coreference mentions which

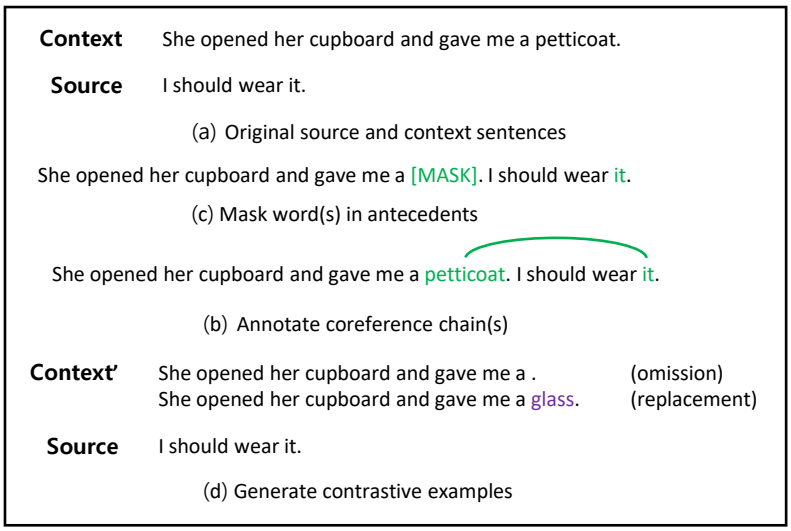


Figure 2: Data augmentation process of CorefCL.

occur between the source and its contextual sentences. This is based on the intuition that coreference is one of the core components of coherent translation.

More formally, steps to forming contrastive examples in CorefCL are as follows (see also Figure 2):

1. Annotate the source documents automatically. We use NeuralCoref[1] to identify the coreference mentions between the source and its previous sentences as contextual sentences

2. Filter the examples with cross-sentential coreference chain(s) between the source and contextual sentences. Around 20 to 30% of the training corpus is annotated in this way. See Section 5.1 for details

3. For each coreference chain, mask every word in the antecedents with a special token. We also keep the original examples for training

4. Masked words are replaced randomly with other words in vocabulary (*word replacement*), or omitted (*word omission*)

In the experiments, we take both of the corruption strategies. Precisely, the masked words are removed with a probability of 0.5, or randomly replaced otherwise. We found that this method is more effective compared to the methods using only one of the two corruption strategies. Please refer to the ablation study in Section 5.5 for more details.

[1]https://github.com/huggingface/neuralcoref

4.2 Contrastive Learning for Context-aware NMT

Context-aware NMT models can implicitly capture dependencies between the source and contextual sentences. CorefCL introduces a max-margin contrastive learning loss to train the model to explicitly discriminate inconsistent contexts. This contrastive loss also encourages a model to be more sensitive to the contents of contextual sentences.

Formally, given the source $\mathbf{x}$, target $\mathbf{y}$, n contextual sentences $C = [\mathbf{c}_1, \cdots, \mathbf{c}_n]$ in the data $\mathcal{D}$, we first train the model by minimizing a negative log-likelihood loss, which is a common MT loss:

$$\mathcal{L}_{MT} = \sum_{(\mathbf{x},\mathbf{y},C)\in\mathcal{D}} -\log P(\mathbf{y}|\mathbf{x}, C).$$

Once the model is trained with MT loss, we fine-tune the model with a contrastive loss. With a contrastive version of context $\tilde{C}$, our contrastive learning objective is minimizing a max-margin loss (Huang et al., 2018; Yang et al., 2019):

$$\mathcal{L}_{CL} = \sum_{(\mathbf{x},\mathbf{y},C,\tilde{C})\in\mathcal{D}} \max\{\eta + \log P(\mathbf{y}|\mathbf{x}, \tilde{C}) - \log P(\mathbf{y}|\mathbf{x}, C), 0\}.$$

Minimizing $\mathcal{L}_{CL}$ encourages the log-likelihood of the ground-truth to be at least η larger than that of the contrastive examples. In our formulation, we want the model to be more sensitive to the subtle changes in the contextual sentences.

The contrastive loss is jointly optimized with MT loss since we empirically found that the joint

optimization has yielded better performance than minimizing CL loss only as similar to (Yu et al., 2020):

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{MT} + \alpha\mathcal{L}_{CL},$$

where $\alpha \in [0, 1]$ is a weight for balancing between contrastive learning and MT loss. For simplicity, we fixed α during fine-tuning.

5 Experiments

5.1 Datasets

We experimented with CorefCL on various document-level parallel datasets: i) 3 English-German datasets including WMT document-level news translation[2] (Barrault et al., 2019), IWSLT TED talk [3] (Cettolo et al., 2017), OpenSubtitles'18[4] (Lison et al., 2018), and ii) our web-crawled English-Korean subtitles corpus.

For all tasks, we take every 2 preceding sentences as contextual sentences and we only consider sentences within the same document (article, talk, movie, one episode of TV programs, etc.) of the source sentence. If split of the validation and the test set is not presented in the data, we apply document-based split to ensure that training and validation/test data is well-separated. Details of datasets are listed as follows:

WMT We use a set of parallel corpora annotated with document boundaries which is released in WMT'19 news translation task. Specifically, we combine Europarl v9, News Commentary v14, and MODEL-RAPID to form a training set containing $3.7M$ examples and $0.85M$ with cross-sentential coreferences. For validation and test sets, we used newstest2013 and newstest2019 which contain $3.05k$ and $2.14k$ examples respectively.

IWSLT The IWSLT dataset consists of transcriptions of TED talks in a variety of languages. We used the 2017 version of the training set, a combination of dev2010, tst2010, tst2015 as a validation set, and tst2017 as a test set. The resulting dataset consists of $232k$ ($50.3k$ with cross-sentential coreferences), $3.5k$, $1.2k$ examples of train, dev, test sets respectively.

OpenSubtitles We also choose the English-German pair of OpenSubtitles2018 corpora. The raw corpus contains $24.4M$ parallel sentences. We follow the filtering methods in (Voita et al., 2019) by removing pairs that have a time overlap of subtitle frames less than 0.9. We also use separate documents for validation / test sets, resulting in $3.9M$ ($1.01M$ with cross-sentential coreferences), $40.7k$, $40.5k$ examples for train / validation / test sets respectively.

En-Ko Subtitles For English-Korean experiments, we first crawled approximately $6.1k$ bilingual subtitle files from websites such as GomLab.com. Since sentence pairs of these subtitles are already soft-aligned by the creators so we applied a simple time-code based heuristics to filter examples. The final data contains $1.6M$ ($0.24M$ with cross-sentential coreferences), $155.6k$, and $18.1k$ examples of consecutive sentences in the training, validation, and test sets respectively.

For preprocessing, all English and German corpus is tokenized first with Moses (Koehn et al., 2007) tokenizer[5]. We then apply the BPE (Sennrich et al., 2016b) using SentencePiece[6], and the size of the merge operation is approximately $16.5k$. We also put a special token [BOC] at the beginning of contextual sentences to differentiate them from the source sentences.

5.2 Settings

We use model hyperparameters, such as the size of hidden dimensions and the number of hidden layers as same the `transformer-base` (Vaswani et al., 2017), since all of the compared models are based on Transformer. Specifically, we set 512 as the hidden dimension, the number of layers is 6, the number of attention heads is 8, and the dropout rate is set to 0.1.

All models are trained with ADAM (Kingma and Ba, 2014) with different learning rates for each dataset. We employ early stopping of the training when the MT loss on the validation set does not improve. We start training each baseline model from scratch with random initialization and document-level dataset. Note that all the baseline models are not trained using iterative training as (Zhang et al., 2018; Huo et al., 2020) which first trains the model from sentence-level task first, then document-level task. All the evaluated models are implemented on top of the transformers[7] framework.

We measure the translation quality by the BLEU score (Papineni et al., 2002). For scoring BLEU,

[2] http://www.statmt.org/wmt19/translation-task.html
[3] https://wit3.fbk.eu/home
[4] https://opus.nlpl.eu/OpenSubtitles-v2018.php
[5] https://github.com/moses-smt/mosesdecoder
[6] https://github.com/google/sentencepiece
[7] https://github.com/huggingface/transformers

System	WMT	OpenSubtitles	IWSLT	En-Ko Subtitles	
				detok.	char.
sent-level	22.7	27.6	29.3	8.6	19.2
concat	22.4	28.3	29.7	9.3	22.1
+ CorefCL	23.5 (+1.1)	29.1 (+0.8)	30.9 (+1.3)	10.9 (+1.6)	24.9 (+2.8)
multi-enc	23.1	28.6	29.8	9.2	21.7
+ CorefCL	24.3 (+1.2)	29.8 (+1.4)	31.1 (+1.3)	10.8 (+1.6)	24.4 (+2.7)
multi-enc-hier	24.4	29.1	30.0	10.3	23.1
+ CorefCL	25.4 (+1.0)	30.2 (+1.1)	31.1 (+1.2)	11.7 (+1.4)	25.7 (+2.6)

Table 1: Corpus-level BLEU scores of compared models on different tasks. For the En-Ko subtitles task, we list both detokenized (detok.) and character-level (char.) scores. Improvements by CorefCL are denoted in (). Underlined score means that the model has the largest BLEU improvements among models in the same task.

we use the sacreBLEU (Post, 2018) case-sensitive, detokenized scores for En-De, and case-insensitive scores with `intl` tokenizer for En-Ko task. We also report case-insensitive char-level scores on En-Ko for comparison.

5.3 Overall BLEU Evaluation

We display the corpus-level test BLEU scores of all compared models on different tasks in Table 1. Among the baseline systems, all context-aware models show moderate improvements over the sentence-level (sent-level) baseline. These results are comparable to that of Huo et al. (2020) on the IWSLT task except for multi-enc-hier, and Yun et al. (2020) on OpenSubtitles task. One exception is a single-encoder model (concat) on WMT task, which seems due to the longer average sentence length.

We evaluated CorefCL by fine-tuning the context-aware models. Results show that models with CorefCL outperformed their vanilla counterparts, with the BLEU gain of up to 1.4 in En-De tasks, and 1.6/2.8 (detokenized/char-level BLEU) in the En-Ko subtitles task.

We observed that while CorefCL consistently improves BLEU on all tasks, it achieves better results on IWSLT and En-Ko subtitles tasks. Since improvements on much larger datasets like WMT and OpenSubtitles are smaller, we suggest that CorefCL also works as a regularization.

5.4 Results on English-German Contrastive Evaluation Set

To assess how CorefCL improves the ability to deal with pronoun-related translations more in detail, we experiment our method with ContraPro.[8] ContraPro is a contrastive test suit for En-De pronoun translation introduced by Müller et al. (2018). The evaluation is done by letting the model scores the German sentence with correct and incorrect pronoun translation, given the source and contextual English sentence. The accuracy is calculated by counting the number of correctly scored examples (i.e. correct examples that received a higher score than their incorrect counterpart).

System	Trained on			
	WMT		OpenSubtitles	
	BLEU	Acc.	BLEU	Acc.
sent-level	19.3	47.9	29.6	48.4
concat	19.9	49.7	30.5	54.4
+ CorefCL	20.3	51.2	32.3	57.9
multi-enc-hier	20.4	50.9	31.7	57.3
+ CorefCL	21.9	52.4	33.6	60.5

Table 2: BLEU and pronoun resolution accuracies on ContraPro (Müller et al., 2018) En-De contrastive test set.

We evaluate the models trained with WMT and OpenSubtitles tasks. We also list BLEU scores of En-De translation using the English source text in ContraPro. As shown in Table 2, CorefCL significantly improves the baselines in scoring accuracy for all models by up to 5.5%, as well as slight improvements in BLEU scores.

One interesting finding is that CorefCL also achieved substantial accuracy gain on the models trained on WMT. Since the ContraPro is created from OpenSubtitles, WMT-trained models would yield lower performance because of domain shift between training and testing. Table2 clearly shows the performance drop in BLEU, nevertheless, moderate improvements in accuracy can also be ob-

[8] https://github.com/ZurichNLP/ContraPro

served on WMT-trained models.

5.5 Analysis

System	BLEU	Accuracy
multi-enc-hier	31.7	57.3
+ CorefCL	33.6	60.5
- Word omission	32.4	59.4
- Word replacement	32.3	58.6

Table 3: Ablation study on coreference corruption strategy. All systems are trained on OpenSubtitles English-German dataset and evaluated on ContraPro.

Ablation Study CorefCL uses the two corruption strategies for generating contrastive coreference mentions; word omission and word replacement. To make a better understanding of influence of these strategies, we evaluate CorefCL of different settings of these strategies.

As shown in Table.3, using both types of corruptions results in better performance. Removing one of the two strategies slightly degrades both the pronoun resolution accuracy and BLEU. Although not being significant, removing the word replacement has more impact on accuracy. This suggests that a standard context-aware model, at least for multi-enc-hier is less sensitive to word substitution. The word replacement strategy can complement this behavior as resulted in better performance.

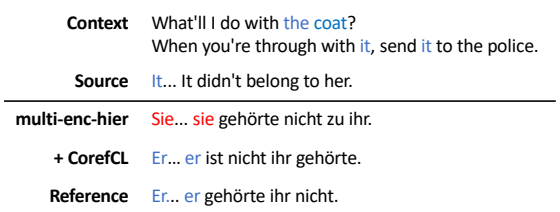


Figure 3: Example translation with and without CorefCL.

Qualitative Example We display a sample from ContraPro corpus and its translations made by multi-enc-hier model trained with OpenSubtitle task. In this example, since "coat" is translated as *Mantel* which is a masculine noun thus *Er* would be adequate translation of "It" instead of *Sie* which is feminine. While multi-enc-hier incorrectly translated "It" as *Sie*, the model fine-tuned with CorefCL correctly resolved it as *Er*.

In practice, context-aware models that do not leverage target-side contexts struggle to maintain these kinds of coreference consistency (Müller et al., 2018; Lapshinova-Koltunski et al., 2019) because of the asymmetric nature of grammatical components and data distributions. Results show that CorefCL can complement the limitation of source-only context-aware models.

6 Conclusions and Future Work

We have presented a data augmentation and contrastive learning scheme based on coreference for context-aware NMT. By leveraging coreference mentions between the source and target sentence, CorefCL effectively generates contrastive examples for applying contrastive learning on context-aware NMT models. In the experiments, CorefCL consistently improves the translation quality and pronoun resolution accuracy.

As future work, we plan to extend CorefCL to target contexts since maintaining coreference consistency needs both the source and the target contexts. It would be also interesting that applying CorefCL for fine-tuning pre-trained big language models like BART (Lewis et al., 2020) or T5 (Raffel et al., 2020) for downstream document-level MT tasks.

Acknowledgements

We thank Minwoo Lee for helpful discussions, as well as the anonymous reviewers for their thoughtful and constructive comments. This works is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2021R1A2C2008855)

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh,

Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, pages 1–14.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A pronoun test suite evaluation of the English–German MT systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 570–577, Belgium, Brussels. Association for Computational Linguistics.

Jiaji Huang, Yi Li, Wei Ping, and Liang Huang. 2018. Large margin neural language model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191, Brussels, Belgium. Association for Computational Linguistics.

Jingjing Huo, Christian Herold, Yingbo Gao, Leonard Dahlmann, Shahram Khadivi, and Hermann Ney. 2020. Diving deep into context-aware neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 604–616, Online. Association for Computational Linguistics.

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

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. 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*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Ekaterina Lapshinova-Koltunski, Cristina España-Bonet, and Josef van Genabith. 2019. Analysing coreference in transformer outputs. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 1–12, Hong Kong, China. Association for Computational Linguistics.

Ekaterina Lapshinova-Koltunski, Marie-Pauline Krielke, and Christian Hardmeier. 2020. Coreference strategies in English-German translation. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 139–153, Barcelona, Spain (online). Association for Computational Linguistics.

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*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive Learning with Adversarial Perturbations for Conditional Text Generation. In *ICLR*, pages 1–25.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518, Online. Association for Computational Linguistics.

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.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal. European Association for Machine Translation.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. 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*, volume 26. Curran Associates, Inc.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Takumi Ohtani, Hidetaka Kamigaito, Masaaki Nagata, and Manabu Okumura. 2019. Context-aware neural machine translation with coreference information. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 45–50, Hong Kong, China. Association for Computational Linguistics.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Karin Sim Smith. 2017. On integrating discourse in machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 110–121.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60, Brussels, Belgium. Association for Computational Linguistics.

Amane Sugiyama and Naoki Yoshinaga. 2019. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China. 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*, December, pages 3104–3112, Montréal, Canada.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium. 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.

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*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

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.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: Contrastive Learning for Sentence Representation. *arXiv preprint*, arXiv:2012.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.

Lei Yu, Laurent Sartran, Po-Sen Huang, Wojciech Stokowiec, Domenic Donato, Srivatsan Srinivasan, Alek Andreev, Wang Ling, Sona Mokra, Agustin Dal Lago, Yotam Doron, Susannah Young, Phil Blunsom, and Chris Dyer. 2020. The DeepMind Chinese–English Document Translation System at WMT2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 326–337, Online. Association for Computational Linguistics.

Hyeongu Yun, Yongkeun Hwang, and Kyomin Jung. 2020. Improving context-aware neural machine translation using self-attentive sentence embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9498–9506.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.

Heike Zinsmeister, Stefanie Dipper, and Melanie Seiss. 2017. Abstract pronominal anaphors and label nouns in German and English: Selected case studies and quantitative investigations.

Author Index