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. ©2021 Association for Computational Linguistics

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.