The University of Edinburgh's English-Tamil and English-Inuktitut Submissions to the WMT20 News Translation Task

Rachel Bawden Alexandra Birch Radina Dobreva
Arturo Oncevay Antonio Valerio Miceli Barone Philip Williams

School of Informatics, University of Edinburgh, Scotland
{rbawden,abirch,rdobreva,a.oncevay,amiceli,pwillia4}@ed.ac.uk

Abstract

We describe the University of Edinburgh's submissions to the WMT20 news translation shared task for the low resource language pair English-Tamil and the mid-resource language pair English-Inuktitut. We use the neural machine translation transformer architecture for all submissions and explore a variety of techniques to improve translation quality to compensate for the lack of parallel training data. For the very low-resource English-Tamil, this involves exploring pretraining, using both language model objectives and translation using an unrelated high-resource language pair (German-English), and iterative backtranslation. For English-Inuktitut, we explore the use of multilingual systems, which, despite not being part of the primary submission, would have achieved the best results on the test set.

1 Introduction

The University of Edinburgh participated in the WMT20 news translation shared task for English-Tamil and English-Inuktitut in both translation directions.[1,2] Neither language pair benefits from large quantities of parallel data, so we approach training using different techniques to compensate for this lack of data: pretraining and iterative backtranslation for English-Tamil and multilingual systems for English-Inuktitut. We use neural machine translation (MT) and specifically the transformer architecture (Vaswani et al., 2017): the base variant for the lower-resourced English-Tamil and the big variant for the mid-resource English-Inuktit. In both cases, significant improvements are seen when compared to the in-house baselines tested, particularly notable for pretraining for English-Tamil.

Awaiting the results of the official human evaluation, we report the automatic evaluation scores using BLEU (Papineni et al., 2002) as implemented in sacreBLEU (Post, 2018). A summary of these results on the dev and test sets can be found in Table 1 for all UEDIN submissions. The details of our submissions can be found in Section 2 for English-Tamil and in Section 3 for English-Inuktitut.

Language direction	Dev	Test
EN→TA	12.30	8.40
TA→EN	21.00	16.60
EN→IN	27.0	8.2
IN→EN	48.8	23.0

Table 1: Summary of results for all UEDIN submissions according to the automatic evaluation (BLEU).

2 English↔Tamil

As for our English↔Gujarati systems last year at WMT19 (Bawden et al., 2019), we use pretraining and data augmentation to tackle the low-resource language pair English–Tamil. Our experiments show that pre-training, training on backtranslated data and then fine-tuning is useful in both directions, although we introduce slight variations in the training and fine-tuning approaches used for each language direction.

2.1 Data and pre-processing

Our models are trained in the constrained scenario, using publicly available WMT20 data. We choose to exclude the terminology-like Wikititles as well as WikiMatrix[3] from our training data, using only

[1]The UEDIN participation for English-German is in a separate submission.

[2]Code and models can be found at http://data.statmt.org/wmt20_systems/.

[3]While term lists contain useful vocabulary, they can inundate the training data due to their large size. This can cause translation problems due to the different nature of the text, notably in terms of sentence length. The EN-TA portion of WikiMatrix corpus is very noisy adn so this is excluded too.

Data type	#sentences	Corpora
Parallel en–ta	340,995	PMindia, Tanzil, NLCP, PIB, MKB, EnTam
Monolingual en (in-domain)	653,606,835	News (crawl, discussions, commentary)
Monolingual en (out-of-domain)	101,692,093	Europarl, Wiki dumps
Monolingual ta (in-domain)	668,008	News crawl
Monolingual ta (out-of-domain)	1,553,160	Wiki dumps
Parallel de–en	43,675,462	Europarl, News commentary, Paracrawl, WikiMatrix, Tilde Rapid

Table 2: Data used for the Tamil-English models. Note that we also use German-English data for some of our experiments as a form of pretraining.

the corpora shown in Table 2. We use both parallel data and monolingual data for English-Tamil and also exploit parallel data available for English-German as a form of pre-training.

All data was first cleaned, keeping sentences of 3–100 (untokenised) units, for which the length ratio between the parallel sentences is maximum 2.2, and do not contain more than 50% non-alphabetic characters or more than 50% of words without an alphabetic character.[4] We deduplicate the data and normalise punctuation using Moses (Koehn et al., 2007). We then apply subword segmentation using SentencePiece (Kudo and Richardson, 2018) and the BPE strategy (Sennrich et al., 2016).[5]

2.2 Approach used

We adopt a three-step approach to training our models, consisting of: (i) *pre-training* model parameters using either an mBART language model or a translation model for the highly resourced De-En language pair, (ii) *iterative backtranslation* to produce synthetic parallel data of increasing quality, and (iii) *final model creation* consisting of fine-tuning pretrained models using both genuine parallel and backtranslated data. We provide the details of these three steps below.

Pre-training We experimented with several pre-training objectives: language modelling using XLM (Lample and Conneau, 2019a) or mBART (Liu et al., 2020), and MT pre-training using a higher-resourced language pair (namely English-German). Using a higher-resourced language pair for pretraining, even if this pair is unrelated to the language pair on which the model is fine-tuned, has shown to be an effective and simple way of boosting performance (Kocmi and Bojar, 2018; Aji et al., 2020). For the De-En models, we had to choose between (i) initialising only model parameters and (ii) preserving all model and training parameters from the parent model (similar to Grundkiewicz et al. (2019)). We chose the first option as it produced better results in our experiments.

For mBART pretraining, we use all Tamil and English monolingual data without shuffling or deduplication. We tag the input segments with a language tag and a domain tag: either in-domain (news) or out-of-domain as in (Caswell et al., 2019). For XLM pretraining we use the deduplicated and shuffled corpus (since cross-sentence context is not needed) and we subsample the English because of computing cost. We also use domain tags, with language information provided in the form of language embeddings as per the standard implementation. For De-En pre-training, we use all De-En parallel data described in Table 2, with a joint English-Tamil-German vocabulary. We experiment with pretraining models in the two directions (De→En and En→De) and find that the De→En model produces better results when fine-tuned on TA-EN data.

System	EN→TA		TA→EN	
	dev	test	dev	test
Parallel-only baseline	5.10	3.10	10.10	10.60
XLM	7.44	5.00	13.44	10.90
mBART	7.40	4.65	14.00	13.40
De-En	7.30	5.00	13.60	14.20

Table 3: Comparison of pre-training methods for EN↔TA (BLEU) after fine-tuning on parallel data.

Table 3 shows the results of each of the pretraining methods once they have been fine-tuned on Ta-En parallel data: the results are very similar and all methods perform substantially better than the baseline, which is trained on parallel data only

[4] An alphabetic character is one belonging to the language in question: the Latin alphabet for English and the Tamil script, which is an abugida script.

[5] All models are learnt jointly over the languages used for training (English, Tamil and in one case German too). The vocabulary size is dependent on the model trained and is specified in the experimental details below.

but optimised in terms of training parameters and subword segmentation. We choose to use De-En pretraining for our final models and a mixture of De-En and mBART pretraining for intermediate MT models used for data augmentation (see the next paragraph).

Iterative backtranslation Data augmentation by backtranslating monolingual data has long been used in MT to provide greater amounts of in-domain parallel data in low resource settings (Bertoldi and Federico, 2009; Bojar and Tamchyna, 2011). We use backtranslation to translate the monolingual in-domain English and Tamil texts into the other language using an intermediate MT model and use the resulting synthetic parallel data to train new MT models.

We apply this iteratively (Hoang et al., 2018), as shown in Figure 1, to produce successively better MT models, initialising the models at each stage using either mBART or De-En pretraining. The intermediate MT models used to produce backtranslations are in white and the final models, which are then fine-tuned (as specified in the section entitled *Final model creation*) are in grey.

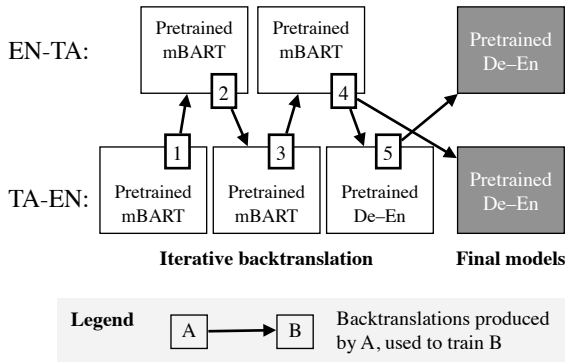

Figure 1: Iterative backtranslation process

1. We first train a Ta→En model initialised with mBART pretraining and fine-tuned on parallel data only. We then use this model to backtranslate all monolingual Tamil data into English.

2. We use the resulting backtranslated data together with the genuine parallel data to train an mBART-pretrained En→Ta model. After early stopping, we continue training using the genuine parallel data only. We then use this model to backtranslate 5M sentences of in-domain English data into Tamil.[6]

3. We use this new backtranslated data together with the genuine parallel data oversampled 7 times to train a second mBART-pretrained Ta→En model. After early stopping, we continue training using genuine parallel data only. We then use this model to backtranslate all the monolingual Tamil data.

4. We repeat step 2 with this latest backtranslated data, generating the final backtranslations to be used for the Ta→En direction.

5. We use 5M of these final backtranslations along with the Ta-En genuine parallel data oversampled 15 times to fine-tune a De-En pretrained model and use this to generate the final backtranslations to be used for the En→Ta direction.

The results of the iterative backtranslation steps on the dev set can be found in Table 4. They show increasing BLEU scores at each successive step.

System	Pretraining	BT dataset	EN→TA dev	EN→TA test	TA→EN dev	TA→EN test
1	mBART	-	-	-	14.00	13.40
2	mBART	1	10.50	5.68	-	-
3	mBART	2	-	-	18.60	**15.19**
4	mBART	3	**11.30**	**6.65**	-	-
5	De-En	4	-	-	**19.30**	-

Table 4: Results (BLEU scores) for the successive models used for backtranslations (BT) (as shown in Figure 1). Each row uses backtranslations produced by the system from the previous row.

System	EN→TA
Parallel-only baseline	5.10
(i) mBART pretraining	7.40
XLM BT	9.90
mBART BT	10.50
De-En BT	10.40
(ii) De-En pretraining	7.30
XLM BT	9.30

Table 5: Dev set results (BLEU scores) for alternative backtranslation schemes for system 2 from Figure 1.

In addition to the described strategy, we also experimented with training different pretrained models using backtranslations produced by different

[6]The En→Ta backtranslations at this step and the following steps are filtered using the same processing as described in Section 2.1, filtered using dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018) and the top sentences are selected to train the next step.

models (e.g. training an mBart pretrained model on XLM-produced backtranslations). We report a small selection of these experiments here for one of the backtranslation steps, comparing the use of alternatives to system 2 (from Figure 1). These results are shown in Table 5: (i) a pretrained mBART model trained on backtranslations from each of the pretrained models, and (ii) a pretrained De-En model trained on XLM backtranslations. For this particular step of the iterative process, training a pretrained mBART model on backtranslations produced by the pretrained mBART model produced the best scores, explaining why this was chosen.

Final model creation Our final models are pretrained De-En models (in grey in Figure 1). After pretraining, the finalisation of these models follows a two-step training strategy to incorporate the synthetic and genuine parallel data:

1. We first train our models on the synthetic data described previously (20M sentences for Ta→En and just over 2.1M sentences for En→Ta).

2. We then fine-tune the models on a mixture of parallel and synthetic data.

This approach of pre-training on synthetic data and fine-tuning on genuine and synthetic data has been found to work well for other tasks (Junczys-Dowmunt and Grundkiewicz, 2016; Grundkiewicz et al., 2019). For the second step, we adopt different strategies for each language direction, depending on which worked best. For Ta→En, we fine-tune on genuine parallel data and 500k of the top scored backtranslations.[7] For En→Ta we fine-tune on a mixture of genuine parallel data, synthetic data produced using multi-agent dual learning (MADL; Wang et al., 2019; Kim et al., 2019) and the top 1M backtranslations. This MADL data comprises a mixture of forward translations and backtranslations of the parallel data created using intermediate models in both directions.

We also carried out preliminary experiments with multilingual training using other Indian languages and experiments with phrase-based MT using Moses (Koehn et al., 2007) but they did not achieve good results.

[7]Scoring is done using dual conditional cross-entropy filtering as specified in Footnote 6.

2.3 Experimental settings

We use the Marian toolkit (Junczys-Dowmunt et al., 2018) for all models except for those using XLM pretraining, for which we use the Facebook XLM toolkit (Lample and Conneau, 2019b). All models trained (including those used to produce backtranslations) use the Transformer-base architecture (Vaswani et al., 2017) with default hyperparameters according to the Marian or XLM implementation (6 encoder and 6 decoder layers, embedding dimension of 512, 16 heads, feedforward dimension of 2,048, standard learning rate warm-up).

Parallel-only baseline Our parallel-only baseline is trained with a joint vocabulary of size 5,414 for Ta-En and 418 for En-Ta. The En-Ta model was trained with a small batch size of 1000 tokens. mBart and XLM models are trained with a joint vocabulary size of 20,000 SentencePiece BPE subwords (including special tokens for language and domain tags, masking and sentence separators).

mBART training English and Tamil sentences are mixed in equal amounts in each batch. We use our re-implementation of mBART using Marian.[8] We deviate from the original implementation by always using two sentences per input segment, whereas the original paper used as many sentences as they could fit into the 512-token limit. The noise hyperparameters are the same as the original paper (35% of tokens are masked in contiguous spans of an average of 3.5 tokens. Masked spans do not cross sentence boundaries). Unlike XLM, we do not use online backtranslation during pre-training. We train until early stopping based on an held-out non-parallel dataset generated using the same noise function as the training data. During monolingual pretraining we early stop after the validation score (measured every 5,000 updates) does not improve for 10 consecutive times. When training on backtranslations or finetuning on parallel data we early stop on the parallel development corpus, measuring the valdation score every 500 updates.

De-En pretraining For models with De-En pretraining, we trained a SentencePiece model with a vocabulary size of 32k on roughly equal amounts of Tamil, English and German data (subsampling Ger-

[8]We implement an online "training harness" that reads monolingual sentences in English and Tamil, converts them to mBART training examples by applying noise and sends them to the Marian training process. Code and training scripts: https://github.com/Avmb/marian-mBART

man and English). The final MT vocabulary size is 49,213 as it is based on using all German-English data for training. The models are trained using tied target embeddings, a learning rate of 0.0002, the Adam optimiser (Kingma and Ba, 2015) and optimiser delay of 2 on 4 GPUs. We train all models until convergence based on the BLEU score on the held-out dev set provided for the task.

2.4 Results

Table 6 shows the final automatic evaluation score of our submissions for both directions on the dev set and the test set, including an ablation of the various components: pretraining using the De-En MT data (and fine-tuned on parallel data), addition of synthetic data to this setup and finally fine-tuning of the resulting model as specified previously.

System	EN→TA		TA→EN	
	dev	test	dev	test
Parallel-only baseline	5.10	3.10	10.10	10.60
Our final models				
Pretraining (De-En)	7.30	5.00	13.60	14.20
+ synthetic data	11.90	7.90	18.80	12.60
+ fine-tuning	**12.30**	**8.40**	**21.00**	**16.60**

Table 6: EN↔TA results (BLEU scores) for the successive steps in the creatino of our final models. The Last row represents the primary submission systems.

The best results (8.40 for En-Ta and 16.60 for Ta-En) are achieved with all three approaches to training. We found that ensembling did not improve our results and therefore our submitted systems are single models. We note that our final approach sees a big difference in the BLEU score between the dev and test sets. While BLEU scores are not directly comparable across datasets, the drop is quite significant and could indicate a domain shift between the two sets. Our models rely heavily on the use of backtranslated data and therefore could be adapting to translationese, which is rewarded in the dev set but not in the test set.

3 English↔Inuktitut

Compared to English-Tamil, the English-Inuktitut language pair is relatively well-resourced at approximately 1.3M sentence pairs. As such we were able to train conventional bilingual Transformer systems, which formed the basis of our submission. We also trained multilingual systems, but opted not to use these in our submission as results on the dev set did not appear to be promising (although evaluation proved challenging for this pair due to overlap between the training and dev data). Post-submission evaluation showed that our multilingual systems actually outperformed our submitted systems on the test sets.

3.1 Data and Preprocessing

We used all of the Nunavut Hansard data provided by the task organisers. For Inuktitut→English, this was supplemented with a similar volume of synthetic data, back-translated from the English side of the Europarl and News Crawl corpora. The only additional monolingual Inuktitut data was 163k sentences of common-crawl data, which we back-translated for the English→Inuktitut system.

We developed two multilingual systems: English→ {Inuktitut,German,Russian} and {Inuktitut,German,Russian} → English. The Russian and German languages were selected due to the availability of suitable volumes of data in the domains of interest (news and parliamentary proceedings). Both multilingual systems used the same dataset, which contains genuine iu-en, synthetic iu-en, genuine de-en, and genuine ru-en in a ratio of approximately 1:1:2:2 (both systems used all of the synthetic data, regardless of back-translation direction). Table 7 lists all of the corpora used for the multilingual systems

Lang. Pair	Size	Corpus
en-iu	1,310k	Nunavut Hansard
en-iu	650k	Synthetic (from en Europarl)
en-iu	650k	Synthetic (from en News 2019)
en-iu	163k	Synthetic (from iu CommonCrawl)
en-de	361k	News Commentary
en-de	1,817k	Europarl
en-de	400k	Paracrawl
en-ru	1,000k	Yandex
en-ru	1,600k	UN

Table 7: Data used for the multilingual English-Inuktitut models. Size is given in sentence pairs.

For the bilingual systems, our preprocessing pipeline consisted of corpus cleaning and segmentation. For corpus cleaning, we used the `clean-corpus-n.perl` script from the Moses toolkit (Koehn et al., 2007). This applies a maximum length threshold of 80 as well as removing empty sentences and sentence pairs with length ratios greater than 9:1.

For segmentation, we trained language-specific SentencePiece models (Kudo and Richardson, 2018) with a vocabulary size of 32,000 BPE subwords and a vocabulary threshold of 50.

Preprocessing was identical for the multilingual systems except that for the English→ {de,iu,ru} we added a token to each source sentence to specify the target language (as in Johnson et al. (2017)).

After the release of the test set, the task organisers reported that some test sentences had been enclosed in extraneous quotes. For our submission, and for post-submission evaluation, we removed outer quotes prior to translation for any test sentence that began and ended with a double quote character.

3.2 Experimental Settings

We used the Nematus toolkit (Sennrich et al., 2017) for all models. For preliminary systems, our hyperparameter settings matched the 'base' configuration of Vaswani et al. (2017). We used these systems for back-translation. For the multilingual systems and final bilingual systems, our settings matched Vaswani et al. (2017)'s 'big' configuration. We used a batch size of 16,384 tokens for all models.

Since the bilingual 'big' systems looked the most promising during development, we trained a second model for each direction and used ensembling in our submission systems.

3.3 Results

Table 8 shows the automatic evaluation scores for our submitted ensemble systems as well as individual bilingual systems and multilingual systems.

Post-submission evaluation on the test set shows that the multilingual systems outperformed the bilingual systems, which is in contrast to the results obtained on the dev sets during system development. We suspect that the large differences in BLEU between dev/test and bilingual/multilingual to overlap between the Nunavut training and dev data. We found that a large proportion of dev sentences were present in the training data, although many were short, frequently used phrases, such as 'Thank you, Mr. Speaker.' and 'The motion is carried.' During development we tried filtering the dev set to reduce overlap at the sentence level. This lowered the scores, but still produced the same overall order: bilingual big > bilingual base > multilingual and so we used this result to guide our decision on which systems to submit. With hindsight, we suspect that the prevalence of formulaic, but not necessarily identical, constructions in the text may be a complicating factor and that more aggressive filtering of the dev set may have produced more robust results. Compared to the bilingual base or multilingual models, the bilingual big models have more capacity available for memorisation of the training data and it seems that our filtering was not enough to counter this effect.

4 Conclusion

In this submission we focused on a low-resource language pair (English-Tamil) and a medium-resource language pair (English-Inuktitut). All our translation systems are based on the Transformer architecture. We found it beneficial to use monolingual data in the form of backtranslations. In the case of En-Ta, we saw notable gains by using pretraining using both the denoising autoencoding (mBART) objective and multilinguality in the form of German-English pretraining. However, we were not able to gain any quality from multilingual training on data for other Indian languages. For English-Inuktit, multilinguality did not appear to help on the dev set, but was found, post-submission, to help on the test set.

In general, we found that English-Tamil is a much more challenging task, where pretraining is absolutely necessary to reach acceptable quality, while for English-Inuktit reasonable translation quality can be achieved using only parallel data.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825299 (GoURMET), 825303 and the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MT-Stretch). The research presented in this publication was conducted in cooperation with Samsung Electronics Polska sp. z o.o. - Samsung RD Institute Poland.

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.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of*

System	EN→IU			IU→EN		
	dev	dev-filt	test	dev	dev-filt	test
Transformer base	22.0	9.8	7.7	35.9	24.3	22.4
Transformer big (1)	24.9	11.4	8.0	45.4	28.3	21.6
Transformer big (2)	24.8	11.5	7.9	45.8	28.4	21.7
Ensemble of (1) and (2)	27.0	12.9	8.2	48.8	31.0	23.0
Multilingual	16.5	8.0	9.7	34.8	24.5	23.3

Table 8: EN↔IU automatic evaluation results (BLEU) on the WMT20 dev and test sets. We also include results for our filtered version of the dev set.

the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), pages 103–115, Florence, Italy.

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.

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, UK.

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.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy.

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.

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. 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.

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.

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.

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.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR'15, San Diego, California, USA.

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.

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.

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.

Guillaume Lample and Alexis Conneau. 2019a. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Lample and Alexis Conneau. 2019b. Cross-lingual Language Model Pretraining. In *arXiv:1901.07291*.

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.

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.

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.

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

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.

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.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *International Conference on Learning Representations*.