Character Mapping and Ad-hoc Adaptation: Edinburgh's IWSLT 2020 Open Domain Translation System

Pinzhen Chen **Nikolay Bogoychev** **Ulrich Germann**
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
{pinzhen.chen, n.bogoych, ulrich.germann}@ed.ac.uk

Abstract

This paper describes the University of Edinburgh's neural machine translation systems submitted to the IWSLT 2020 open domain Japanese↔Chinese translation task. On top of commonplace techniques like tokenisation and corpus cleaning, we explore character mapping and unsupervised decoding-time adaptation. Our techniques focus on leveraging the provided data, and we show the positive impact of each technique through the gradual improvement of BLEU.

1 Introduction

The University of Edinburgh presents its neural machine translation (NMT) systems for the IWSLT 2020 open domain translation task (Ansari et al., 2020). The task requires participants to submit systems to translate between Japanese (Ja) and Chinese (Zh), where the sentences come from mixed domains. For training purpose, 1.96 million existing sentence pairs and 59.49 million crawled sentence pairs[1] are provided, making the task a high-resource one. In our experiments, we focused on three aspects:

1. Corpus cleaning which consists of handcrafted rules and cross-entropy based methods.
2. Japanese and Chinese character mapping to maximise vocabulary overlap and make embedding tying more intuitive.
3. Unsupervised ad-hoc adaptation during decoding time to translate a multi-domain test set, experimented at the sentence, cluster (subdocument) and document levels.

Our techniques are mostly data-centric and each technique improves translation in terms of BLEU on our development set. In the final automatic evaluation based on 4-gram character BLEU, our systems rank 6th out of 14 for Ja→Zh and 7th out of 11 for Zh→Ja.

2 Baseline with Rule-Based Cleaning

2.1 Preprocessing

We first tokenise our data at word-level, which is commonly done for the Japanese and Chinese (Barrault et al., 2019; Nakazawa et al., 2019). While it is unclear whether word-level or character-level models are superior (Bawden et al., 2019), word-level segmentation could resolve ambiguity and dramatically reduce sequence length. The tools we use are `KyTea` (Neubig et al., 2011) for Japanese and `Jieba_fast`[2] for Chinese.

2.2 Rule-based cleaning

We then apply a series of rule-based cleaning operations on both existing and crawled data to create baseline models. These steps are mostly inspired by submissions to the corpus filtering task at WMT 2018 (Koehn et al., 2018). The task shows that effective corpus filtering brings substantial gain in translation performance.

Language identification: One way of parallel corpus filtering is to restrict source sentences to be in the source language, and target sentences to be in the target language. However, distinguishing between Japanese and Chinese, particularly short sentences, is tricky because both share a set of common characters. Hence, we decide to relax this rule by keeping all sentences (pairs) which are identified as either Chinese or Japanese using `langid.py` (Lui and Baldwin, 2012). This inevitably leaves

[1] We mistakenly used an outdated dataset which is larger but noisier. The dataset was extracted from crawled texts with encoding issues and inconsistent handling of Japanese characters "ブ" and "で".

[2] https://github.com/deepcs233/jieba_fast, a faster implementation of `Jieba`

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 122–129
July 9-10, 2020.
https://doi.org/10.18653/v1/P17

some Chinese on the Japanese side and vice versa. It might have a beneficial copying effect (Currey et al., 2017), especially given the vocabulary overlap between the two languages.

Length ratio: We use the provided high-quality existing data to estimate the average Japanese and Chinese sentence lengths at character-level. We find the length ratio of Japanese to Chinese is about 1.4 to 1. We remove sentence pairs which have a length ratio outside the 3 standard deviations from this mean. This a lenient choice in order to keep short translations. This is applied to both existing and crawled data.

Sentence length: We remove sentence pairs with more than 70 tokens on the Chinese side or more than 100 tokens on the Japanese side, for both existing and crawled data.

Chinese simplification: The Chinese datasets contain both traditional and simplified characters, so we use `hanziconv`[3] to simplify them. This rule-based converter has a minor flaw that it sometimes confuses on characters that are in both traditional and simplified Chinese. An example is "著", the traditional form of "着", but also a simplified character on its own with a different meaning.

2.3 Model training

For the baseline model, we try out three combinations of data, namely existing only, crawled only and both. For Ja→Zh and Zh→Ja, this results in six models. As a comparison, we also train vanilla models without previously described cleaning steps.

All models are Transformer-Base with default configurations (Vaswani et al., 2017). We use `Marian` (Junczys-Dowmunt et al., 2018) to train our systems, with SentencePiece (Kudo and Richardson, 2018) applied on tokenised data. As stated previously, Chinese and Japanese share some characters, so it is intuitive to use a shared vocabulary between source and target, and to enable three-way weight-tying between source, target and output embeddings (Press and Wolf, 2017).

We report character-level BLEU on development set, using the evaluation script provided.[4] The baseline results are shown in Table 2 as "(1) vanilla" and "(2) rule-based cleaning". We see a significant improvement in BLEU after applying rule-based cleaning. BLEU scores reported for the development set are based on tokenised output, but we perform de-tokenisation and normalisation of full-width numbers and punctuation symbols for our final submission to make the texts natural Chinese or Japanese.

[3] https://github.com/berniey/hanziconv

[4] https://github.com/didi/iwslt2020_open_domain_translation/tree/master/eval

3 Chinese and Japanese Mapping

In ancient times, Japanese borrowed (at that time, traditional) Chinese characters (Hanzi) to use as a written form (Kanji). After a long time of co- and separate evolution (e.g. Chinese simplification), the relationship between Hanzi and Kanji is complicated. Some Hanzi and Kanji stay unchanged, some develop different meanings, and some develop different written forms. A detailed description is given by Chu et al. (2012). More importantly, they released a Kanji to traditional and simplified Hanzi mapping table. With each Kanji being a key, there can be zero, one or many corresponding traditional and simplified Hanzi. In total, there are mapping entries for around 5700 Kanji to simplified Hanzi. Chu et al. (2013) use this character mapping to enhance word segmentation in statistical machine translation (SMT). Recently, Song et al. (2020) map characters in a Chinese corpus to Japanese, making it a pseudo-Japanese corpus for the purpose of pre-training Japanese↔English NMT.

In our work, we take a step forward to map Chinese and Japanese to each other for Chinese↔Japanese NMT directly. Without mapping as a data processing step, an NMT system needs to learn the mapping between Kanji and Hanzi implicitly. Therefore we hypothesise that mapping them before training a model will:

1. maximise character overlap percentage, reduce vocabulary size and make embedding-tying more effective, and
2. reduce the computation needed to learn to model the mapping.

Since we already simplified all Chinese characters, hereafter we refer to simplified Chinese as Hanzi. Mapping from Kanji to Hanzi is straightforward from the character mapping table. Next, according to the mapping table, we re-construct a mapping table indexed by Hanzi, but a minor difference is that each Hanzi will have at least one corresponding Kanji. It is not possible to get perfect one-to-one mappings due to the existing many-to-many

	Chinese→Japanese				Japanese→Chinese			
	Zh	**Ja**	**Total**	**Overlap**	**Zh**	**Ja**	**Total**	**Overlap**
no mapping	21168	18502	24387	15283	21168	18502	24387	15283
conservative	20958		24117	15343		16659	22891	14936
aggressive	20560		24086	14976		16341	22759	14750

Table 1: Character statistics of Chinese (Zh) and Japanese (Ja)

relationship between Hanzi and Kanji. In order to simplify post-processing. we only map source characters to target, so the target outputs are always in the genuine target language. Hence we map Chinese to Japanese or Japanese to Chinese depending on the translation direction. We design two simple mapping scheme variants:

1. **Conservative mapping**: apply one-to-one mapping and ignore all one-to-many cases. All target characters must be constrained to target corpus, in order not to introduce new characters.
2. **Aggressive mapping**: apply one-to-one mapping, and for the one-to-many mapping cases, pick the character that has the highest frequency in the target corpus. The target constraint applies too.

Table 1 shows the counts of characters before and after mapping in each language as well as the total counts, for Chinese→Japanese and Japanese→Chinese respectively, on all available data. We only map characters on the respective source side and leave the target side of the training data as it is.

We then train models on the mapped data for both directions, with results displayed in Table 2 as "(3) mapping". We observe that aggressive mapping is marginally better than conservative on Ja→Zh and much better on Zh→Ja. Thus, we pick aggressive mapping for our following experiments.

4 Filtering Based on Cross-Entropy

Our initial rule-based cleaning shows its effectiveness through improvement in BLEU scores. We further adopt two filtering steps based on cross-entropy proposed by Junczys-Dowmunt (2018):

4.1 Dual conditional cross-entropy

Dual conditional cross-entropy score is obtained from the absolute difference between cross-entropies of two translation models in inverse directions, weighted by the sum of cross-entropies of the two models. The score of a sentence pair (x, y) is calculated according to Equation 1, where $H_{a\to b}(b|a)$ is the cross-entropy from a translation model that translates a to b. A lower score implies a better sentence pair.

$$\begin{aligned}\text{adequacy} = &\left|H_{x\to y}(y|x) - H_{y\to x}(x|y)\right| \\ &+\frac{1}{2}(H_{x\to y}(y|x) + H_{y\to x}(x|y))\end{aligned} \tag{1}$$

This step finds sentence pairs that are adequate, and more importantly, equally adequate in both directions. It effectively filters out non-parallel sentences, or even machine translations which have been optimised for just a single direction. We want to score sentence pairs with the best translation model we have, so we use the aggressive mapping models built in the previous section to score mapped corpus for both directions.

4.2 Language model cross-entropy difference

The previous step ensures the adequacy of sentence pairs, but it does not pick out unnatural sentences. For example, a concatenation of texts from a website's navigation bar, together with its translation, get a good score by fulfilling adequacy. To alleviate this issue, we apply cross-entropy difference scoring. The score for a single sentence a is calculated according to Equation 2, where $H_{desired}(a)$ is the cross-entropy from a language model trained on desired data (clean, in-domain) and $H_{undesired}(a)$ is the cross-entropy from a language model trained on undesired data (noisy, out-of-domain). It has an interpretation that, a high-quality sentence should be similar to the desired data but different from the undesired data. We used `KenLM` (Heafield et al., 2013) to build 4-gram language models on the existing and the crawled data respectively.

$$H_{\text{desired}}(x) - H_{\text{undesired}}(x) \tag{2}$$

Since our data serve both translation directions, we score both sides of a sentence pair and take the

Category	Data	Transformer size	BLEU Ja→Zh	BLEU Zh→Ja	
(1) vanilla	existing	base	21.88	27.11	
(2) rule-based cleaning	existing	base	26.57	26.59	
	crawled		25.15	27.25	
	all		28.26	27.70	√
(3) mapping	conservative	base	29.09	24.37	
	aggressive		29.41	27.78	√
(4) cross-entropy filtering	best 50M	base	29.66	28.84	
	best 35M		30.45	28.92	
	best 20M		30.58	29.67	√
(5) deeper models	best 20M	big	30.91	30.13	
	best 10M		30.65	30.42	√ (a)
	best 10M		30.68	30.40	(b)
	best 5M		30.35	29.94	(c)
	best 5M		29.71	30.08	(d)
(6) ensembles	ensemble of (c) and (d)		30.63	30.55	
	ensemble of (a) and (b)		31.55	30.86	
	ensemble of (a), (b), (c) and (d)		31.61	30.90	

Table 2: Our models' 4-gram character-level BLEU on development set. A √ symbol denotes the best configuration in each category.

sum to get an overall fluency score:

$$\begin{aligned}\text{fluency} = & H_{\text{existing\_ja}}(ja) - H_{\text{crawled\_ja}}(ja) \\ & + H_{\text{existing\_zh}}(zh) - H_{\text{crawled\_zh}}(zh)\end{aligned} \quad (3)$$

4.3 Ranking and cut-off

To combine both filtering methods, Junczys-Dowmunt (2018) negates the scores and exponentiate them. Furthermore, extreme cross-entropy difference scores are capped or cut to 0. Finally, a product of the two determines the quality of sentence pairs. After applying this procedure, we observe that the top-ranking sentences are dominated by the ones with perfect adequacy but not fluency (e.g. a translation of navigation bar). Thus we keep multiplication but omit capping and cutting to weight fluency more. Equation 4 shows how the final score of a sentence pair is calculated.

$$\text{score} = \exp(-\text{adequacy}) \times \exp(-\text{fluency}) \quad (4)$$

After we rank all sentences pairs by their scores, we empirically determine the data cut-off point. We test with top 50, 35 and 20 million sentence pairs with Transformer-Base architecture for both translation directions. We report BLEU scores in category "(4) cross-entropy filtering" in Table 2, where we observe that translation performance improves as the size of training data drops. Thus we further experiment with 20, 10 and 5 million data on Transformer-Big. Results are displayed in the same table under category "(5) deeper models". In addition, we run ensemble decoding, combining the models trained on 10 million and 5 million sentences, and report results in the same table in category "(6) ensembles".

5 Ad-hoc Domain Adaptation

NMT is sensitive to domain mismatch (Koehn and Knowles, 2017), and there are numerous techniques for domain adaptation for NMT (Chu and Wang, 2018). Some model and training techniques require prior knowledge of the domain and cannot be easily applied. Nonetheless, one method that can be adopted during test sentence translation is retrieving samples that are similar to the input from the available training data, and fine-tuning a trained generic model on these samples. Such ad-hoc domain adaptation can be done at sentence level (Farajian et al., 2017; Li et al., 2018) or document level (Poncelas et al., 2018).

5.1 Similar sentence retrieval

A crucial factor for domain adaptation to work is to accurately retrieval representative sentences of

test sentences. Farajian et al. (2017) store training data in the Lucene search engine and take the top-scoring outcomes ranked by sentence-level BLEU. Li et al. (2018) use word-based reverse indexing and explore three similarity measures: Levenshtein Distance, cosine similarity between average word embeddings, and cosine similarity between sentence embeddings from NMT. Additionally, they suggest an alternative approach, phrase coverage, inspired by phrase-based SMT, when no high-scoring match is found.

Sentence-level adaptation is computationally expensive because, for each sentence, a separate model needs to be fine-tuned. In contrast, Poncelas et al. (2018) synthesise data similar to the whole test set. They leverage a feature decay algorithm to select monolingual data in the target language that are similar to test sentences translated by a generic source-to-target model. Then, the selected sentences are back-translated to source language (Sennrich et al., 2016), forming synthetic parallel sentences for fine-tuning.

In our work, we adopt a pure phrase-coverage approach, which is compatible for both sentence and document level retrieval. As originally suggested for phrase-pair extraction in phrase-based SMT by Callison-Burch et al. (2005) and Zhang and Vogel (2005), we index the source side of the training data via a suffix array (Manber and Myers, 1990) for very fast identification of sentence pairs that contain a given phrase. Then we simply use the test data as a query to retrieval sentences based on n-gram overlapping. Figure 1 shows how efficiently our sentence retrieval method scales up.

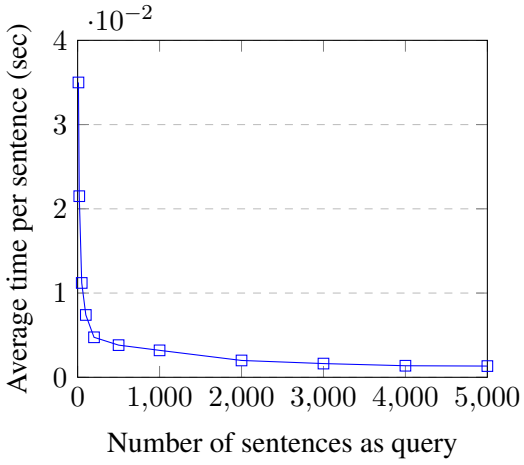

Figure 1: Average time to query one sentence against number of sentences in the query.

We set a threshold T, such that n-grams which occur more than T times in training data are disregarded, under the assumption that the generic model will already have learned to translate such phrases adequately. This is similar to Li et al. (2018)'s approach, but we try different T values. For other n-grams, we always include all matching sentences in the fine-tuning data.

5.2 Fine-tuning experiments

Due to time constraint, we only experiment our on-the-fly fine-tuning on Ja→Zh. We pick the generic baseline model to be the best-performing one trained on 10 million data. We test three different ways of doing the adaptation. First is the single-sentence adaptation, where the generic model is fine-tuned on selected training sentences for each sentence in development (dev) set. However, careful choice of hyperparameters is necessary to prevent overfitting because only a small number of sentences are retrieved. Next thing we try is to use 1 dev sentence and other 9 closet dev sentences together as a query. To form such a cluster of 10 dev sentences, we convert all dev sentences into n-gram TF-IDF vectors and score cosine similarity in a pairwise manner. This allows us to find the most similar sentences to any given one. For the above two choices, we set the threshold T to be 20, and fine-tune for 1 and 10 epochs separately. The results are reported in Table 3.

We observe that BLEU drops even we only fine-tune for a single epoch. Our intermediate conclusion is that there is overfitting or misfitting to out-of-domain sentences that have been incorrectly retrieved. Furthermore, sentence-level adaptation is fairly expensive, which prevents us from performing a grid search to find the most suitable configurations. Hence, we move on to document-level adaptation by using the whole dev set as a query to find similar sentences. As a comparison, we also use the whole test set, and a combination of dev and test as queries. This results in hundreds of thousands of sentences being retrieved, compared to hundreds to thousands for sentence-level retrieval. To prevent overfitting, we also raise threshold T to 120 and validate on dev set frequently instead of specifying an epoch budget.

As Table 3 shows, using a query of both dev and test data leads to the biggest improvement of 0.55. Surprisingly, using the whole test set as a query to retrieve sentence for dev set fine-tuning only leads to a small drop of 0.19 BLEU. This shows that our

Query	T	Epoch	BLEU
generic baseline			30.65
1 sentence	20	1	26.60
	20	10	26.25
a cluster of	20	1	25.81
10 sentences	20	10	27.24
dev set	120		31.09
test set	120	N/A	30.46
dev and test sets	120		31.20
ensemble	4 FT		32.12
	4 FT & 4 non-FT		32.06

Table 3: Character-level BLEU of ad-hoc fine-tuning experiments on Ja→Zh, at sentence, cluster and document levels. FT denotes fine-tuned models.

document adaptation is conservative, thanks to a large number of retrieved sentences. The considerations underlying adaptation over the entire dev and test sets (irrespective of the domain of individual sentences) are as follows: very frequent phrases including words, are the features of a language rather than a domain. For phrases that are frequent in some domains but not others, the generic model will probably have learned to translate them appropriately. What we are concerned about are the phrases seen rarely during generic model training, because of the bias in training data, or coming from niche domains. Sentences that share such phrases, we conjecture, are likely from the same or related domains anyway, so fine-tuning on them all is effective. For sentences with no overlap in such words and phrases, we are probably fine-tuning different areas in the overall parameter space, which can be harmless to each other.

6 Results and Conclusion

In our work, we explore a series of techniques which lead to improvements on Ja↔Zh NMT. Rule-based filtering brings a marginal increment in BLEU for Zh→Ja but a significant one for Ja→Zh. Character mapping, which increases source and target vocabulary overlap, has a tiny effect on Zh→Ja, but makes 1 BLEU improvement for Ja→Zh. Next, cross-entropy filtering adds 2.5 BLEU for Zh→Ja and 2 BLEU for Ja→Zh. Ad-hoc fine-tuning, aiming at enhancing open domain translation, delivers another 0.55 BLEU. Finally, an ensemble of 4 fine-tuned models boosts up 1 BLEU. Overall, our work has improved 10 and more than 3 BLEU for Ja→Zh and Zh→Ja respectively.

Character mapping between Japanese and Chinese may inspire two directions of research: applying character mapping on other tasks, and trying character mapping for other language pairs.

Due to time constraint, we could not perform exhaustive experiments to find the best configuration for sentence-level and cluster-level adaptation, which can be further investigated. We also propose to study further on cluster (sub-document) adaptation, where a system can group test sentences, and fine-tune before translating them. This can make adaptation more fine-grained compared to document adaptation, without the huge risk of overfitting at sentence-level.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825303 (Bergamot) and 825627 (European Language Grid).

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 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). We thank Kenneth Heafield for providing us with computing resources.

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay

Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, and Changhan Wang. 2020. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA.

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

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 255–262, Ann Arbor, Michigan. Association for Computational Linguistics.

Chenhui Chu, Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2013. Chinese-japanese machine translation exploiting chinese characters. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):1–25.

Chenhui Chu, Toshiaki Nakazawa, and Sadao Kurohashi. 2012. Chinese characters mapping table of Japanese, traditional Chinese and simplified Chinese. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2149–2152, Istanbul, Turkey. European Languages Resources Association (ELRA).

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.

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.

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.

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. 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.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. 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.

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.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*,

pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Udi Manber and Gene Myers. 1990. Suffix arrays: A new method for on-line string searches. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '90, page 319–327, USA. Society for Industrial and Applied Mathematics.

Toshiaki Nakazawa, Nobushige Doi, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th workshop on Asian translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 1–35, Hong Kong, China. Association for Computational Linguistics.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA. Association for Computational Linguistics.

Alberto Poncelas, Andy Way, and Kepa Sarasola. 2018. The ADAPT system description for the IWSLT 2018 Basque to English translation task. In *International Workshop on Spoken Language Translation*, pages 72–82, Bruges, Belgium.

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.

Haiyue Song, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. 2020. Pre-training via leveraging assisting languages and data selection for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. To appear.

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.

Ying Zhang and Stephan Vogel. 2005. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *In Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT-05)*, pages 294–301.