OPPO's Machine Translation System for the IWSLT 2020 Open Domain Translation Task

Qian Zhang and **Tingxun Shi** and **Xiaopu Li** and **Dawei Dang** and
Di Ai and **Zhengshan Xue** and **Jie Hao**
{zhangqian666, shitingxun, lixiaopu, dangdawei
aidi1, xuezhengshan, haojie}@oppo.com
Manifold Lab, OPPO Research Institute

Abstract

In this paper, we demonstrate our machine translation system applied for the Chinese-Japanese bi-directional translation task (aka. open domain translation task) for the IWSLT 2020 (Ansari et al., 2020). Our model is based on Transformer (Vaswani et al., 2017), with the help of many popular, widely-proved effective data preprocessing and augmentation methods. Experiments show that these methods can improve the baseline model steadily and significantly.

1 Introduction

Machine translation, proposed even before the first computer was invented (Hutchins, 2007), has been always a famous research topic of computer science. In the recently years, with the renaissance of neural network and the emergence of attention mechanism (Sutskever et al., 2014) (Bahdanau et al., 2014), the old area has stepped into a new era. Furthermore, the Transformer architecture, after being published, has immediately attracted much attention nowadays and is dominating the whole field now.

Although Transformer has achieved many SOTA results, it has tremendous amount of parameters so is hard to be fit on small datasets, therefore it has a high demand on good and large data source. Despite of the lack of high quality data of parallel corpus, the document level comparable data is relatively easy to be crawled, so exploring an effective and accurate way of mining aligned sentence pairs from such large but noisy data, to enrich the small parallel corpus, could benefit the machine translation system a lot. Besides, currently, most open access big volume machine translation datasets are based on English, and many of them are translated to/from another European language – As many popular European languages are fusional languages, most corpus are composed by two fusional languages together. To understand whether the existing model architectures and training skills can be applied on the translation between Asian languages and other type of languages , such as between an analytic language like Chinese and an agglutinative language like Japanese, interest us.

In this paper, we demonstrate our system applied for the IWSLT 2020 open domain text translation task, which aims to translate Chinese from/to Japanese [1]. Besides describing how we trained the model that is used to generate the final result, this paper also introduces how do we mine extra parallel sentences from a large but noisy data released by the organizer, and several experiments inspired by the writing systems of Chinese and Japanese.

2 Data Preprocessing

Four pairs of parallel data are provided in the campaign, which are

- The very original file, which is crawled from various websites, and is very huge. According to the information of the campaign page, the corpus contains 15.7 million documents, which are composed by 941.3 million Japanese sentences and 928.7 million Chinese sentences — From the counts of sentences it can be immediately observed that the original corpus is not parallel, so cannot be directly used for the model. Mining parallel corpus from this mega size file is another work we have done during the campaign, which will be covered in another section of this report.

- A pre-filtered file, consists of 161.5 million "parallel" sentences. We tried to filter this

[1] In some cases later in the report, the two languages are noted by their ISO 639-1 codes, `zh` and `ja` respectively

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 114–121
July 9-10, 2020.
https://doi.org/10.18653/v1/P17

dataset to extract parallel lines, and this work will also be presented later.

- A filtered file, which has 19 million sentences, is aligned officially from the data described in the previous item. And,
- An existing parallel file, which contains 1.96 million pairs of sentences, is obtained by the provider from current existing Japanese-Chinese parallel datasets.

However, per our investigation, even the sentences in the existing parallel file are actually not fully aligned. For example, a sentence "*1994* 年 *2* 月、ジャスコはつるまいの全株式を取得。" (means "Jasco bought all shares in February, 1994") in the corpus is translated into "次年 6 月乌迪内斯买断了他的全部所有权。", which means "Udinese bought out all his ownership in June in the next year", so here is clearly a noise. Since deep neural network demands high quality input data, we combined the filtered file and the existing parallel file into a 20 million pairs dataset (noted as *combined dataset* afterwards) , and made a further data preprocessing, including two main steps:

2.1 Rule Based Preprocessing and Filtering

We first feed the combined dataset to a data preprocessing pipeline, including the following steps:

- Converting Latin letters to lower case. This step helps to decrease the size of the vocabularies, but since the evaluation is case-sensitive, we applied a further post-processing step: Having generated the results from the model, we extract all Latin words from the sources and the hypotheses, and convert the words in the hypo side according to the case forms of their counterparts in the source side.
- For Chinese, converting traditional Chinese characters to simplified form; for Japanese, converting simplified Chinese characters to kanji.
- Converting full width characters to half width.
- Normalizing punctuations and other special characters, e.g. different forms of hyphen "-".
- Unescaping html characters.
- Removing html tags.
- Removing extra spaces around the dot symbol of float numbers
- Removing unnecessary spaces

Because both Chinese language and Japanese language don't use spaces to mark borders of words, we applied segmentation on each side (A branched experiment will be presented later in this report). For Chinese, we use *PKUSEG* (Luo et al., 2019) and for Japanese it is *mecab*[2]. After having observed the preprocessed data, sentence pairs are filtered out according to the following orders:

1. Sentences that contain too many non-sense symbols (including emojis ,kaomojis and emoticons, such as "(@ ^ □ ^)". Although these symbols could bring semantic information, we don't consider they are important to machine translation system)
2. Sentence pairs that have abnormal length ratio, here "length" is the count of words of a sentence. As Chinese character is also an important constituent of Japanese writing system, we don't expect the Japanese sentences will be too much longer than the Chinese side; however in another hand, since Japanese is an agglutinative language, it always needs several additional (sub)words to express its own syntactical structure, so the Japanese sentences can neither be too short. We set the upper bound of words count ratio between Japanese and Chinese to 2.4 and the corresponding lower bound is 0.8.
3. Sentence pairs that occur more than once. We deduplicated and left only one single pair.
4. Sentence pairs that target is simply a replica of the source sentence.
5. Sentence pairs that target sentence shares the same beginning or ending 10 characters with source sentence.
6. Sentence pairs that the amount of Chinese words is less than 40% of the total word count in the Chinese side. Here "Chinese word" is defined as a word which is composed by Chinese characters only.

[2]https://taku910.github.io/mecab/

7. Sentence pairs that the amount of Japanese words is less than 40% of the total word count in the Japanese side. Here "Japanese word" is defined as a word which is composed by kanjis or kanas only. As Chinese language and Japanese language each has its own special "alphabets", this step together with the previous one can be seen as a way of language detection.

8. Sentence pairs that the count difference between numbers in Chinese side and numbers in Japanese side is greater than or equal to 3

9. Sentence pairs that cannot be aligned on numbers and Latin letters.

2.2 Alignment Information Based Filtering

Processing rules listed in the previous subsection can be applied to filter out sentence pairs that have obvious noises, but some pairs still have subtle noises that cannot be directly discovered. Therefore we use *fast_align* to align the source and target sentences, generate alignment score in the sentence level and word level [3], then further filter the combined dataset by the alignment results. For the sentence level alignment score, the threshold was set to -16 and for the word level it was -2.5. After multiple rounds cleaning, 75% of the data provided are swiped out, leaving about 5.4M sentence pairs as the foundation of our experiments described in the next section.

3 Main Task Experiments

Taking the 5.4M corpus in the hand, we further divided the words in the text into subwords (Sennrich et al., 2016b). BPE code is trained on the Chinese and Japanese corpus jointly, with 32,000 merging operations, but the vocabulary is extracted for each language individually, so for Chinese the size of its vocabulary is 31k and for Japanese it is 30k. Vocabularies for both two directions (`ja-zh` and `zh-ja`) are shared. We trained 8 heads Transformer Big models with Facebook FAIR's fairseq (Ott et al., 2019) using the following configuration:

- learning rate: 0.001
- learning rate schedule: inverse square root
- optimizer: Adam (Kingma and Ba, 2014)
- warmup steps: 4000
- dropout: 0.3
- clip-norm: 0.1

The first model trained on the filtered genuine parallel corpus (i.e. the 5.4M corpus) is not only seen as the baseline model of the consequent experiments, but also used as a scorer[4]. We re-scored the alignment scores of the sentences using this model, and again filtered away about one quarters data. The model trained on the refined data improved the BLEU score by 1.7 for `zh-ja` and 0.5 for `ja-zh`.

As many works proved, back-translation (BT) (Sennrich et al., 2016a) is a common data augmentation method in the machine translation research. Besides, (Edunov et al., 2018) also provides some other ways to back-translate. We applied both of them and in our experiments, top-10 sampling is effective on `zh-ja` direction and for `ja-zh` traditional argmax-based beam search is still better.

Firstly, 4M data in the original corpus is selected by the alignment score and translated by the models (models for the different directions) got in the previous step to build synthetic corpus, then for each direction a new model is trained on the augmented dataset (contains 5.4M + 4M + 4M = 13.4M pairs). To get a better translation result, we used ensemble model to augment the dataset. One more thing could be clarified that, in this augmentation step we not only introduced 4M back-translated data, but also generated 4M synthetic target sentences by applying knowledge distillation (KD) (Freitag et al., 2017).

On this genuine-BT-KD mixture dataset, we tried one more round of back-translation and knowledge distillation, but just saw a minor improvement. Afterwards we trained language model on the 5.4M parallel corpus for each language using kenlm (Heafield, 2011). With the help of the language model, 3M Chinese sentences and 4M Japanese sentences with the highest scores are selected from the unaligned monolingual corpus as the new input of BT models, augmented the mixture dataset to 20.4M pairs (noted as *final augmented dataset*, which will be referenced later

[3] To get a word level alignment score, we divide the sentence level score by the average length of source sentence and target sentence

[4] Many related works used to train a model in the very early stage, for example train from the rawest, uncleaned dataset. We did considered doing so at first but since the original dataset is too noisy, we decided to clean the corpus first to achieve a more meaningful baseline score.

in the report), and we did another round of back-translation and knowledge distillation. After these three rounds iterative BT (Hoang et al., 2018) and KD, several best single models are further composed together to an ensemble model. In the last step, following (Yee et al., 2019), we use both backward model (for `zh-ja` task, model from `ja-zh` is its backward model, and vice versa) and Transformer language model to rerank the n-best candidates of the output from the ensemble model, to generate the final results.

Detailed results on the dev dataset of each intermediate step is shown in table 1. We strictly followed the organizer's requirement to build a constrained system, means that we didn't add in any external data, nor made use of the test data in any other form besides of generating the final result.

4 Branched Task Experiments

Besides the main task experiments demonstrated in the previous section, as the *introduction* part says, we are also interested in how to mine or extract parallel data from such huge but noisy datasets, and explore some special skills on translating from Chinese to Japanese (and also vice versa). This section will mainly discuss our work on these two parts.

4.1 Filtering the Noisy Dataset

We first tried to extract parallel sentences from the pre-filtered, 161.5 million dataset. Since this dataset is "nearly aligned", it is assumed that for a given sentence pair, if the target side doesn't match the source, the whole pair can be safely dropped because the counterpart of the source doesn't exist in other places of the corpus. We first use *CLD* as the language detector to remove sentences that are neither Chinese nor Japanese — only in this step nearly 110 million pairs are filtered out. Next, we feed the data into the preprocessing pipeline which is the same as the one introduced in the *Preprocessing* section. The preprocessed corpus are then filtered in a similar way described in the *Preprocessing* section, with the following additional steps:

- We compared the url counts of each side and remove the inconsistent line pairs.
- We kept a set of common special characters as a white list, removed all other special characters
- We removed the sentence pairs that the source side is too similar to the target side. Concretely, we compared the Levenshtein distance between the sentences, divided it by the average length (count of characters) of the text in the pair. If this ratio is above 0.9, we consider the source and the target are too similar.

After the filtering, 14.92 million sentence pairs are kept, and based on them we trained a model by Marian (Junczys-Dowmunt et al., 2018) using Transformer base model, see it as the baseline model for the current task. 36k BPE merge operations are applied on the remained sentence pairs, independently for each language, led to two vocabularies each contains 50k words. We use Adam optimizer with learning rate set to 3×10^{-4} and 16,000 warmup steps, clip-norm set to 0.5, dropout of attention set to 0.05, label smoothing set to 0.1. Decoder searches with a 6 beam-width and the length normalization is 0.8. [5]

To filter the noisy parallel corpora, We followed dual conditional cross-entropy filtering proposed by (Junczys-Dowmunt, 2018): for a parallel corpus D, in which the source language is noted as $\mathcal{X}$ and the target language is noted as $\mathcal{Y}$, two translation models can be trained: model A is trained from $\mathcal{X}$ to $\mathcal{Y}$ and model B is trained in the reversed direction. Given a sentence pair $(x, y) \in D$ and a translation model M, the conditional cross-entropy of the sentence pair normalized by target sentence length can be calculated:

$$
\begin{aligned}
H_M(y|x) &= -\frac{1}{|y|} \log P_M(y|x) \\
&= -\frac{1}{|y|} \sum_{t=1}^{|y|} \log P_M(y_t|y_{<t}, x)
\end{aligned}
$$

As we have two models A and B, two scores achieved by each can be combined to calculate the maximal symmetric agreement (MSA) of the sentence pair, following:

$$
\begin{aligned}
\text{MSA}(x, y) =& |H_A(y|x) - H_B(x|y)| \\
&+ \frac{1}{2}(H_A(y|x) + H_B(x|y))
\end{aligned}
$$

[5] In the branched experiments, the machine translation framework and hyper-parameters applied are all different from those used in the main task. The reason is these experiments were taken concurrently by different team members, so they have each own hyperparameter settings.

	zh-ja **BLEU**	ja-zh **BLEU**
Baseline	34.6	32.6
+ Filtered by alignment information from baseline model	36.3 (+1.7, +1.7)	33.2 (+0.6, +0.6)
+ 1st round BT using genuine parallel corpus (13.4M pairs)	37.5 (+2.9, +1.2)	34.6 (+2.0, +1.4)
+ 2nd round BT using genuine parallel corpus (13.4M pairs)	37.6 (+3.0, +0.1)	34.6 (+2.0, +0.0)
+ BT using monolingual corpus (20.4M pairs)	38.8 (+4.2, +1.2)	35.4 (+2.8, +0.8)
+ 3rd round BT using both parallel and monolingual corpus (20.4M pairs)	39.2 (+4.6, +0.4)	36.0 (+3.4, +0.6)
+ Ensemble	40.1 (+5.5, +0.9)	36.6 (+4.0, +0.6)
+ Reranking	**40.8 (+6.2, +0.7)**	**37.2 (+4.6, +0.6)**

Table 1: Results of the main task experiments, evaluation is taken on the validation dataset provided officially. The improvement amount of each row is expressed in two forms: absolute improvement (current score - baseline score) and relative improvement (current score - previous step score). Note to get a more strict BLEU score, we used SacreBLEU (Post, 2018) to calculate the final BLEU score, and we didn't split words composed by Latin letters and numbers into characters, which differs from the official evaluation process. If the same splitting is applied, and evaluated by `multibleu` (https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu-detok.perl) which is officially designated, the score could be higher by 1.x points

Since $\text{MSA}(x, y) \in [0, +\infty)$, we can re-scale the score to $(0, 1]$, by

$$\text{adq}(x, y) = \exp(-\text{MSA}(x, y))$$

This method is noted as "adq" adapting the notation proposed in the original paper. We took a series of experiments on the direction `zh-ja`, but the results are not so good as we expected. Detailed information is listed in table 2. We also added dataset C mentioned in table 2 to the original dataset used for training baseline model of the main task, but still didn't see too much improvement. Using the configuration introduced in the *main task section*, the model's BLEU score is 34.7, only 0.1 points higher than the baseline score listed in table 1.

4.2 Mining the Unaligned Dataset

Besides the officially released pre-filtered dataset, we also paid our attention on the very original, huge but dirty dataset, tried some methods to clean it. As previously said, both Chinese and Japanese have its own closed characters set respectively, so we first simply remove the lines that don't contain any Chinese characters (for Chinese corpus), or those don't contain any katas or kanjis (for Japanese lines). This simple step directly removed about 400 million lines. We also applied the same preprocessing described before, like language detection, deduplication, and the cleaning pipeline. This preprocessing reserved 460 million lines.

For the remained data, as they are not aligned, we cannot follow the filtering process shown in the previous sub-section. However, we assumed that for a Chinese sentence, if we can find its Japanese counterpart, the corresponding line can only exist in the same document. As the dataset gives document boundary, we split the whole dataset into millions of documents, and use *hunalign* (Varga et al., 2007) to mine aligned pairs in each document (dictionaries are extracted from a cleaned version of the combined dataset). Although still hold the intra-document alignment assumption, we kept reading documents, didn't perform hunalign until the accumulated lines reached 100k (but we don't break the document), for the possible cross-document alignment. We kept all lines which have alignment scores higher than 0.8, and of which the words count ratio between source and target falls into $[0.5, 2]$. Then we removed all lines contains url, replaced numbers and English words which have more than 3 letters with tags, and deduplicated again, leaving only 5.5 million lines. We trained a Transformer base model using marian on the dataset which is utilized for training the baseline model in the *main task experiments*, applying the same configuration given in the previous sub-section, and ranked the results using *bleualign* (Sennrich and Volk, 2010) (Sennrich and Volk, 2011), finally kept 4 million lines. This dataset is patched to the original dataset which is used the main task, and a minor improvement (+0.6 BLEU) can be seen. However, due to the time limit this part of data were not further used in the whole main task experiments.

Filtering method	BLEU on dev set
Baseline	27.1
A. adq, 8M data with highest scores	**27.2 (+0.1)**
B. adq, 5M data with highest scores	26.2 (-0.9)
C. Filter A by fast_align scores and Japanese language models	26.8 (-0.3)

Table 2: `zh-ja` experiments using data filtered from the pre-filtered "parallel" corpus. BLEU is calculated by sacreBLEU in the same way depicted in the *main task experiments* section

4.3 Character Based Models and Some Variations

From the perspective of writing system research, Chinese characters system is a typical *logogram*, means a single character can also carry meaningful semantical information, which differs to phonologic writing systems widely used in the world. Previous research (Li et al., 2019) argues that for Chinese, character-based model even performs better than subword-based models. Moreover, For the Japanese language, its literacy "was introduced to Japan in the form of the Chinese writing system, by way of Baekje before the 5th century" [6], even today Chinese characters (Hanzi, in simplified Chinese 汉字, in traditional Chinese 漢字) are still important components of Japanese writing system (in Japanese called kanji, written as 漢字), so intuitively characters between two languages could have strong mapping relationship. (ngo, 2019) also shows that for Japanese-Vietnamese machine translation system, character-based model takes advantages to the traditional methods. As both Vietnamese and Japanese are impacted by Chinese language, it is reasonable to try character-based machine translation systems on Chinese ⇔ Japanese language pairs.

Inspired from the intuition and the previous related works, we further split the subwords in the *final augmented dataset* (presented in the main task experiments) into characters in three different ways, which are

- Split CJK characters (hanzi in Chinese and kanji in Japanese) only, since we assume that the characters are highly related between these two sets
- Split CJK characters and *katakana* (in kanji 片仮名). In Japanese writing system, besides kanji, another component is called *kana* (in kanji 仮名), which belongs to syllabic system (one character is corresponding to a syllable). Kana further consists of a pair of syllabaries: *hiragana* (in kanji 平仮名) and katakana, the latter is generally used to transliterate loanword (including foreign names). Although a single katakana character doesn't carry semantical information, only imitates the pronunciation, the same situation exists in Chinese, too — when transliterating foreign names, a single Chinese character is only used to show the pronunciation, loses the means it could have. Therefore, katakanas can also be roughly mapped to Chinese characters.
- Split CJK characters and all kanas

For each direction, we trained four different Transformer Big models using the splitting methods described above (another one is subword-based model as baseline). In this series of experiments, we used FAIR's fairseq, set clipnorm to 0, max tokens to 12,200, update-freq to 8, dropout to 0.1, warmup-updates to 15,000. Length penalties are different among all models, we set the optimal value according to the results reported on the validation set. However, surprisingly, there is still no improvement can be observed, and for `zh-ja` direction models generally perform worse (detailed results are listed in table 3). It needs some extra work to find out the reason, one possible explanation is the big amount of back-translated synthesis corpus, which was generated by model based on subwords, changed the latent data distribution.

5 Final Results

From the evaluation results provided by the organizer officially, Our BLEU score for `jazh` direction is 32.9, for `zhja` is 30.1.

However, per our observation on the dev dataset, we found most of the numbers and Latin words are styled in full width characters, so we made an extra step in post-processing to convert all

[6] https://en.wikipedia.org/wiki/Japanese_language#Writing_system

Splitting method	zh-ja BLEU	ja-zh BLEU
zh-ja Baseline	**39.0**	35.1
Split cjk chars only	37.9 (-1.1)	**35.4 (+0.3)**
+ katakanas	37.9 (-1.1)	34.9 (-0.2)
+ hiraganas	38.2 (-0.8)	35.3 (+0.2)

Table 3: Experiments using character-based models. BLEU is calculated by sacreBLEU in the same way depicted in the *main task experiments* section. The "baseline" model here is trained on the 21M data after three rounds back-translation, compared to zh-ja 39.2/ja-zh 36.0 step in the *main task experiments* section, not to be confused with the baseline demonstrated in the previous section

jazh BLEU	zhja BLEU
55.8	43.0
34.0	**34.8***
32.9	34.3
32.5	33.0
32.3	31.7
30.9	31.2
29.4	**30.1**
26.9	29.9
26.2	28.4
25.3	26.3
22.6	25.9
11.6	7.1
1.8	
0.1	

Table 4: Leaderboard released officially just after the submission. Scores shown in the table are character-level BLEU calculated by multibleu (https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu-detok.perl). Our results are styled in bold and the contrastive one is marked with an additional asterisk symbol $*$. At the date of submitting the camera-ready version report, the leaderboard hasn't marked which system(s) is/are unconstrained

the numbers and Latin words in our final submission of zhja to full width characters. For example, "2008" was converted to "２００８"[7]. Our contrastive result, in which all the numbers and Latin words are composed by half width characters (and this is the only difference compared with the primary submission we made), was scored 34.8, gained an improvement of nearly 5 points. The contrastive result is generated by the same model we trained on the constrained dataset. All the results reported above is shown in table 4

[7]Whether a letter or a digit is styled in half width or full width doesn't change its meaning

6 Conclusion and Future Works

In this report, we demonstrate our work for the Chinese-Japanese and Japanese-Chinese open domain translation task. The system we submitted is a neural MT model based on Transformer architecture. During the experiments, many techniques, such as back-translation, ensemble, reranking are applied and are proved to be effective for the MT system. Parallel data extraction, noisy data filtering methods and character-based models are also experienced and discussed, although currently they are not integrated into our systems, there will be still a lot work on them to find out proper ways to optimize the procedure and models, or to prove their limitations.

References

2019. *How Transformer Revitalizes Character-based Neural Machine Translation: An Investigation on Japanese- Vietnamese Translation Systems*. Zenodo.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics.

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.

John Hutchins. 2007. Machine translation: A concise history. *Computer aided translation: Theory and practice*, 13(29-70):11.

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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of chinese representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

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

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

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

Rico Sennrich and Martin Volk. 2010. Mt-based sentence alignment for ocr-generated parallel texts.

Rico Sennrich and Martin Volk. 2011. Iterative, mt-based sentence alignment of parallel texts. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 175–182.

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

Dániel Varga, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages.

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

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *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 5700–5705.