JParaCrawl: A Large Scale Web-Based English-Japanese Parallel Corpus

Makoto Morishita, Jun Suzuki, and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation

2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan

`{makoto.morishita.gr, masaaki.nagata.et}@hco.ntt.co.jp`

`jun.suzuki@ecei.tohoku.ac.jp`

Abstract

Recent machine translation algorithms mainly rely on parallel corpora. However, since the availability of parallel corpora remains limited, only some resource-rich language pairs can benefit from them. We constructed a parallel corpus for English-Japanese, for which the amount of publicly available parallel corpora is still limited. We constructed the parallel corpus by broadly crawling the web and automatically aligning parallel sentences. Our collected corpus, called JParaCrawl, amassed over 8.7 million sentence pairs. We show how it includes a broader range of domains and how a neural machine translation model trained with it works as a good pre-trained model for fine-tuning specific domains. The pre-training and fine-tuning approaches achieved or surpassed performance comparable to model training from the initial state and reduced the training time. Additionally, we trained the model with an in-domain dataset and JParaCrawl to show how we achieved the best performance with them. JParaCrawl and the pre-trained models are freely available online for research purposes.

Keywords: parallel corpus, machine translation, English, Japanese

1. Introduction

Since current machine translation (MT) approaches are mainly data-driven, one key bottleneck has been the lack of parallel corpora. This problem continues with the recent neural machine translation (NMT) architecture. As the amount of training data increases, NMT performance improves (Sennrich and Zhang, 2019).

Our goal is to create large parallel corpora to/from Japanese. In our first attempt, we focused on the English-Japanese language pair. Currently, ASPEC is the largest publicly available English-Japanese parallel corpus (Nakazawa et al., 2016), which contains 3.0 million sentences for training. Unfortunately, this is relatively small compared to such resource-rich language pairs as French-English[1]. Also, available domains remain limited. We address this problem, which hinders the progress of English-Japanese translation research, by crawling the web to mine for English-Japanese parallel sentences.

Current NMT training requires a great deal of computational time, which complicates running experiments with few computational resources. We alleviate this problem by providing NMT models trained with our corpus. Since our web-based parallel corpus contains a broad range of domains, it might be used as a pre-trained model and fine-tuned with a domain-specific parallel corpus.

The following are the contributions of this paper:

- We released the largest freely available web-based English-Japanese parallel corpus: JParaCrawl.

- We also released NMT models trained with JParaCrawl for further fine-tuning.

The rest of the paper is organized as follows. In Section 2, we introduce previous work that tried to create parallel corpora by mining parallel texts. In Section 3, we show how we created JParaCrawl. We conducted experiments to show how effectively our corpus and pre-trained models worked with typical NMT training settings. These experimental settings and results are shown in Section 4. Finally, we conclude with a brief discussion of future work in Section 5. JParaCrawl and the NMT models pre-trained with it are freely available online[2] for research purposes[3].

2. Related Work

One typical kind of source for parallel texts is the documents of international organizations. Europarl (Koehn, 2005) is an example of the early success of creating a large parallel corpus by automatically aligning parallel texts from the proceedings of the European Parliament. The United Nations Parallel Corpus (Ziemski et al., 2016) is another similar example that was created from UN documents. These texts were translated by professionals, and aligning the documents is easy because they often have such meta-information as the identities of speakers, although their domains and language pairs are limited.

Another important source of parallel texts is the web. Uszkoreit et al. (2010) proposed a large-scale distributed system to mine parallel text from the web and books. Smith et al. (2013) proposed an algorithm that creates a parallel corpus by mining Common Crawl[4], which is a free web crawl archive. Schwenk et al. (2019) mined Wikipedia and created a parallel corpus of 1,620 language pairs. The web, which includes a broad range of domains and many language pairs, is rapidly and continually growing. Thus, it has huge potential as a source of parallel corpora, although identifying correct parallel sentences is difficult.

Our work was inspired by the recent success of the

---

[1] Currently the largest French-English parallel corpus is ParaCrawl v5, which contains 51.3 million training data.

[2] `http://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/`

[3] This research is based on JParaCrawl v1.0. During the reviewing process, we released v2.0, which contains more than 10 million sentence pairs.

[4] `https://commoncrawl.org/`

ParaCrawl[5] project, which is building parallel corpora by crawling the web. Their objective is to build parallel corpora to/from English for the 24 official languages of the European Union. They released an earlier version of the corpora, and this version is already quite large[6]. This early release has already been used for previous WMT shared translation tasks for some language pairs (Bojar et al., 2018; Barrault et al., 2019), and task participants reported that ParaCrawl significantly improved the translation performance when it was used with a careful corpus cleaning technique (Junczys-Dowmunt, 2018). We extend this work to mine English-Japanese parallel sentences.

3. JParaCrawl

To collect English-Japanese parallel sentences from the web, we took a similar approach to that used in the ParaCrawl project. Figure 1 shows how we mined parallel corpora from the web. First, we selected candidate domains to crawl that might contain English-Japanese parallel sentences from Common Crawl data (Section 3.1). Then, we crawled the candidate domains (Section 3.2). Finally, we aligned the parallel sentences from the crawled data and filtered out noisy sentence pairs (Section 3.3).

3.1. Crawling Domain Selection

One key to creating a parallel corpus from the web is to decide which websites to crawl (i.e., finding candidate domains that contain a large number of parallel sentences). To select candidate domains, we first identified the languages of all the Common Crawl text data by `CLD2`[7] and counted how much Japanese or English data each domain has. We used an `extractor`[8] provided by the ParaCrawl project for creating language statistics for each domain.

If the amount of each language is balanced, the website might contain good parallel sentences. Thus, we used the ratio between Japanese and English as a criterion to select the candidate domains. We ranked all websites based on their language ratios and listed the top 100,000 domains to crawl.

3.2. Crawling the Web

The next step was to crawl the candidate websites to mine for parallel sentences. Since the crawled data stored on Common Crawl may not contain entire websites or might be outdated, we recrawled the websites ourselves with `HTTrack`[9]. We stopped crawling a website if we could not crawl it in 24 hours. In this experiment, we focused on text data; future work will include other formats, such as PDFs. After crawling 100,000 domains, our crawled data exceeded 8.0 TB with gzip compression.

3.3. Bitext Alignment

Next, we describe how we mined parallel text from the crawled web data.

	# sentences	# words
Raw	27,724,570	594,013,165
Filtered	8,763,995	196,084,272

Table 1: Number of collected sentences and words on English side in JParaCrawl corpus

We crawled a large number of websites, but some were too small to mine for parallel sentences. Therefore, we filtered out those domains whose compressed archive size was less than 1 MB, and only 39,936 domains remained.

To align parallel sentences, we used the `Bitextor` toolkit[10] provided by the ParaCrawl project. Our experiment was based on version 7.0, and we fixed several components for Japanese sentences. To extract text from HTML files, we used `extractontent`[11], which was developed for Japanese text. We used `split-sentences.perl`[12] contained in the Moses toolkit (Koehn et al., 2007) to split a text into sentences. We fixed the script to deal with Japanese end-of-sentence tokens.

There are two primary approaches to align parallel text. One algorithm uses a bilingual lexicon to generate crude translations along with such other features as sentence length to find the best sentence pair (Varga et al., 2005). The other algorithm uses an external MT system to translate one language into the other and find a sentence pair that maximizes BLEU scores (Sennrich and Volk, 2011; Papineni et al., 2002). Since the latter approach needs more computational resources for external MT, we used the bilingual lexicon-based algorithm. We used the EDR English-Japanese dictionary as a bilingual lexicon (Miyoshi et al., 1996).

Table 1 shows the number of collected parallel sentences and words. After the bitext alignment process, we mined over 27 million parallel sentences. However, these collected sentences contained many noisy pairs. Therefore, we filtered the corpus with `Bicleaner`[13] (Sánchez-Cartagena et al., 2018). The filtering model was trained with our in-house English-Japanese parallel corpora. After removing sentence pairs whose scores were lower than 0.5, we retained around 8.7 million sentences.

As our initial corpus release, we open-filtered 8.7 million parallel sentences to the public. However, we still found noisy sentence pairs that should have been filtered out. Future work will improve our filtering algorithm.

4. Experiments

We carried out three experiments to show how our corpus effectively works with typical NMT training settings. Figure 2 shows an overview of the experiments. In Section 4.1, we trained an NMT model with JParaCrawl. Then, we pre-trained an NMT model with JParaCrawl and fine-tuned it

---

[5] https://paracrawl.eu/

[6] For example, the English-German sentence pairs in the v5.0 release already exceed 36 million.

[7] https://github.com/CLD2Owners/cld2

[8] https://github.com/paracrawl/extractor

[9] http://www.httrack.com/

[10] https://github.com/bitextor/bitextor

[11] https://github.com/yono/python-extractcontent

[12] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/ems/support/split-sentences.perl
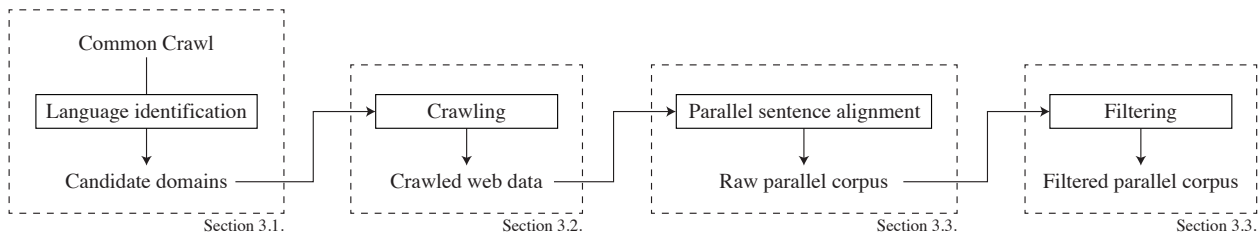
[13] https://github.com/bitextor/bicleaner

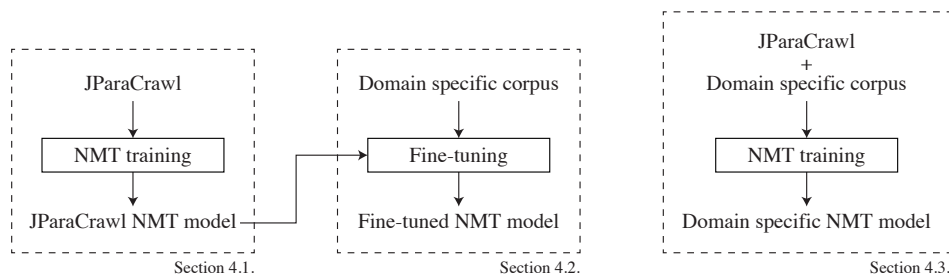Figure 1: Flowchart showing how we mined parallel sentences from web

Figure 2: Overview of experiments: We first trained an NMT model with JParaCrawl in Section 4.1 and fine-tuned it with a domain-specific corpus in Section 4.2. We also trained a model for a specific domain without fine-tuning in Section 4.3.

Data	# sentences	# words
ASPEC	1,812	39,573
JESC	2,000	13,617
KFTT	1,160	22,063
IWSLT	1,194	20,367

Table 2: Number of sentences and words on English side in test sets

Data	# sentences	# words
ASPEC	3,008,500	68,929,413
JESC	2,797,388	19,339,040
KFTT	440,288	9,737,715
IWSLT	223,108	3,877,868

Table 3: Number of sentences and words on English side in training sets. The original version of ASPEC contains 3.0 million sentences, but we used only the first 2.0 million sentences for training based on previous work (Neubig, 2014).

with an existing corpus in Section 4.2. Finally, we trained a model for a specific domain with JParaCrawl and other corpora without fine-tuning in Section 4.3. In the following experiments, we used a filtered JParaCrawl corpus that contained 8.7 million parallel sentences.

4.1. Training NMT with JParaCrawl

In this section, we trained NMT models with JParaCrawl and tested the models on several test sets to see how our corpus covers a broader range of domains.

4.1.1. Experimental Settings

Data To see how our corpus covers a broader range of domains, we used four test sets: scientific paper excerpts (ASPEC, Nakazawa et al. (2016)), movie subtitles (JESC, Pryzant et al. (2017)), texts on Wikipedia articles related to Kyoto (KFTT, Neubig (2011)), and TED talks (tst2015, provided for the IWSLT 2017 evaluation campaign, Cettolo et al. (2017), Cettolo et al. (2012)). Table 2 shows the details of the test sets. During training, we used the ASPEC dev set as a validation set.

We preprocessed the data with sentencepiece (Kudo and Richardson, 2018) to split the sentences into subwords. We set the vocabulary size to 32,000 and removed sentences whose length exceeded 250 subwords from the training data. Since JParaCrawl was NFKC-normalized, we also normalized the test sets.

For comparison, we trained NMT models with domain-specific bitexts. Table 3 shows the number of sentences and words in the domain-specific training sets. Since the sentences in ASPEC are ordered by their alignment confidence scores, the former sentences tended to be clean and the latter might contain noisy sentence pairs. Based on previous work (Neubig, 2014), we used only the first 2.0 million sentences for training, although the original ASPEC corpus contained 3.0 million sentences.

NMT Models We trained an NMT model with fairseq (Ott et al., 2019). Our model was based on Transformer (Vaswani et al., 2017). We trained three models for each direction by varying the hyper-parameters: small, base, and big settings. The base and big settings are based on Vaswani et al. (2017). For the base settings, we used an encoder/decoder with six layers. We set their embedding size to 512 and their feed-forward embedding size to 2048. We used eight attention heads for both the encoder and the decoder. For the big settings, we changed their embedding size to 1024 and their feed-forward embedding size to 4096. We also used 16 attention heads for both the encoder and the decoder. For the small settings, we used the same set-

tings as for the base settings, except we changed the number of heads to four. For all the settings, we used dropout with a probability of 0.3 (Srivastava et al., 2014). We used big settings for ASPEC and JESC, base settings for KFTT, and small settings for IWSLT.

As an optimizer, we used Adam with $\alpha = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.98$. We used a root-square decay learning rate schedule with a linear warmup of 4000 steps (Vaswani et al., 2017). We clipped gradients to avoid exceeding their norm 1.0 to stabilize the training (Pascanu et al., 2013). For the base and small settings, each mini-batch contained about 5,000 tokens (subwords), and we accumulated the gradients of 64 mini-batches for updates (Ott et al., 2018). For the big settings, we set the mini-batch size to 2,000 tokens and accumulated 160 mini-batches for updates. We trained the model with 24,000 iterations, saved the model parameters every 200 iterations, and averaged the last eight models. To achieve maximum performance with the latest GPUs, we used mixed-precision training (Micikevicius et al., 2018). When decoding, we used a beam search with a size of six and length normalization by dividing the scores by their lengths.

We slightly changed the settings for the in-domain baseline training because some of the above settings are inappropriate for smaller datasets. For IWSLT, we accumulated 16 mini-batches per update instead of 64. Since we confirmed that the model had already converged based on the validation loss, we stopped training at 20,000 iterations for all the in-domain baselines.

Evaluation To evaluate the performance, we calculated the BLEU scores (Papineni et al., 2002) with `sacreBLEU` (Post, 2018). Since `sacreBLEU` does not internally tokenize Japanese text, we tokenized both the hypothesis and reference texts using MeCab[14] with an IPA dictionary when evaluating the English-Japanese translations.

4.1.2. Experimental Results and Analysis

Table 4 shows the BLEU scores of the in-domain and JParaCrawl NMT models (see in-domain and JParaCrawl columns). Since JParaCrawl is not supposed to contain a specific domain, its BLEU scores were lower than those of the model trained with the in-domain corpus. However, we expect that these in-domain models only focus on a specific domain and do not work well with out-of-domain data. To test this, we created four out-of-domain test sets by selecting the first 1,000 sentences from each domain (ASPEC, JESC, KFTT, IWSLT) except for one set of in-domain data, resulting in 3,000 sentences for each test set, and measured the BLEU scores. Table 5 shows the BLEU scores of the out-of-domain test set. As we expected, the models trained with domain specific corpora did not perform well with the out-of-domain test sets; the JParaCrawl model achieved better results. This means that JParaCrawl contained a broader range of domains. Thus, a model trained with it might work well as a pre-trained model for fine-tuning with the domain-specific corpora, which we discuss in the following section.

4.2. Fine-Tuning with Pre-Trained Models

Next, we consider a situation in which we want a domain-specific NMT model with low computational cost. We fine-tuned the JParaCrawl pre-trained NMT model with a specific domain corpus for domain adaptation. Such a situation frequently occurs, especially in practical use, since training NMT from the initial state requires huge computational resources. Below, we address whether we can achieve a comparable performance while reducing the training time.

4.2.1. Experimental Settings

The domain-specific corpus we used for fine-tuning is the same as that described in Section 4.1.1. Table 3 shows the corpus statistics.

We started the training from the last saved model trained with JParaCrawl, as described in Section 4.1. Then we further trained the model for 2,000 iterations with the domain-specific corpus. For the small settings, we changed the number of mini-batches for updates to 16, since the IWSLT corpus is too small to accumulate a large number of mini-batches. We kept the other settings identical to those described in Section 4.1.1. We trained the models on eight NVIDIA RTX 2080 Ti GPUs. For evaluation, we measured the BLEU scores as well as the training time.

4.2.2. Experimental Results and Analysis

Table 4 shows the BLEU scores of the fine-tuned models (fine-tuning columns). Compared to a model trained just with JParaCrawl, a significant performance gain can be seen from the fine-tuning for all the settings. The AS-PEC and JESC experiments show that our fine-tuned models achieved almost comparable performance to the model trained with the in-domain data or even surpassed the AS-PEC English-Japanese model. Our fine-tuned models also surpassed the KFTT and IWSLT settings for both directions. ASPEC and JESC are already large enough to train solely with the corpus, although smaller corpora such as KFTT and IWSLT have room for improvement by our fine-tuning approach.

Table 6 shows the time required to train a model. For the fine-tuning experiments, we did not include the time for pre-training the model with JParaCrawl. Our fine-tuning approach drastically reduced the training time compared to training from the initial state. These results showed a domain-specific NMT model can be trained in a few hours starting from the pre-trained models with a parallel corpus of the domain.

As shown in this section, the JParaCrawl pre-trained model reduced the training time while maintaining (or boosting) the performance. This confirms that JParaCrawl can be useful as a pre-training corpus. We freely provided online the pre-trained models that we used for the experiments[15].

4.3. Training NMT for a Specific Domain with JParaCrawl

In the previous section, we discussed the pre-training and fine-tuning approaches to train domain-specific NMT models with reduced computational time. Next, we focus on the

---

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

[15] http://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/

Test data	English-to-Japanese			Japanese-to-English		
	In-domain	JParaCrawl	Fine-tuning	In-domain	JParaCrawl	Fine-tuning
ASPEC	44.3	24.7 (−19.6)	43.5 (−0.8)	28.7	18.3 (−10.4)	29.3 (+0.6)
JESC	14.5	6.6 (−7.9)	13.6 (−0.9)	17.8	7.0 (−10.8)	17.3 (−0.5)
KFTT	31.8	17.1 (−14.7)	33.0 (+1.2)	23.4	13.7 (−9.7)	25.4 (+2.0)
IWSLT	11.1	11.5 (+0.4)	14.2 (+3.1)	13.7	11.0 (−2.7)	17.2 (+3.5)

Table 4: BLEU scores of trained NMT models: Differences between in-domain scores are shown in brackets. See Section 4.1 for in-domain and JParaCrawl columns. For fine-tuning column, see Section 4.2.

Test excluded	English-to-Japanese		Japanese-to-English	
	Particular dataset only	JParaCrawl	Particular dataset only	JParaCrawl
ASPEC	7.9	14.6 (+6.7)	5.7	13.8 (+8.1)
JESC	5.4	20.1 (+14.7)	8.6	16.5 (+7.9)
KFTT	4.6	16.8 (+12.2)	5.7	14.7 (+9.0)
IWSLT	5.0	18.5 (+13.5)	3.7	14.6 (+10.9)

Table 5: BLEU scores of out-of-domain test sets

Data	Model	Without pre-training	Fine-tuning
ASPEC	big	13.22	1.60
JESC	big	14.19	1.69
KFTT	base	5.65	0.59
IWSLT	small	0.44	0.20
JParaCrawl	big	19.37	—
JParaCrawl	base	7.36	—
JParaCrawl	small	6.83	—

Table 6: Time required (hrs) to train NMT model for English-to-Japanese

Data	En-Ja	Ja-En
ASPEC only	44.3	28.7
JParaCrawl only	24.7	18.3
JParaCrawl → ASPEC fine-tune	43.5	29.3
JParaCrawl + ASPEC oversample	44.3	29.8

Table 7: BLEU scores on ASPEC test set

For both directions, the oversampled model outperformed the fine-tuning approach. In particular, in the Japanese-to-English experiments our model surpassed the model trained only with ASPEC.

These results show that adding JParaCrawl to the existing corpus improved the performance of a specific domain. To achieve the best translation performance, training should start from the initial state, though the computational time required for this is much greater than that for the fine-tuning approach.

4.4. Translation Example

To identify the strengths and weaknesses of the models, we compared the translations of the models trained for the previous experiments. Table 8 shows example translations from a domain of scientific paper excerpts.

Our results show that the JParaCrawl model sometimes made mistakes when translating domain-specific words. This example includes "The large eddy simulation," which is an uncommon, domain-specific expression. The model trained with the in-domain corpus correctly translated it; however, the model trained with JParaCrawl only mistranslated it as "large-scale." The output of the JParaCrawl model is somewhat understandable, but it does not work as accurately as the one trained with an in-domain corpus, in particular when the input is very domain-specific.

However, after fine-tuning with the in-domain corpus, the model can translate these domain-specific words correctly. This is almost the same as the model trained with

performance on a specific domain, scientific paper excerpts, and ignore the computational time to train the model. We compare a model trained from the initial state with the existing corpus or with JParaCrawl to achieve the best performance.

4.3.1. Experimental Settings

We trained a model for translating scientific domains with ASPEC and with JParaCrawl. Since JParaCrawl is much larger than ASPEC, we oversampled the latter three times and concatenated it to the former, resulting in 14.76 million sentences in the training data.

As an NMT model, we used Transformer with big settings, as described in Section 4.1.1. All hyper-parameters and training/evaluation procedures were identical to those in Section 4.1.1, except we changed the number of iterations to 25,000 based on the validation perplexity.

4.3.2. Experimental Results and Analysis

Table 7 shows the experimental results on the ASPEC corpus. The first and second rows correspond to Section 4.1, and the third row corresponds to Section 4.2. The last row shows the result trained with oversampled ASPEC and JParaCrawl, as described above.

Source	角柱を過ぎる粘性流体の乱流をラージエディシミュレーションし，フィルタ幅と数値粘性の影響を調べた。
Reference	The large eddy simulation of a turbulent flow of a viscous fluid passing through a square column was conducted, and the effects of the filter width and numerical viscosity were examined.
In-domain only	Turbulent flow of a viscous fluid past a square cylinder was simulated by a large eddy simulation to examine the effects of filter width and numerical viscosity.
JParaCrawl only	We simulated large-scale turbulence of viscous fluids past the square column, and investigated the influence of film width and numerical viscosity.
JParaCrawl → ASPEC fine-tune	Turbulent flow of viscous fluid past a square cylinder was simulated by large eddy simulation, and the effects of filter width and numerical viscosity were examined.
JParaCrawl + ASPEC (without pre-training)	Turbulent flow of viscous fluid past a square cylinder is simulated by large eddy simulation, and the effects of filter width and numerical viscosity are investigated.

Table 8: Example translations of trained models: The translation domain is scientific paper excerpts.

JParaCrawl + ASPEC (without pre-training), showing that the pre-training and fine-tuning approach could improve the translation accuracy while only requiring small computational time.

5. Conclusion

We introduced JParaCrawl, a large web-based English-Japanese parallel corpus. It was made by crawling the web and finding English-Japanese bitexts. After filtering out possibly noisy sentences, we retained around 8.7 million parallel sentences, which we publicly released as the corpus.

Our experiments showed how JParaCrawl contains a broader range of domains and can be used for general purposes. We also drastically reduced the training time by fine-tuning the JParaCrawl pre-trained NMT models and in doing so maintained or even boosted the performance. Finally, we showed that JParaCrawl also improved the performance of a specific domain when training a model with an existing corpus from an initial state.

In future work, we will crawl more websites and make the dataset larger. We also plan to improve the bitext aligner and cleaner, especially for Japanese. We only focused on English-Japanese in the initial release, but we hope to eventually add more language pairs to/from Japanese.

6. Acknowledgements

We gratefully acknowledge the ParaCrawl project for releasing the software and useful discussions. We thank the three anonymous reviewers for their useful comments. We would also like to thank Hisashi Itoh and Takumi Asai for their technical support.

7. Bibliographical References

Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. (2019). Findings of the 2019 conference on machine translation (WMT19). In Proceedings of the 4th Conference on Machine Translation (WMT), pages 1–61.

Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., and Monz, C. (2018). Findings of the 2018 conference on machine translation (WMT18). In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 272–303.

Cettolo, M., Girardi, C., and Federico, M. (2012). WIT3: web inventory of transcribed and translated talks. In Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT), pages 261–268.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. (2017). Overview of the IWSLT 2017 evaluation campaign. In Proceedings of the International Workshop on Spoken Language Translation (IWSLT), pages 2–14.

Junczys-Dowmunt, M. (2018). Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 425–430.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), pages 177–180.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In Proceedings of the Machine Translation Summit X, pages 79–86.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 66–71.

Micikevicius, P., Narang, S., Alben, J., Diamos, G. F., Elsen, E., García, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training. In Proceedings of the 6th International Conference on Learning Representations (ICLR).

Miyoshi, H., Sugiyama, K., Kobayashi, M., and Ogino, T. (1996). An overview of the EDR electronic dictionary and the current status of its utilization. In Proceedings of the 16th International Conference on Computational Linguistics (COLING), pages 1090–1093.

Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., and Isahara, H. (2016). ASPEC: Asian scientific paper excerpt corpus. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC).

Neubig, G. (2011). The Kyoto free translation task. http://www.phontron.com/kftt.

Neubig, G. (2014). Forest-to-string SMT for asian language translation: NAIST at WAT2014. In Proceedings of the 1st Workshop on Asian Translation (WAT), pages 20–25.

Ott, M., Edunov, S., Grangier, D., and Auli, M. (2018). Scaling neural machine translation. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 1–9.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 48–53.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 311–318.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning (ICML), volume 28, pages 1310–1318.

Post, M. (2018). A call for clarity in reporting BLEU scores. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 186–191.

Pryzant, R., Chung, Y., Jurafsky, D., and Britz, D. (2017). JESC: Japanese-English Subtitle Corpus. arXiv preprint arXiv:1710.10639.

Sánchez-Cartagena, V. M., Bañón, M., Ortiz-Rojas, S., and Ramírez-Sánchez, G. (2018). Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 955–962.

Schwenk, H., Chaudhary, V., Sun, S., Gong, H., and Guzmán, F. (2019). WikiMatrix: Mining 135m parallel sentences in 1620 language pairs from Wikipedia. arXiv preprint arXiv:1907.05791.

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

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pages 211–221.

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., and Lopez, A. (2013). Dirt cheap web-scale parallel text from the common crawl. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), pages 1374–1383.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958.

Uszkoreit, J., Ponte, J. M., Popat, A. C., and Dubiner, M. (2010). Large scale parallel document mining for machine translation. In Proceedings of the 23rd International Conference on Computational Linguistics (COLING), pages 1101–1109.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., and Trón, V. (2005). Parallel corpora for medium density languages. In Proceedings of the Recent Advances in Natural Language Processing (RANLP), pages 590–596.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), pages 6000–6010.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. (2016). The united nations parallel corpus v1.0. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC), pages 3530–3534.