Octanove Labs' Japanese-Chinese Open Domain Translation System

Masato Hagiwara
Octanove Labs LLC
Seattle, WA
masato@octanove.com

Abstract

This paper describes Octanove Labs' submission to the IWSLT 2020 open domain translation challenge. In order to build a high-quality Japanese-Chinese neural machine translation (NMT) system, we use a combination of 1) parallel corpus filtering and 2) back translation. We have shown that, by using heuristic rules and learned classifiers, the size of the parallel data can be reduced by 70% to 90% without much impact on the final MT performance. We have also shown that including the artificially generated parallel data through back-translation further boosts the metric by 17% to 27%, while self-training contributes little. Aside from a small number of parallel sentences annotated for filtering, no external resources have been used to build our system.

1 Introduction

Building a robust, open domain machine translation (MT) system for non-English, Asian languages remains a challenge since many MT research efforts have focused mainly on European languages (such as English and German) and/or on particular domains (such as news). This is especially the case when there is lack of high-quality, human-curated parallel corpora and one needs to bootstrap an MT system from noisy parallel data crawled from the Web.

This is the exact setting of the IWSLT 2020 open domain translation challenge (Ansari et al., 2020), where the organizers provide large, noisy parallel datasets crawled from the Web and the participants build open-domain machine translation systems between Japanese (JA) and Chinese (ZH). The participants are also encouraged to only use the provided datasets to train their models. Therefore, the key to building high-quality MT systems seems to be in how to filter and make the most of the provided, noisy datasets.

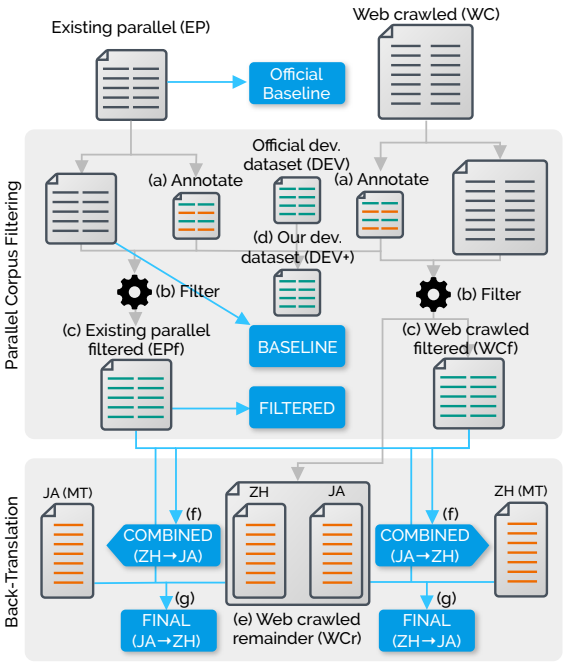

Figure 1: Overview of the data/training pipeline

Based on these insights, we used a combination of 1) parallel corpus filtering (Koehn et al., 2018, 2019) and 2) back-translation (Sennrich et al., 2016; Edunov et al., 2018) techniques as our main strategy. For 1), we showed that we can reduce the size of the parallel corpora by 70% to even 90% without much impact on the final MT performance. As demonstrated in (Chen et al., 2019), we also verified that artificially generated parallel data through back-translation can further help improve the performance by 17% to 27% depending on the direction. We used the vanilla Transformer (Vaswani et al., 2017) as our NMT model.

In the following sections, we describe the data and training pipeline for building our NMT system. We start with the two datasets provided by the shared task organizers—the existing parallel (EP)

dataset that includes public, parallel sentences, as well as the Web crawled (WC) dataset created by crawling, aligning, and filtering JA-ZH parallel sentences from the Web. Aside from a small number of parallel sentences annotated for filtering, no external resources besides these two have been used to build our NMT system. The entire data and training pipeline is illustrated in Figure 1.

2 Parallel Corpus Filtering

Our data processing pipeline consists of two main strategies–parallel corpus filtering and back-translation, which are shown as two main blocks in Figure 1. This section describes the first.

2.1 Sentence Pair Quality Analysis

The first observation we make is that many of the sentence pairs, even from the existing parallel (EP) dataset, are not high quality. In order to investigate the quality of the datasets, we first extracted roughly 1,000 sentence pairs from each dataset ((a) in Figure 1) and had two fluent speakers of both languages annotate each pair with a label indicating whether the pair is an accurate translation of each other, and if not, the reason why. We used the following tags to indicate the reasons:

- INVALID: text is garbled or contains few natural language words
- MT: text is suspected to be generated by MT. We made sure at least one native speaker of each language double checks this label.
- MISSING: information is missing from either side
- MISALIGNED: information is missing from both sides
- NOT TRANSLATED: both sides are identical except minor variations (e.g., simplified vs traditional Chinese)
- THIRD LANGUAGE: text is written in a language that is neither Japanese nor Chinese

Table 1 shows the breakdown of the labels and reasons annotated to sentence pairs, both for the existing parallel (EP) and the Web crawled (WC) datasets. Only 38% and 29% of the sentence pairs were deemed suitable for EP and WC, respectively. The most common error was MISALIGNED, meaning the sentences contain similar information but have some degree of mismatch that disqualifies the pair as a quality translation of each other. This can happen when the both segments are crawled from the same source (e.g., a webpage) but mis-aligned due to the way the text is segmented.

Label	Reason	EP	WC
NG	BOTH INVALID	0	2
	BOTH MT	0	92
	BOTH ZH	0	60
	JA INVALID	1	1
	JA MISSING	149	33
	JA MT	3	49
	ZH INVALID	8	0
	ZH MISSING	80	21
	ZH MT	9	24
	MISALIGNED	366	371
	NOT TRANSLATED	2	50
	THIRD LANGUAGE	4	11
OK		380	287
Total		1002	1001

Table 1: Result of sentence pair quality analysis

2.2 Training Sentence Pair Classifiers

These results led us to decide to use heuristic rules and build learned classifiers to filter out low quality sentence pairs from both datasets, illustrated as (b) in Figure 1. There is a large body of research on parallel corpus filtering (Koehn et al., 2018, 2019). We used a combination of heuristics rules as well as classifiers learned from the annotated data mentioned above.

First, we filter out sentence pairs that violate any one of the following criteria:

- both sides are 512 or fewer Unicode characters.
- $L_{JA}/L_{ZH} < 9$ and $L_{ZH}/L_{JA} < 9$ where L_{JA} and L_{ZH} are the lengths of the Japanese and the Chinese side, respectively.
- the detected languages match the expected ones (Japanese and Simplified Chinese). We used a neural language detector NanigoNet[1] to automatically detect the language of text.

[1] https://github.com/mhagiwara/nanigonet

	EP	WC
Before	1,963,238	18,966,595
After	627,811	1,973,068

Table 2: Size of the datasets before and after filtering

Second, we trained a binary logistic regression classifier from the annotated sentence pairs mentioned above, and applied it to the rest of the dataset to filter out low-quality sentence pair candidates. The classifier uses only three features. We built one classifier per each dataset (EP and WC) only using the annotated portion of the dataset and applied to the rest.

- log length of the Japanese text (in Unicode characters)
- log length of the Chinese text (in Unicode characters)
- cosine similarity between the sentence embeddings computed using the Universal Sentence Encoder (USE) (Cer et al., 2018)

As a result, we were able to reduce datasets to 31.9% (EP) and 10.4% (WC) of their original size (Table 2). We call the resulting filtered datasets the existing parallel filtered (EPf) and the Web crawled filtered (WCf), respectively, as shown as (c) in Figure 1. We achieved this with little impact on the translation quality. See the experiment section for more details.

Finally, we note that the official development dataset (DEV), which is created from the JEC Basic Sentence Data[2], might not be the best choice for evaluating an open domain machine translation system. Due to the way the the dataset is created (by first mining "typical" Japanese sentence structures from a large text corpus, then by translating these sentences to Chinese), it may not be well suited to evaluate ZH-to-JA MT systems. We augmented this dataset by adding sentence pairs that were tagged "OK" in the annotation process. This increased the size of the development dataset from 5,304 to 5,970 pairs. All the subsequent experiments were validated using this dataset, which we call DEV+ hereafter ((d) in Figure 1).

[2] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JEC%20Basic%20Sentence%20Data

3 Back-Translation

One of the effective techniques, especially for low-resource settings, is the use of back-translation (Sennrich et al., 2016; Edunov et al., 2018). The idea is to first train a target-to-source MT system to translate a large, monolingual dataset in the target language into the source language, and add the resulting, artificial parallel dataset to existing ones and retrain a source-to-target MT system.

We decided to reuse the "leftover" from the filtering process, that is, the set of sentence pairs that deemed low-quality in the parallel corpus filtering phase described in the previous section. Specifically, after running sentence pairs through the set of heuristic rules described above, we break them into the source side (Japanese) and the target side (Chinese) and treat each as an independent monolingual corpus. We call this corpus the Web crawled remainder (WCr) dataset ((e) in Figure 1).

We then trained ZH-to-JA and JA-to-ZH NMT systems from a combination of EPf and WCf datasets and used the systems to generate artificial source sides for both directions ((f) in Figure 1). When generating artificial source sides, we used top-k sampling (versus beam search) based on the findings of Edunov et al. (2018). The final models, shown as (g) in Figure 1, were trained from the combination of EPf, WCf, as well as WCr and its machine translated version.

4 Experiments

4.1 Experimental Settings

We used the vanilla Transformer (Vaswani et al., 2017) as our neural MT model. All the experiments were conducted using the fairseq library (Ott et al., 2019) with half precision floating point (fp16). The training objective is the label smoothed cross entropy, which was optimized by the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.997$, and $\varepsilon = 1.0 \times 10^{-9}$. We ran each experiment for 40 epochs and chose the best checkpoint based on the development set loss. The beam width was 20.

We tokenized both Japanese and Chinese with the SentencePiece library (Kudo and Richardson, 2018) with a shared vocabulary. The translation quality was evaluated with the character 4-gram BLEU (Papineni et al., 2002).

At the test time, we resolved unknown words (which often arise when there are rare unknown characters on the source side) using word alignment

obtained by fast_align (Dyer et al., 2013)[3].

4.2 Hyperparameters

Before we experiment with parallel corpus filtering and back-translation, we ran random parameter search with the baseline dataset (EP) to find the optimal set of hyperparameters. The type and the range of hyperparameters we considered are as follows:

- Size of SentencePiece vocabulary: **10k**, 15k, 20k, 25k, 30k
- Frequency threshold for including tokens (both sides): **3**, 5, 10
- Number of encoder/decoder layers: 4, 5, **6**
- Embedding dimension: 256, 512, 768, **1024**
- Feedforward dimension: 256, 512, 1024, 2048, 4096, **8192**
- Number of attention heads: 1, 2, 4, 8, **16**
- Gradient clipping: 0.0, 10.0, **25.0**, 50.0
- Learning rate: 1e-6, 2.5e-6, 5e-6, 1e-5, 2.5e-5, 5e-5, **1e-4**, 2.5e-4, 5e-4
- Number of warmup steps: 2000, 4000, 8000, **16000**, 32000
- Dropout: **0.1**, 0.2, 0.3, 0.4, 0.5
- Weight decay: 0.0, 1.0e-4, **2.5e-4**, 5.0e-4
- Label smoothing: 0.1, **0.2**, 0.3
- Batch size in tokens: 2048, 4096, **6144**

We ran about 20 rounds of random parameter search and settled with the hyperparameter setting shown by the bold face in the list above. The final models were an ensemble of 6, 8, and 10-layer Transformers with all other hyperparameters being identical.

4.3 Results

Here is the list of all the models trained from different combinations of datasets:

[3] https://github.com/clab/fast_align

	JA→ZH	ZH→JA
Official baseline	20.03	27.03
BASELINE	24.02	27.68
FILTERED	23.15	27.11
COMBINED	26.19	29.68
BT w/ 500k pairs	28.39	30.68
BT w/ 1M pairs	28.47	31.29
BT w/ 2.6M pairs	29.24	32.66
BT w/ 13M pairs	30.67	33.46
FINAL	31.21	33.81

Table 3: BLEU scores for different models

- Official baseline: the baseline BLEU scores provided by the organizer. Note that the these scores are not comparable to other models below since the development set is different. We also note that the official baseline model is very similar to our model in terms of the neural architectures as well as the number of parameters.
- BASELINE: baseline model trained with EP
- FILTERED: same as BASELINE but trained with EPf
- COMBINED: model trained with EPf and WCf
- BT: model trained with EPf, WCf, and back-translated WCr with varying size
- FINAL: BT trained on the entire WCr with ensemble

Table 3 shows the BLEU scores of these models computed against the DEV+ dataset. By comparing BASELINE and FILTERED, you see that filtering had little impact on the final BLEU scores. By comparing COMBINED and BT with different sizes, you see that adding back-translation helped the performance—the larger the amount of back-translation, the larger the increase was. These results confirm that our strategy—parallel corpus filtering and back-translation—was effective.

5 Discussion

5.1 Negative Results

Finally, here we include the list of things we tried but didn't contribute to the improvement of the MT quality:

- Filtering by provenance: upon cursory review, we found that the quality of parallel sentences varies a lot by their provenance. We included the source of each pair as an extra set of indicator features, although doing so ended up removing too many pairs and hurt the final performance.

- Self-training (Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019): we also tried using forward-direction MT models to generate the target side from WCr. Including artificially generated parallel data this way didn't improve the final BLEU score.

- Beam search: when generating back-translation, using beam search instead of top-k sampling didn't improve the metrics as much.

- Normalizing to Simplified Chinese: we tried normalizing the Chinese side to the simlified script using the OpenCC toolkit[4] to ensure the consistency. We observed that doing so inadvertently normalized many traditional characters that should be preserved between Japanese and Chinese and didn't improve the final performance.

We note that increasing the size of the Transformer beyond 6 layers did not necessarily lead to improved quality, while ensembling multiple large models did. We also considered leveraging the unaligned version of the Web crawled dataset provided by the organizers, although the dataset contains a large amount of low-quality text that appears to be generated by templates (such as updates on currency exchange rates) and we believe it would add little value as an extra data source.

5.2 Use of External Data

Finally, we ran a follow-up experiment in order to explore the extent to which our model can be improved by adding external data. Specifically, we obtained parallel sentences from HiNative[5], a community-driven language learning QA service, by collecting Japanese-Chinese question-answer pairs in the form of "How do you say X in Japanese/Chinese?" Both the questions and the answers are written by the user community and the resulting dataset is fairly noisy. In addition to the heuristic rules, we trained a logistic regression classifier in a similar way to the ones described in Section 2, except that we trained only one classifier using a combined held-out data from both EP and WC. After filtering, the HiNative dataset has been reduced to around 80k sentence pairs, which we added to EP to explore its impact on the NMT performance.

	JA→ZH	ZH→JA
BASELINE	24.02	27.68
BASELINE w/ HiNative data	25.32	29.20

Table 4: BLEU scores for models with external data

As Table 4 shows, even though the amount of the added data is a fraction of the original size (80k vs 1.9M), BLEU scores improved by more than 5%. This result suggests that our filtering method is very effective in only retaining high-quality pairs and the newly added data from HiNative provides new perspectives and genres that were not covered by the existing parallel dataset.

As future work, we wish to explore other external datasets for Japanese-Chinese translation, namely, JParaCrawl (Morishita et al., 2019) and WikiMatrix (Schwenk et al., 2019).

6 Conclusion

This paper describes Octanove Labs' submission to the IWSLT 2020 open domain translation challenge. We combined parallel corpus filtering and back-translation to build a Japanese-Chinese open domain NMT system. Through a series of experiments, we verified that our filtering method is effective in preserving the translation accuracy while greatly reducing the size of parallel data required to train the NMT model. We also found that use of artificially generated parallel data from the remainder of the filtered corpus through back-translation improved the final performance of the system.

Acknowledgments

We would like to thank the team at Lang-8, Inc. for providing the HiNative data.

[4] https://opencc.byvoid.com/
[5] https://hinative.com/

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.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Peng-Jen Chen, Jiajun Shen, Matt Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English translation task submission. In *Proceedings of the 6th Workshop on Asian Translation*, pages 112–122.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

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.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72.

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.

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.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2019. JParaCrawl: A large scale web-based japanese-english parallel corpus. *arXiv preprint arXiv:1911.10668*.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. WikiMatrix: Mining 135m parallel sentences in 1620 language pairs from Wikipedia. *ArXiv*, abs/1907.05791.

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.

Nicola Ueffing. 2006. Using monolingual source-language data to improve mt performance. In *IWSLT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.