KYB General Machine Translation Systems for WMT22

Shivam Kalkar, Yoko Matsuzaki, and Ben Li

NRI Digital, Ltd,
{s-kalkar, y-matsuzaki, b4-li}@nri.co.jp

Abstract

We here describe our neural machine translation system for the general machine translation shared task in WMT 2022. Our systems are based on the Transformer (Vaswani et al., 2017) with base settings. We explore the high-efficiency model training strategies, aimed to train a model with high-accuracy by using a small model and a reasonable amount of data. We performed fine-tuning and ensembling with N-best ranking in English to/from Japanese directions. We found that fine-tuning by filtered JParaCrawl data set leads to better translations for both directions in English to/from Japanese models. In the English to Japanese direction model, ensembling and N-best ranking of 10 different checkpoints improved translations. By comparing with another online translation service, we found that our model achieved a great translation quality.

1 Introduction

We participated in the Japanese to/from English translation for the general machine translation shared task of WMT 2022. Japanese ←→English is one of the challenging language pairs for machine translation since their differences are large in both vocabulary and grammatical structure. Recent advances in neural machine translation models have greatly promoted the development of the community. The transformer is the current key model and most recent participants are using a big-setting transformer model to improve the quality of translations. However, developing a more efficient model is also important. We here use a smaller model and limited computation resources to pursue high-quality translation models.

Our systems are based on the Transformer model with base settings, and the models are trained on the parallel corpus of Japanese and English (Morishita et al., 2019). We compared the quality of translations by using fine-tuning with several datasets. Also, we tested several different hyperparameters of the training to find suitable values for the task. After the fine-tuning, we tried to perform ensembling of multiple results from the model to earn a better-quality translation in the English to/from Japanese model. Here we describe the details of our systems.

2 Data selection and preprocessing

We select a suitable parallel corpus for model fine-tuning. We compare WMT provided dataset (which contained 7 different sources including the JParaCrawl dataset), KFTT (Kyoto Free Translation Task data set, Neubig, 2011), the JParaCrawl dataset (ver 2) and so on. We performed fine-tuning for these datasets and found that the model trained on the JParaCrawl dataset achieved better performance. We used a test data set made from WMT provided data and compare model performances by BLEU score. The score of the no fine-tuned model was 37.21, KFTT fine-tuned model was 14.87 and JParaCrawl fine-tuned model was 44.09. Therefore, we decided to use JParaCrawl as our fine-tuning dataset finally. We also consider that JParaCrawl has a reasonable amount of data for our high-efficiency training strategies.

Before we use the dataset, we check the corpus data to clean up. The JParaCrawl dataset contains over 10 million sentence pairs which were constructed by broadly crawling the web and automatically aligning. Therefore, there were noise and low-quality translations. We filtered low quality translation pairs and made a better translation dataset for fine-tuning. We also find that there were some contaminations of non-Japanese languages (e.g., Korean, Chinese) in the Japanese data. We also remove these pairs from the dataset.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 290 - 294

3 Tokenization

We perform the tokenization procedure using the SentencePiece toolkit[1] which provides us with a segmented sentence as tokens. In Japanese and some other languages like Chinese, words were not separated by spaces, therefore, tokenization needs to detect divided positions to separate each token. For Japanese, tokenization can be performed by a lattice-based tokenizer like MeCab[2]. A lattice-based tokenizer performs tokenization based on a dictionary and if the contents of the dictionary cover whole words in data, it provides highly accurate tokenization. However, in the development of machine translation using Neural Network mechanisms, more efficient tokenization methods like Byte-Pair-Encoding (BPE) were proposed (Sennrich et al. 2016c). SentencePiece was developed based on these methods and provides more efficient tokenization for the NMT (Kudo and Richardson, 2018).

SentencePiece is especially effective for languages not using spaces to separate words, has agglutinating morphology, and contains many compound words. Using SentencePiece helps extract subwords within compound words and create a more robust tokenizer. SentencePiece was used again to detokenize by removing the meta symbols from the output translation. For preprocessing the data, we have used the SentencePiece model, in which the vocabulary size is set to 32,000, and sentences whose length exceeded 250 subwords are removed from the training data.

4 Model Training

We train our NMT models with the fairseq[3] toolkit. The models are based on Transformer (Vaswani et al., 2017) with base settings. We use an encoder/decoder with six layers. We set their embedding size to 512, and their feed-forward embedding size to 2048. We use eight attention heads for both the encoder and the decoder. We used dropout with a probability of 0.3. 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. We clipped gradients to avoid exceeding their norm 1.0 to stabilize the training. For the base settings, each mini-batch contained about 5,000 tokens (subwords), and we accumulated the gradients of 64 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 use mixed-precision training. When decoding, we used a beam search with a size of six as the default condition and length normalization by dividing the scores by their lengths. We test other parameters of a beam search in the model of Japanese → English translations (size = 2, 3, 4, and 10) and found that size = 2 provide the best BLEU score for this task. We also compared models output by scaraBLEU (Post, 2018).

Our models are trained on the Google Cloud Platform's compute engine with 2-T4 GPUs. Model training generally took approximately 3.5 hours. We train our models in mixed precision to save costs without compromising on the accuracy.

Model condition	JParaCrawl data
Pretrained Model	39.4
Finetuned Model	45.1
Finetuned with ensemble	46.9*[1]

Table 1: BLEU Scores of English → Japanese direction, each column uses the same test dataset for three conditions.

*1 This result was not submitted due to our system's trouble.

5 Model Ensembling and N-Best Reranking for English → Japanese direction

After we fine-tuned our base model, we performed model ensembling with N-Best Reranking (Le et. al., 2021). For n-best reranking, we have created a script by referring to a script by Xu Song[4], bert-as-

[1] https://github.com/google/sentencepiece
[2] https://taku910.github.io/mecab/
[3] https://github.com/facebookresearch/fairseq
[4] https://github.com/xu-song/bert-as-language-model

Models	BLEU
Our model	43.9
DeepL	26.6

Table 2 Test result of our model and DeepL

a-language-model. We performed some changes in the scripts for its application to Japanese. For measuring the likelihood of the Japanese sentences produced by the NMT model, we have used the bert-Japanese model released by Yohei Kikuta[5].

For ensembling, the basic idea is to calculate the probability of tokens and perplexity of sentences produced by 10 different checkpoint files of a finetuned model. These 10 checkpoints will create 10 different translations for a given English sentence. Later, we are using bert-as-language-model to calculate the best sentence (the one with the lowest perplexity) score. We have used this sentence output for the submission. This method ensures the selected sentence has maximized fluency compared to other candidates.

6 Results and discussions

6.1 English → Japanese direction

We performed an experiment to compare ensembling effect (Table 1). In the experiment, we prepare training data from the JParaCrawl dataset to fine-tune our model and compare translations with/without ensembling. Based on the same training conditions, the score of the ensembling model is higher than the result of the model without ensembling.

To evaluate our translation quality, we compare the result with the online translation service (DeepL) by using a test dataset which created by the JParaCrawl dataset. The test data contains 1000 sentences that were not contained in the train data. The BLEU score of our model was higher than DeepL this means our fine-tuning procedure leads to better translation for the JParaCrawl dataset (Table 2).

We also check the translation result of the test set released by WMT2022. The dataset consists of 2037 English sentences and there were no reference sentences of Japanese. Therefore, we cannot calculate BLEU score here. Alternatively, we calculate perplexity[6] (PPL), by using bert-japanese model[5], which is explained in the model ensembling section. PPL is a metric of a language model and lower values mean better. We also check the translation quality by the human evaluation of a Japanese native speaker.

The average of the PPL of our model was lower than DeepL (Table 3). The result suggested that our small model established a high-fluently prediction rather than DeepL. In detail, for 941 cases in the test set with 2037 sentences, our PPL was lower than DeepL. We presented several examples of these cases in appendix examples 1 to 4. In these examples, the quality of translations for our model is also better than DeepL based on the confirmation of a native speaker. As a bad case, we list example-5 in the appendix. Although the translation of DeepL has better quality, however, the PPL score was higher than our model's output.

The results above (Table 2 and Table 3) suggested that we can establish a high-quality NMT model by small model and a reasonable amount of data, by using high-efficiency training strategies.

6.2 Japanese → English direction

For the Japanese to English direction, we perform finetuning with the Transformer model base setting on the JParaCrawl dataset. Table 4 shows our training results. For the final submission, we also performed post-processing to delete some extra punctuations that appeared in the translation results. We found that post-processing improved our results by 0.1 BLEU score.

Model condition	Our_PPL	DeepL_PPL	No. of cases
Average	41.59	51.75	2037
Average (our < DeepL)	21.86	90.61	941
Average (our > DeepL)	59.84	15.79	1096

Table 3 Comparison of our model and DeepL outputs by PPL

[5] https://github.com/yoheikikuta/bert-japanese

[6] https://huggingface.co/docs/transformers/perplexity

Model condition	JParaCrawl data
Pretrained Model	37.2
Finetuned Model	44.3

Table 4: BLEU Scores of Japanese to English direction.

7 Conclusions

We explored the high-efficiency model training strategies with a small model and a reasonable amount of data. Our systems are based on the transformer with a base setting. In our experiments, we found that data cleaning, model averaging, model ensembling, beam search, finetuning, parameter-tuning, and post-processing are useful techniques to train a high-quality model. Finally, we compared the translation results between our model and the online translation service, we found that our model achieved better translation quality. Our experiments suggested that exploring more efficient training strategies with a smaller model, a reasonable amount of data, and limited computational resources is promising to achieve a high-quality translation model.

References

Makoto Morishita, Jun Suzuki and Masaaki Nagata. 2019. JParaCrawl: A large scale web-based English-Japanese parallel corpus. *arXiv preprint* arXiv:1911.10668. https://doi.org/10.48550/arXiv.1911.10668

Graham Neubig. 2011. The Kyoto Free Translation Task, http://www.phontron.com/kftt

Rico Sennrich, Barry Haddow and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint* arXiv:1508.07909. https://doi.org/10.48550/arXiv.1508.07909

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. https://aclanthology.org/D18-2012

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30. https://doi.org/10.48550/arXiv.1706.03762

Giang Le, Shinka Mori, and Lane Schwartz. 2021. Illinois Japanese↔ English News Translation for WMT 2021. In *Proceedings of the Sixth Conference on Machine Translation*, pages 144–153, Online. Association for Computational Linguistics.. https://aclanthology.org/2021.wmt-1.11

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics. https://aclanthology.org/W18-6319

A Appendices

Example1

English: *[Not this time.]*

our_translation (Ja)	our_ppl
″今回はそうではありません。″	3.219
DeepL_translation (Ja)	DeepL_ppl
″今回は違う″	365.825

These two translations are similar, our model translation is a bit better.

Example2

English: *["How are we going to handle this?" he continued.]*

our_translation (Ja)	our_ppl
″どのように私達はこれを処理しようとしているか? 彼は続けた。″	20.209
DeepL_translation (Ja)	DeepL_ppl
″「そして、「この問題にどう対処していくのか?″	84.582

The quality of translations is better for our model based on the confirmation of a native speaker.

Example3

English: *[I have checked and this would be contactless so they would not be able to bring the item to your property I am afraid, I do apologise about this]*

our_translation (Ja)	our_ppl
私はチェックしました、そして、彼らは私が恐れているあなたの財産にアイテムを持って来ることができないので、これは非接触になるでしょう、私はこれについて謝ります。	13.010
DeepL_translation (Ja)	DeepL_ppl
″このような場合、私は、彼らがあなたの財産に項目をもたらすことができないだろう、私はこのことについて謝罪している非接触型であることを確認しました。″	48.198

The quality of translations is better for our model based on the confirmation of a native speaker.

Example4

English: *[If you have any questions, please feel free to contact us through the eBay emailing system.]*

our_translation (Ja)	our_ppl
″ご不明な点がございましたら、Eメールにてお気軽にご連絡ください。″	3.439
DeepL_translation (Ja)	DeepL_ppl
″質問があったら、eBay の emailing システムによって私達に連絡すること自由に感じて下さい。″	15.750

The quality of translations is better for our model based on the confirmation of a native speaker.

Example5

English: *[I've looked into it and I can see that your area is currently having a high volumes of order that is why they were assigning a rider for your order.]*

our_translation (Ja)	our_ppl
″私はそれを調べて、私は、あなたの地域が、現在、それらが、あなたの注文のためにリカーを割り当てていた理由である大量の注文を持っているのを見ることができます。″	18.24
DeepL_translation (Ja)	DeepL_ppl
″調べたところ、あなたの地域では現在注文が集中していて、そのためライダーが割り当てられることになったようです。″	85.75

Although our model PPL is lower, the quality of translations is better for DeepL based on the confirmation of a native speaker.