KYB General Machine Translation Systems for WMT23

Shivam Kalkar, Ben Li and Yoko Matsuzaki

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

Abstract

This paper describes our approach to constructing a neural machine translation system for the WMT 2023 general machine translation shared task. Our model is based on the Transformer architecture's base settings. We optimize system performance through various strategies. Enhancing our model's capabilities involves fine-tuning the pretrained model with an extended dataset. To further elevate translation quality, specialized pre- and post-processing techniques are deployed. Our central focus is on efficient model training, aiming for exceptional accuracy through the synergy of a compact model and curated data. We also performed ensembling augmented by N-best ranking, for both directions of English to Japanese and Japanese to English translation.

1 Introduction

In the context of the WMT 2023 general machine translation shared task for Japanese to/from English, we tackle the inherent challenges posed by the diverse linguistic structures of these languages. The transformative impact of the Transformer model on neural machine translation is undeniable. While current trends prioritize larger models and extensive datasets, our focus remains on achieving efficient translation with modest resources. This study underscores our use of a compact model and limited computational assets to enhance translation quality.

Built upon the Transformer model's base settings, our approach uses pre-trained models trained on Japanese-English parallel data (Morishita et al., 2019). A previous study involved fine-tuning on various datasets, yielding excellent translation within specific domains (Kalker et al., 2021). In this study, we refined our fine-tuning dataset and systematically tuned hyperparameters to optimize results. Post-fine-tuning, we harnessed model ensembling techniques to amalgamate multiple model outputs, leading to better translation quality. Our study highlights the specifics of our system configurations and methods, offering a concise overview of our strategies.

2 Data selection and Preprocessing

In this section, we elaborate on the process of creating our fine-tuning dataset for the Neural Machine Translation (NMT) system, with a focus on enhancing translation quality for the WMT competition. Our approach involved meticulous data selection and preprocessing to ensure the effectiveness of our system. We describe the details behind selecting the base dataset, incorporating additional parallel corpora, and performing data cleaning to curate a high-quality training dataset.

2.1 Base Dataset Selection

Our foundational dataset for training the initial NMT models is derived from the JParacrawl Version 3 dataset, which offers a diverse array of content spanning various domains. This choice was made due to its comprehensive coverage, which provides a strong starting point for training the base NMT models.

2.2 Augmenting the Dataset

Upon training our base models, we identified an opportunity to further enhance translation quality by incorporating additional datasets. To achieve this, we integrated parallel corpora obtained from sources recommended by the WMT competition organizers. This augmentation was aimed at increasing the diversity of the training data, which often contributes to improved translation accuracy.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 137–142
December 6–7, 2023.

2.3 Data Cleaning

Data cleaning played a pivotal role in refining the quality of our training dataset. During this stage, we implemented several key steps to ensure the integrity of the data:

Language Focus: Given our goal of improving Japanese-to-English translation, we focused on maintaining language homogeneity within the dataset. Therefore, non-Japanese languages, such as Korean and Chinese, as well as their corresponding English translations, were deleted from the dataset.

Sentence Length and Quality: To uphold the overall coherence and effectiveness of the NMT model, we eliminated sentences that were excessively short or of low quality. This step aimed to prevent the model from learning suboptimal translation patterns and to maintain a high standard of translation output. Furthermore, to maintain coherence, we curate our training data by excluding sentences longer than 250 subwords (see 3.1).

Translation Pair Quality: Translation Pair Quality: JParacrawl v3 provides a score for translations labeled as "Accuracy", so we removed sentences with lower scores from the dataset. The threshold for the score was set at 0.5 for training data and 0.75 for the validation dataset. This procedure was not applied to the other parallel corpora.

Normalize symbolic characters: Normalize symbolic characters: Especially for Japanese sentences, since the language has more variations of symbolic characters like 「」, 『』, and "" for quotation marks. We added pre-processing to normalize symbolic characters based on rules. We decided that emojis were not included in this process, as these characters are translated as they are. By adding these rule-based translations, the final BLEU score increased by +0.1.

2.4 Final Dataset Composition

The outcome of our data selection and preprocessing efforts yielded a curated dataset (22.2M). We divided the processed data into a training dataset and a test dataset (2.5M) to evaluate model quality. The training dataset was further divided into a train and a validation to perform fine-tuning on our NMT models.

We also developed a fusion dataset by combining the processed JParacrawl v3 training data with other parallel corpora that were provided by WMT 2023. The dataset finally contains 49.9M sentences (Figure 1, Table 1).

Figure 1 Dataset development

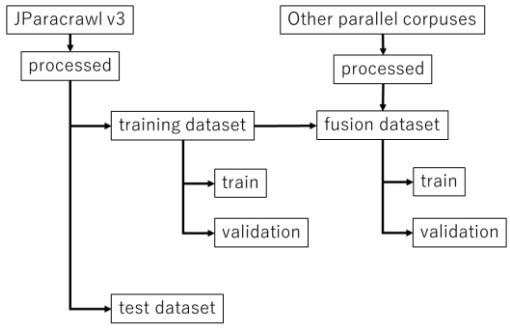


Table 1 Data selection summary

Dataset	Sentences
JParaCrawl Ver.3	25.7M
JParaCrawl Ver.3 (processed)	22.2M
Other Parallel Corpus	33.8M
Other Parallel Corpus (processed)	27.7M
Fusion Corpus (JParacrawlVer3 + Other Parallel)	49.9M

3 Tokenization

3.1 SentencePiece Toolkit for Tokenization and Detokenization

We use the SentencePiece toolkit (Kudo and Richardson, 2018) for tokenization. SentencePiece is suited for languages with complex linguistic structures and compound words. Its efficacy is pronounced in languages with ambiguous word boundaries, agglutinative morphology, and compound word usage. This enables the extraction of subword components from intricate terms, enhancing our tokenization precision. It can remove meta-symbols from translated output, ensuring fluidity and linguistic correctness in the final translations.

3.2 Customized Vocabulary

To bridge vocabulary disparities between our base model (JParacrawl Version 1) and our Fusion Corpus, we train a SentencePiece tokenizer. This

tokenizer aligns with our data's linguistic nuances, enhancing token accuracy. Our SentencePiece model employs a vocabulary size of 32,000 tokens.

4 Model Training

This section details the training process of our translation models using the fairseq toolkit. The selection and configuration of models, as well as the optimization parameters, are presented. Additionally, our model training strategy (Figure 2), including the utilization of advanced techniques such as mixed-precision training and beam search during decoding, is outlined.

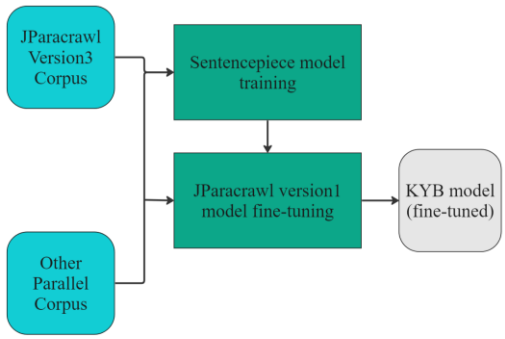
4.1 Model Selection and Configuration

From the array of available models, including MBART and JParacrawl, our evaluation led us to opt for Jparacrawl due to its favorable accuracy-performance trade-off. Jparacrawl models are underpinned by the Transformer architecture (Vaswani et al., 2017) with base settings. The encoder and decoder feature six layers each, embedding sizes of 512, and feed-forward embedding sizes of 2048. Eight attention heads are employed for both the encoder and decoder. Dropout with a probability of 0.3 is applied to enhance generalization. The Adam optimizer with $\alpha = 0.001$, $\beta 1 = 0.9$, and $\beta 2 = 0.98$ is utilized. A square root decay learning rate schedule with a linear warmup of 4000 steps is implemented. Gradient clipping maintains stability by ensuring gradients do not exceed a norm of 1.0. Mini-batches contain around 5,000 tokens, with gradient accumulation of 64 mini-batches per update. Training spans 24,000 iterations, with model parameter snapshots saved every 200 iterations. The final model is an average of the last eight snapshots. The use of mixed-precision training optimizes performance on modern GPUs.

4.2 Decoding Strategy

During decoding, various beam search sizes (2, 4, 6, 8) were employed to compare translation results. We also compared the best checkpoint and averaged checkpoint to obtain various translation results. In the training of the Ja to En model, we also compared the optimal trade-off between translation quality and computational efficiency by optimizing the training precision.

Figure 2 Model training system



4.3 Training Environment

Our model training is executed on Google Cloud Platform's compute engine equipped with 4-T4 GPUs. Mixed precision (float16) training takes approximately 12 hours and full precision (float32) training takes approximately 30 hours. Although the BLEU score is higher for full precision, the difference is not so large (+0.6, Table 2). So, we decided to use mixed precision for training the Fusion Corpus. We used the train-validation-test split ratio of 90:5:5.

Table 2 En-Ja models summary

Test Dataset	Model	Training Precision	BLEU score	change
JPC V3	JPC V1	-	38.0	0.0
JPC V3	KYB	Mixed (fp16)	45.3	+7.3
JPC V3	KYB	Full (fp32)	45.9	+7.9
Fusion	JPC V1	-	22.3	0.0
Fusion	KYB	Mixed (fp16)	29.8	+7.5

※JPC: Shorthand for JParacrawl
*The results used the best checkpoint with beam size 4. Test set was based on JPC V3 data.

Table 3 Ja-En models summary

Test Dataset	Model	Training Precision	BLEU score	change
JPC V3	JPC V1	-	19.2	
JPC V3	KYB	Full (fp32)	43.8	+24.6

*The results used the best checkpoint with beam size 4. Test set was based on JPC V3 data.

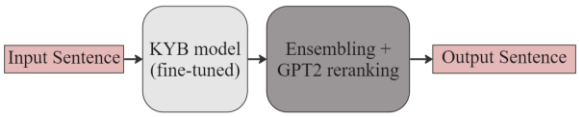
4.4 Performance Evaluation

To assess the efficacy of our trained models, we employ validation data from our dataset. The sacreBLEU metric (Post, 2018) is employed to calculate BLEU scores, offering a quantitative evaluation of translation quality. For testing, we used a set of 2.5M sentences from JParacrawl v3.

5 Model Ensembling with N-Based Reranking

In this section, we delve into the intricacies of our advanced model ensembling approach (Figure 3) coupled with N-Based Reranking, a technical strategy inspired by Le et al. (2021). Our objective is to optimize translation quality through a combination of models and a refined reranking mechanism.

Figure 3 Model inference system



5.1 Model Averaging via Ensembling

Our ensembling technique involves the aggregation of multiple trained model files. Through model averaging, we synthesize the insights and strengths of various models into a unified translation framework. This process not only enhances the stability of our translation outputs but also contributes to an overall improvement in translation quality. Moreover, we implemented checkpoint averaging by considering the last eight checkpoints to create an averaged model and employed in-training evaluation to identify the best checkpoint model.

5.2 N-Based Reranking Strategy

The crux of N-Based Reranking revolves around the calculation of token probabilities and sentence perplexities for translations generated by distinct checkpoint files of our fine-tuned model. We generate 4 alternative translations for each source sentence by using different beam search sizes for one checkpoint file. We compared 6 checkpoint files for the En-Ja side from three different trained models, which yielded 24 different translations for each source sentence. For the Ja-En side, we use two different checkpoints from trained models and use the previous study's model to make 12 alternative translations for each source sentence. This multifold approach introduces diversity into our pool of translation candidates, a crucial aspect of refining translation quality.

To identify the optimal translation candidate among these alternatives, we employ a GPT-2 based ranker (Radford et al., 2019). The ranker computes perplexity for each alternative translation, then chooses the lowest perplexity score as the most proper translation. We submitted the best translation result from all the alternative translations.

6 Post-processing of translation

We found specific tendencies of mistranslation, such as adding double quotation marks, deleting part of the quotation marks, or repeating a specific word endlessly in our translation results. These phenomena are typical problems in machine translation tasks, so we added post-processing to reduce the mistranslations.

After the post-processing, we submitted our results. Our final submission was scored as shown in Table 4 in the automatic evaluation.

Table 4 Submission results

Submission	COMET	BLEU	chrF
Ja-En	76.6	17.6	43.9
En-Ja	80.8	17.8	27.7

7 Discussion and Future Work

7.1 Discussion

In this study, we participated in the general translation task to achieve better translation quality with a relatively compact model and dataset. We made two different datasets: one is based on JparaCrawl version 3 data (JPC V3), and the other included additional parallel corpora provided by WMT23 (Fusion Corpus). The results of our local test suggested that the JPC V3 fine-tuned model shows a better BLEU score than the Fusion Corpus fine-tuned model. However, we exercised caution in interpreting these scores, since the test dataset

contains only sentences from Jparacrawl v3 dataset. The model trained with same origin data might show the result of overfitting to the domain of JPC3 dataset, even the test and train dataset contains different sentences. We then tried to ensemble these different models to achieve more robust translation results. Additionally, we generated alternative translations using different inference parameters for each model and then chose the most proper translation by using PPL.

We use GPT-2 to calculate PPL in this study since the model's knowledge of language is somewhat better than BERT's (see Appendix for a comparison of PPLs). Using a pre-trained large model is obviously effective when computational resources are limited. We use PPL to select a better result from the alternative translations; however, PPL is a relative metric of how fluent the sentence is or how acceptable the sentence order is for the model, so it is not a direct metric of translation quality (Kalkar et al., 2022; Wang et al., 2022). According to that, we recognize that the method that relies on PPL still has some limitations in improving the quality of translations.

Additionally, we added some post-processing to reduce specific mistranslations through the model. Although we carefully cleaned up noisy symbolic characters such as quotation marks from the training dataset, the model's output is still not reliable for translating symbolic characters properly. We performed some rule-based translations to modify symbolic characters.

7.2 Future Work

Back-Translation / Forward-Translation

To improve our translation pipeline, we explored the integration of back-translation as a potential enhancement. Back-translation involves using a trained model to translate from the target language back to the source language (forward-translation is vice versa), effectively creating a synthetic parallel dataset. While we attempt to do the back-translation, we need to consider the quality of the synthetic dataset, especially the variety of translations. When our synthetic dataset does not have enough variation in translation, the model can be easily overfit to the specific translation pattern.

To avoid that, we tried to use a common API for translation, like the Google API; however, this proved to be a very time-consuming task to obtain enough dataset for training (e.g., when one response takes 1 sec, it takes more than 55h to obtain 200,000 sentences). In this study, we attempted to perform back/forward translation, however, we were not able to obtain enough volume of content with reasonable quality. We would like to find a practical method to develop datasets with reasonable quality for back or forward translation in future work.

8 Conclusion

In this study, we embarked on an extensive exploration of high-efficiency model training strategies, leveraging limited computational resources alongside a streamlined model architecture rooted in the Transformer framework's base settings. Our investigation yielded crucial insights and techniques that converge to create a high-quality translation system. Through our experimentation, we identified data cleaning, model averaging, ensembling, beam search, finetuning, parameter-tuning, and post-processing as pivotal techniques, enhancing the quality of our compact model and modest dataset.

References

1. 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

2. Shivam Kalkar, Yoko Matsuzaki, and Ben Li. 2022. KYB General Machine Translation Systems for WMT22. https://aclanthology.org/2022.wmt-1.22/

3. 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

4. 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

5. 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

6. 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

7. Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

8. Yequan Wang, Jiawen Deng, Aixin Sun, and Xuying Meng. 2022. Perplexity from PLM Is Unreliable for Evaluating Text Quality. https://doi.org/10.48550/arXiv.2210.05892

Appendix-1: Perplexity comparison BERT vs GPT2 for Japanese

PPL metric: Lower PPL corresponds to better semantic quality.

Example1: BERT gives low score for bad sentences

Sentence	BERT	GPT2
返品は与えられたものではありません! [Returns are not allowed!]	15.9	26.3
NATO ストラップは、時計の下にループし、最後に追加のキーパーを通します。 [The NATO strap loops under the watch and finally passes through an additional keeper.]	9.7	45.6
この音は、マスターユニットによるセカンドロックです。 [This sound is of the second lock by the master unit.]	12.5	33.4

Example2: BERT gives high score for good sentences

Sentence	BERT	GPT2
国際郵便 - 日本郵便 [International Mail - Japan Post.]	49.6	1.9
大切なことは、毎晩 3 つのことを書き続けることです。 [The important thing is to keep writing three things every night.]	23.5	8.7
タングステン重合金は無毒で環境にも優しいため、子供や大人がタングステン重合金を扱ったり作業したりするのに安全です。 [Tungsten heavy alloys are non-toxic and environmentally friendly, making it safe for children and adults to handle or work with tungsten heavy alloys.]	78.2	6.8