SKIM at WMT 2023 General Translation Task

Keito Kudo[♢], **Takumi Ito**[♢], **Makoto Morishita**[♠], **Jun Suzuki**[♢]
[♢]Tohoku University [♠]NTT Communication Science Laboratories

Abstract

The SKIM team's submission used a standard procedure to build ensemble Transformer models, including base-model training, back-translation of base models for data augmentation, and retraining of several final models using back-translated training data. Each final model had its own architecture and configuration, including up to 10.5B parameters, and substituted self- and cross-sublayers in the decoder with a cross+self-attention sublayer (Peitz et al., 2019). We selected the best candidate from a large candidate pool, namely 70 translations generated from 13 distinct models for each sentence, using an MBR reranking method using COMET and COMET-QE (Fernandes et al., 2022). We also applied data augmentation and selection techniques to the training data of the Transformer models.

1 Introduction

This paper provides a system description of submissions by our team, called SKIM[1], at WMT-2023. We took part in English to Japanese (En→Ja) and Japanese to English (Ja→En) General Machine Translation tracks (Kocmi et al., 2023). We specifically participated in the constrained track, which places restrictions on the available data and pre-trained models.

The trial of this year's submissions is a reranking part. Our submission system consists of multiple translation models, followed by a reranking module (Kobayashi, 2018) based on COMET (Rei et al., 2022a) and COMET-QE (Rei et al., 2021). This reranking approach serves to identify and select high-quality translations from the hypothesis candidate set generated by multiple translation models. Among the Transformer-based translation models, we also incorporated a large Transformer model with 10.5B parameters. We also applied data augmentation techniques based on our previous year's system (Morishita et al., 2022b). We briefly describe the system overview, including the experimental results that could not be submitted.

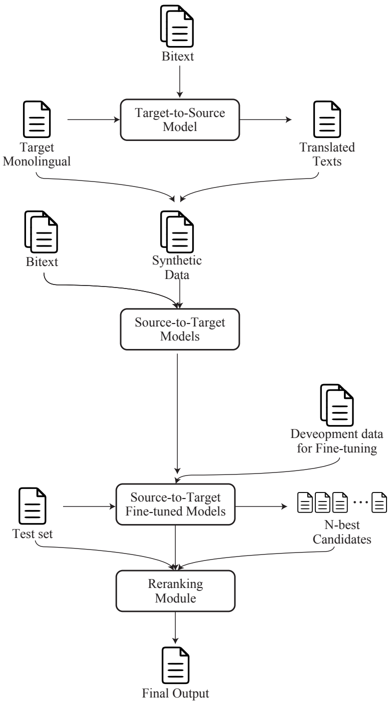

Figure 1: System overview.

2 System Overview

An overview of our submission system is shown in Figure 1. Following the development process used for last year's system (Morishita et al., 2022b), we used Transformer (Vaswani et al., 2017) as the model architecture and conducted pre-training and fine-tuning. In the pre-training phase, we used both a synthetic dataset created by back translation (Sennrich et al., 2016) and the provided bitext dataset. Here, we refer to the target-to-source translation model to generate this synthetic dataset as the initial translation model. Furthermore, we conducted fine-tuning on the translation models derived from pre-training using high-quality bitext

[1] The team name is an anagram of the first letters of the authors' last names.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 128–136
December 6–7, 2023. ©2023 Association for Computational Linguistics

Initial Translation Model	
Subword Size	32,000
Architecture	Transformer (big) with FFN size of 4,096
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3
Gradient Clip	1.0
Batch Size	1,280,000 tokens
Number of Updates	50,000 steps
Averaging	Save a checkpoint every 200 steps and average the last eight
Implementation	`fairseq` (Ott et al., 2019)
Pre-training Configuration	
Subword Size	64,000
Architecture	(See Table 4)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3 / 0.1
Gradient Clip	0.1 / 1.0
Batch Size	1,024,000 / 64,000 tokens
Max. Num. of Updates	60,000 / 100,000 (stoped at 64,000)
Averaging	Save a checkpoint every 2,000 steps and average the last ten
Implementation	`fairseq` (Ott et al., 2019)
Fine-tuning Configuration	
Subword Size	Identical to Pre-training Configuration
Architecture	(See Table 4)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Fixed
Warmup Steps	N/A
Max Learning Rate	0.00001
Dropout	0.3 / 0.1
Gradient Clip	1.0
Batch Size	16,000 / 14,400 tokens
Number of Updates	400 / 200
Averaging	Save a checkpoint every ten steps and average the last ten
Implementation	`fairseq` (Ott et al., 2019)

Table 1: List of hyper-parameters. We used the initial translation model for creating synthetic data, pre-training configuration to construct pre-training models described in Section 4.2, and fine-tuning configuration to construct models for submission. Note that we used slightly different settings for 10.5B models in a few parameters. We show their settings at the righthand side of the slash mark (/). We used several different model configurations for ensembling. See Table 4 for more details.

Corpus	w/o Filtering	w/Filtering
JParaCrawl v3.0	25.7 M	25.0 M
WikiMatrix	3.89 M	3.64 M
JESC	2.80 M	2.57 M
Wiki Titles v3	757 K	327 K
KFTT	440 K	371 K
TED Talks	242 K	224 K
NewsCommentary v18	3.8 K	3.7 K

Table 2: Number of sentence pairs in bitext corpus.

datasets (i.e., development data provided by the organizers). When developing last year's submission system, we found that fine-tuning with clean data enhanced translation quality more effectively than domain adaptation. Therefore, we used a similar fine-tuning approach for this year's submission system. By using these datasets, we trained multiple Transformer-based translation models with heterogeneous configurations. During the inference phase, we translated the source sentences using these translation models individually and selected the final translation results using a subsequent reranking process. As reranking, we tried two methods: one used COMET-QE and the other used COMET-MBR (Fernandes et al., 2022) extended to the outputs of multiple models.

3 Dataset Construction

3.1 Provided Data

Bitext Corpus We used all the provided bitext corpora: JParaCrawl v3.0 (Morishita et al., 2022a), News Commentary v18, Wiki Titles v3, WikiMatrix, Japanese-English Subtitle Corpus (JESC) (Pryzant et al., 2018), The Kyoto Free Translation Task (KFTT) Corpus (Neubig, 2011), and TED Talks (Cettolo et al., 2012). We filtered out the potentially noisy pairs using the straightforward parallel corpus filtering methods, just as we did with last year's system (Morishita et al., 2022b). Table 2 shows the size of each dataset with/without filtering. Compared to the previous year, the organizers updated the NewsCommentary, resulting in an increase of 1.8 K sentences.

Monolingual Corpus We also used the following provided monolingual data: News Crawl, News Commentary, and Common Crawl. We back-translated the monolingual sentences using a target-to-source model (i.e., an initial translation model) trained only with the provided bitext dataset, as described in Section 3.2, and used them as synthetic data (Sennrich et al., 2016).

	#sent. pairs	#subwords (JA)	#subwords (EN)
En→Ja	587 M	12.9 B	15.0 B
Ja→En	681 M	17.2 B	16.7 B

Table 3: Statistics of synthetic data used for pre-training.

3.2 Building Pre-Training Data

Synthetic Data Construction To augment the training data, we constructed synthetic data by applying the initial translation model trained with bitext to the monolingual data. As a preprocessing step, we truecased[2] both the bitext and monolingual data. We then tokenized the data into subwords using the `Sentencepiece` tool (Kudo and Richardson, 2018) with the unigram language model option.

We set the vocabulary size to 64,000, the same as the previous year's submission. To integrate insights from the method to create vocabulary for recent large-language models (Touvron et al., 2023), we activated the "byte_fallback" and "split_digits" options. Through preliminary experiments, we confirmed that activating these options leads to enhanced translation performance. As our initial translation model, we used the identical initial translation model we used for last year's submission system (Morishita et al., 2022b). The detailed hyperparameters are described in the initial translation model section of Table 1. Finally, we respectively translated 3.3 B (English) and 1.4B (Japanese) monolingual sentences.

Data Cleaning For both the provided bitext and synthetic data, we carried out cleaning based on a combination of sentence embeddings and hand-crafted rules.

For both the bitext and synthetic data, we removed the too-long sentences (>500 characters) and using the `langid`[3] toolkit, removed the sentences that were identified as not being written in English or Japanese.

For the synthetic data, we further applied a sentence embedding-based filtering approach. We took advantage of LaBSE (Feng et al., 2022) to embed the Japanese and English sentences into the same embedding space. We then scored and ranked the parallel sentence pairs based on the cosine similarity of their sentence embeddings. We subsequently filtered out the following items from the synthetic data:

- Duplicated sentence pairs
- Sentences with over 150 words[4] or single words with over 40 characters
- Sentences where the ratio between the word and the character count is > 12
- Sentences that contain invalid Unicode characters
- Sentence pairs where the source/target word ratio exceeds 4
- sentence pairs where the source/target length ratio exceeds 6
- sentence pairs where the source and target sentences are identical
- sentence pairs where the cosine similarity is greater than 0.96[5]

Finally, we respectively selected the top 587M and 681M (approximately) sentences, respectively, from the translated 1.4 B and 3.3 B monolingual sentences as the En→Ja and Ja→En synthetic data for the rank orders. Table 3 shows the statistics of the synthetic data used for our pre-training.

3.3 Fine-Tuning Data

As mentioned in Section 2, during the development of last year's submission system, we found that fine-tuning the model with clean data was more effective for improving translation quality than domain adaptation. Following this finding, we used the WMT'20 test set, WMT'20 development set, WMT'21 test set and WMT'22 test set as clean data for fine-tuning. The WMT'20 test and development sets were all used as clean data. However, for the WMT'21 and WMT'22 test sets, only the opposite language direction data were used (i.e., only Ja→En data were used as clean data for the En→Ja models) because these data were used for development and evaluation. The clean data included 9,002 sentences for En→Ja and 9,026 sentences for Ja→En.

4 Primary Translation Module

We trained several Transformer models for the reranking in the decoding phase. We describe the

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl

[3] https://github.com/saffsd/langid.py

[4] We tokenized the Japanese sentences using MeCab (Kudo, 2006) with the IPA dictionary. Note that this tokenization is for this cleaning purpose only.

[5] We found that sentence pairs with high cosine similarities can be noisy; for example, the source and target sentences are sometimes identical. Thus, we removed them from the training data.

details of the models in this section. Furthermore, alongside the newly trained models, we reused the primary translation models from the previous year's submission system (Morishita et al., 2022b).

4.1 Model Configuration

We independently trained models with heterogeneous model configurations. Our configuration has several notable characteristics: a cross+self-attention mechanism and a large number of parameters (i.e., 10.5B). In the following sections, we describe the details of the configurations.

Cross+Self-Attention Mechanism We introduced a cross+self-attention mechanism (Peitz et al., 2019) to the Transformer decoder. This mechanism was expected to reduce the model parameters and provide faster training while maintaining the translation performance. In this approach, we eliminated the decoder's cross-attention layer and unified the self-attention and cross-attention into a single attention layer. Specifically, the self-attention layer within the Transformer decoder simultaneously performs the cross-attention calculation by concatenating the output from the encoder's final layer to the query and key matrices.

Suppose Q, K, and V are the query, key, and value matrices, respectively; H_{enc} is the matrix form of concatenating all the output vectors of the encoder's final layer; W_q, W_k, W_v are the weight matrices for the query, key, and value, respectively; and d_k denotes the dimension of the key matrix. It is then formulated as follows:

$$\begin{aligned} &\text{Attention}(Q, K, V, H_{enc}) = \\ &\qquad \text{softmax}\left(\frac{Q_{concat}K_{concat}^{T}}{\sqrt{d_k}}\right)V' \\ &Q_{concat} = (Q \oplus H_{enc})W_q \\ &K_{concat} = (K \oplus H_{enc})W_k \\ &\quad V' = VW_v \end{aligned} \tag{1}$$

where $\oplus$ means concatenating two matrices in this equation.

Note that cross+self-attention, as well as standard self-attention, assume Q, K, and V to be identical matrices, namely, $Q = K = V = H_{dec}$, where H_{dec} is the matrix form of concatenating input vectors of the corresponding decoder layer.

10.5B Model As demonstrated in Kaplan et al. (2020), the performance of neural models improves as the number of parameters increases. Moreover, previous WMT shared tasks systems, such as Chen et al. (2020), achieved improvements in translation quality using model scaling. Following this insight, we attempted to scale up the translation model. Considering the constraints of GPU memory and training time, we finally configured the model size to be 10.5B parameters.

We also applied the position encoding methods used in last year's submission system (Morishita et al., 2022b). Namely, in the encoder, we employed relative position encoding (Shaw et al., 2018). In the decoder, we used SHAPE (Kiyono et al., 2021). We specified the maximum shift size of SHAPE to be 10.

Previous year's submission models We also incorporated the transformer models developed for the previous year's submission system as the primary translation module. We introduced the bottom-to-top (B2T) connection (Takase et al., 2023) to these models for training stability and relative position encodings (Shaw et al., 2018) to improve their generalization ability to unseen sentence lengths during training. For more details, please refer to (Morishita et al., 2022b).

4.2 Pre-Training

We trained each translation model shown in Table 4 with the filtered bitext and synthetic data described in Section 3.2. In this phase, we used the pre-training configuration shown in Table 1.

Following last year's submission system (Morishita et al., 2022b), the bitext was upsampled until it reached to a ratio of 1:1 with the synthetic data. Moreover, we used the tagged back-translation technique (Caswell et al., 2019) by adding a special token ⟨BT⟩ to the beginning of the source sentences in the synthetic data.

4.3 Fine-Tuning

The fine-tuning data are detailed in Section 3.3, and the hyperparameters utilized during training are as described in Table 1.

4.4 Ensemble

We ensembled the fine-tuned models, except for the 10.5B model, due to the computational resource limitations. We included the ensembled model and individual model outputs as the reranking candidates.

Direction	Configuration	#Params.	Cross+self attention	LN pos.	Encoder				Decoder			
					Layer	d_{model}	d_{ffn}	#Heads	Layer	d_{model}	d_{ffn}	#Heads
Both	NTT-Base	547M		Pre.	9	1024	8192	16	9	1024	8192	16
Both	ABCI-Base	622M		Pre.	9	1024	16384	16	9	1024	4096	16
Both	ABCI-EncBig	2.0B		Pre.	12	1024	65536	16	9	1024	8192	16
Both	ABCI-EncDeep	736M		Pre.	18	1024	8192	16	9	1024	8192	16
Both	Failab-EncBig	1.7B		Pre.	9	1024	61440	16	9	1024	16384	16
Both	Failab-DecBig	1.7B		Pre.	9	1024	16384	16	9	1024	61440	16
Both	NTT-A	408M		Post.	6	1024	8192	16	6	1024	8192	16
Both	NTT-B	547M		Post.	9	1024	8192	16	9	1024	8192	16
Both	NTT-C	622M		Post.	9	1024	16384	16	9	1024	4096	16
Both	NTT-D	698M		Post.	9	1024	16384	16	9	1024	8192	16
En-Ja	NTT-E	547M		Pre.	9	1024	8192	16	9	1024	8192	16
En-Ja	NTT-F	509M	✓	Post.	9	1024	8192	16	9	1024	8192	16
En-Ja	NTT-G	551M	✓	Post.	10	1024	8192	16	10	1024	8192	16
Both	Failab-LM	10.5B	✓	Pre.	16	4096	16384	32	32	4096	16384	32

Table 4: List of model configurations used by the primary translation module. The upper half of the table shows the models also used in last year's submission system (Morishita et al., 2022b), and the lower half shows the models newly trained this year. d_{model} and d_{ffn} respectively denote sizes of embedding and feedforward layers. LN pos. means the position of layer normalization. Post. denotes that layer normalization is applied after the residual connection. Pre. denotes that layer normalization is performed before the residual connection. ABCI-Base and NTT-Base were each trained with two different seeds.

5 Reranking

To enhance translation quality, we applied a reranking process to the candidate set of hypotheses translated by each model described in Section 4. We conducted a comparative analysis of the various methods, as presented in the following sections.

5.1 Methods

The reranking approach was used to obtain the final output $\hat{y}$ from C, where C represents the candidate set generated by multiple translation models for a given source x.

Quality Estimation (QE) This approach involves scoring the candidates using quality estimation methods (e.g., COMET-QE) and selecting the one with the highest score, as follows:

$$\hat{y} = \operatorname*{argmax}_{c\in\mathcal{C}} \mathrm{QE}\left(x, c\right). \tag{2}$$

where, $\mathrm{QE}(\cdot,\cdot)$ is a quality estimation function.

Minimum Bayes Risk (MBR) This method uses reference-based metrics such as COMET, to yield the best output as follows (Fernandes et al., 2022);

$$\hat{y} = \operatorname*{argmax}_{c_i\in C} \frac{1}{|C|} \sum_{c_j=1}^{|C|} \mathrm{RefMetric}\left(c_i, c_j\right). \tag{3}$$

where $\mathrm{RefMetric}(\cdot,\cdot)$[6] is a reference-based metric. Note that MBR uses reference-based metrics but not reference texts. MBR is applied to the output of a single model in Fernandes et al. (2022). We extended this method to the outputs from multiple models.

MBR after QE (QE → MBR) This approach is a combination of QE and MBR (Fernandes et al., 2022). We denoted the top-p samples from set C, according to the score calculated by the quality estimation function $\mathrm{QE}(\cdot,\cdot)$, as $C_{\text{top-p}}$. Then, MBR was applied for $C_{\text{top-p}}$.

5.2 Post Evaluation

We experimented with the performance of the translation models and the reranking process. Note that this experiment was conducted after the primary system was submitted.

5.2.1 Experimental Setup

We used WMT21-COMET-QE[7] and WMT22-CometKiwi (Rei et al., 2022b)[8] for the QE, and WMT22-COMET-DA[9] as the refernece-

[6] Some reference-based metrics, such as COMET, also use source x as an input.

[7] https://unbabel-experimental-models.s3.amazonaws.com/comet/wmt21/wmt21-comet-qe-mqm.tar.gz

[8] https://huggingface.co/Unbabel/wmt22-cometkiwi-da

[9] https://huggingface.co/Unbabel/wmt22-comet-da

Models		En→ Ja	Ja → En
NT5	single model	8	8
	4-models ensemble	1	1
	all models ensemble	1	1
NTT	single model	70	40
	all models ensemble	10	10
Failab-LM		10	10
Total		100	70

Table 5: Breakdown of candidates for reranking. The NT5 four-model ensemble consists of ABCI-EncBig, ABCI-EncDeep, Failab-EncBig, and Failab-DecBig. The NT5 all-model ensemble consists of NTT-Base (two different seeds), ABCI-Base (two different seeds), ABCI-EncBig, ABCI-EncDeep, Failab-EncBig, and Failab-DecBig. The NTT all-model ensemble consists of NTT-A to NTT-G.

based metric for MBR. WMT22-COMET-DA was also used as the evaluation metric. The candidate sets contained 100 hypothesis for En→Ja and 70 for Ja→En. The breakdown of each candidate set is shown in Table 5.

5.2.2 Reranking Analysis

Table 6 shows the results of the reranking. Oracle (a) is the upper-bound setting, selecting the final output by using WMT22-COMET-DA with reference text (denoted r):

$$\hat{y} = \operatorname*{argmax}_{c \in \mathcal{C}} \texttt{WMT22-COMET-DA}\,(c, r)\,. \qquad (4)$$

Comparing the QE and MBR approaches (f and g vs. q) showed that MBR achieved higher performance. As for the QE approach, WMT21-COMET-QE achieved better performance than WMT22-CometKiwi in both translation directions (f vs. g). Therefore, we used WMT21-COMET-QE for the QE → MBR approach. The best performance was achieved by the QE → MBR at smaller p (h, i, j and k) in both translation directions. Moreover, QE → MBR often achieved a higher performance than MBR. These results suggest that the poor quality hypothesis in the candidates has a negative impact on MBR reranking.

5.2.3 10.5B Model Analysis

As described in Section 4.1, we trained a large-scale translation model with 10.5B parameters (failab-LM). The experimental results showed that the 10.5B parameters models were inferior to the best single model. However, when comparing the loss, we found that the 10.5B parameters models achieved a lower loss than the other smaller models. These results might suggest that 10.5B is overparametrized for sentence-level translation. For document-level translation, there may be an opportunity to harness the potential of the large number of parameters. However, the availability of document-level parallel corpora for En↔Ja is limited, highlighting the necessity of expanding the resources for document-level data.

In studies on large language models (LLMs), several papers discuss the scaling laws. For example, Hoffmann et al. (2022) introduces the optimal number of tokens with respect to model size, which is often referred to as the Chinchilla rule in the community. If we straightforwardly apply this rule to MT models, the optimal tokens of the 10B parameters MT model are estimated to be 205.1B tokens. This is much larger than the tokens we used to train for 10.5B parameter models. Therefore, we posit that effectively harnessing the 10.5B model may be possible by increasing both the quantity of training data and the number of training steps. We could not investigate this perspective due to the limited time and computational resources. Thus, we leave to clarify this perspective for future work.

5.2.4 Effectiveness of applying cross+self-attntion

In a preliminary experiment, we confirmed the effectiveness of applying cross+self-attention by comparing performance with the standard setup (cascading computation of self- and cross-attentions) of Transformer encoder-decoder models. Table 7 shows the results of our preliminary experiments. As we see, there were no considerable performance degradations when we compared the performance of cross+self-attention models (NTT-F) with those of standard self-attention and cross-attention cascading models (NTT-B).

In addition, cross+self-attention models reduce the computation of cascading self- and cross-attention into single cross+self-attention. Therefore, the cross+self-attention models are slightly faster and require less memory than standard self-attention and cross-attention cascading models.

6 Submission System

Initially, we planned to submit several versions of the system, with the highest-scoring system selected as the final version. However, the reranking process took longer than expected, and we were

ID	Candidates	Reranker	En→Ja		Ja→En	
			wmt22test	wmt23test	wmt22test	wmt23test
(a)	All	Oracle	0.9298	0.9136	0.8804	0.8737
(b)	Failab-LM	-	0.8840	0.8590	0.8127	0.8119
(c)	NT5-ensemble	-	0.8926	**0.8713**	**0.8269**	**0.8234**
(d)	NTT-ensemble	-	0.8880	0.8633	0.8215	0.8198
(e)	Best Single Model	-	**0.8937**	0.8692	0.8232	0.8198
(f)	All	WMT21-COMET-QE	**0.9085**	**0.8879**	**0.8379**	**0.8345**
(g)	All	WMT22-CometKiwi	0.9049	0.8847	0.8338	0.8329
(h)	All	QE(Top10%) → MBR	0.9102	**0.8905**	0.8425	0.8372
(i)	All	QE(Top20%) → MBR	**0.9111**	0.8904	**0.8437**	0.8393
(j)	All	QE(Top30%) → MBR	**0.9111**	**0.8905**	0.8425	0.8394
(k)	All	QE(Top40%) → MBR	0.9107	0.8903	0.8429	**0.8402**
(l)	All	QE(Top50%) → MBR	0.9099	0.8901	0.8431	0.8401
(m)	All	QE(Top60%) → MBR	0.9096	0.8897	0.8426	0.8401
(n)	All	QE(Top70%) → MBR	0.9092	0.8897	0.8418	0.8396
(o)	All	QE(Top80%) → MBR	0.9092	0.8892	0.8411	0.8390
(p)	All	QE(Top90%) → MBR	0.9088	0.8891	0.8408	0.8389
(q)	All	MBR	0.9084	0.8890	0.8405	0.8384

Table 6: Post evaluation results. Best Single model (b) represents the highest score achieved by an individual translation model (not an ensembled model).

Configuration	Cross+self	#Params.	En→Ja	
	attention		wmt22test	wmt23test
NTT-B		547M	0.8865	0.8624
NTT-F	✓	509M	0.8862	0.8612
NTT-G	✓	551M	0.8862	0.8635

Table 7: Comparison of performance on applying cross+self-attention compared with the standard setup (cascading computation of self- and cross-attentions) of Transformer encoder-decoder models.

unable to submit multiple submissions within the time limit. Therefore, the system that was actually submitted system was slightly different from the one described in this paper, as follows:

- For the En→Ja system, we submitted the results of the ensembled model of NTT-A to NTT-G.

- For the Ja→En system, we opted for the QE(Top 80%) → MBR configuration.

Unlike the post evaluation setting (Section 3.3), these models were fine-tuned using all of the WMT'20 test set, the WMT'20 development set, the WMT'21 test set, and the WMT'22 test set.

7 Conclusion

This paper described our submission system for the constrained track of the WMT'23 general translation task. We developed a translation system for En↔Ja. We perform reranking on the candidates generated by multiple translation models, which include a large-scale model with 10.5 billion parameters. Post evaluation (Section 5.2) confirmed the limitations of sentence-level translation quality improvement through model scaling and the effectiveness of our reranking approach.

Acknowledgments

We thank an anonymous reviewer who provided feedback. Also, we would like to also appreciate the member of Tohoku NLP Group for their cooperation in conducting this research. This work was mainly done under the NTT-Tohoku University collaborative research agreement. The work of Jun Suzuki was partly supported by JST Moonshot R&D Grant Number JPMJMS2011 (fundamental research).

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman

Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook AI's WMT20 News Translation Task Submission. In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:1706.02677*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *CoRR*, abs/2001.08361.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted Absolute Position Embedding for Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3309–3321.

Hayato Kobayashi. 2018. Frustratingly Easy Model Ensemble for Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4165–4176, Brussels, Belgium. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, and Maja Popović. 2023. Findings of the 2023 Conference on Machine Translation (WMT23). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, Singapore, Singapore (Hybrid). Association for Computational Linguistics.

Taku Kudo. 2006. MeCab: yet another part-of-speech and morphological analyzer. http://mecab.sourceforge.net.

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 (EMNLP)*, pages 66–71. Association for Computational Linguistics.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022a. JParaCrawl v3.0: A Large-scale English-Japanese Parallel Corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6704–6710, Marseille, France. European Language Resources Association.

Makoto Morishita, Keito Kudo, Yui Oka, Katsuki Chousa, Shun Kiyono, Sho Takase, and Jun Suzuki. 2022b. NT5 at WMT 2022 General Translation Task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 318–325, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

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 (NAACL)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Stephan Peitz, Sarthak Garg, Udhay Nallasamy, and Matthias Paulik. 2019. Cross+self-attention for transformer models. `https://github.com/pytorch/fairseq/files/3561282/paper.pdf`.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English Subtitle Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022a. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are References Really Needed? Unbabel-IST 2021 Submission for the Metrics Shared Task.

In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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 (ACL)*, pages 86–96. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 464–468.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. 2023. B2T Connection: Serving Stability and Performance in Deep Transformers. In *Findings of the Association for Computational Linguistics (ACL)*, pages 3078–3095, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

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 31 (NIPS)*, pages 5998–6008.