NTTSU at WMT2024 General Translation Task

Minato Kondo[♠], **Ryo Fukuda**[♢], **Xiaotian Wang**[♠], **Katsuki Chousa**[♢], **Masato Nishimura**[♠], **Kosei Buma**[♠], **Takatomo Kano**[♢], **Takehito Utsuro**[♠]
[♠]University of Tsukuba [♢]NTT Communication Science Laboratories

Abstract

The NTTSU team's submission leverages several large language models developed through a training procedure that includes continual pre-training and supervised fine-tuning. For paragraph-level translation, we generated synthetic paragraph-aligned data and used these data for training. In the task of translating Japanese to Chinese, we focused on speech domain translation. Specifically, we built Whisper models for Japanese automatic speech recognition (ASR). Since the dataset used for Whisper training contains many noisy data pairs, we combined the Whisper outputs using ROVER (Fiscus, 1997) to refine the transcriptions. Furthermore, we employed forward translation from audio as data augmentation, using both ASR models and a base translation model. To select the best translation from multiple hypotheses of the models, we applied Minimum Bayes Risk decoding after Quality Estimation (Fernandes et al., 2022), incorporating scores such as COMET-QE, COMET, and cosine similarity by LaBSE. We explored three different reranking strategies to handle two types of candidates from sentence- and paragraph-level translation and employed a fusion method that integrates all three.

1 Introduction

This paper provides a system description of the NTTSU team's submissions to WMT 2024. We took part in the General Translation Task (Kocmi et al., 2024a) for English-to-Japanese (En-Ja) and Japanese-to-Chinese (Ja-Zh). This task has three tracks with different constraints on the use of training data and pre-trained models. For En-Ja, we participated in the constrained track, which provides sets specifically allow training data and pre-trained models for use in traning the translation models. Additionally, for Ja-Zh, we participated in the open track, which allows the use of software and data under any open-source license.

Our team's submission leveraged several large language models developed through a training procedure (Guo et al., 2024; Kondo et al., 2024) that includes continual pre-training and supervised fine-tuning. For paragraph-level translation, we generated synthetic paragraph-aligned data and used these data for training.

In the task of translating Japanese to Chinese, we focused on speech domain translation. Specifically, we built Whisper models (Radford et al., 2022) for Japanese automatic speech recognition (ASR). We used the YODAS dataset (Li et al., 2024) for Whisper training. Since these data contained many noisy data pairs, we combined the Whisper outputs using ROVER (Fiscus, 1997) to refine the transcriptions. Furthermore, to enhance the robustness of the translation model against errors in the transcriptions, we performed data augmentation by forward translation from audio, using both ASR and base translation models.

To select the best translation from multiple hypotheses of the models, we applied Minimum Bayes Risk decoding after quality estimation (Fernandes et al., 2022), incorporating scores such as COMET-QE, COMET, and cosine similarity by LaBSE. We explored three different reranking strategies to handle two types of candidates from sentence- and paragraph-level translation and employed a fusion method that integrates all three.

2 System Overview

Our system had three main components: automatic speech recognition (ASR) models, translation models, and a reranking.

This year, speech domain translation was newly incorporated in the above task, and audio data, along with the organizer's transcription, were provided as input data. We were interested in the feasibility of speech translation from Japanese, so we created an ASR model for the Ja-Zh and used its transcription as the additional source text. More-

Proceedings of the Ninth Conference on Machine Translation, pages 270–279
November 15-16, 2024 ©2024 Association for Computational Linguistics

over, we used ROVER to refine the transcriptions.

For the translation model's architecture, we employed and trained the Transformer model and LLMs. To train the LLms, we carried out monolingual/parallel continual pre-training and supervised fine-tuning. The evaluation for this year was conducted at the paragraph level. To address this, we created sentence- and paragraph-level parallel data and utilized these data to build translation models for each level.

During the inference step, we used the translation models to independently translate at the sentence and the paragraph level, generating multiple candidates. We then selected the best translation candidate using a reranking that combines sentence- and paragraph-level reranking with MBR decoding after quality estimation.

3 Automatic Speech Recognition

For Ja-Zh speech translation, we fine-tuned various Whisper-based ASR models for the Japanese ASR task. We used the Japanese subset (ja100) of the YODAS dataset, which consists of approximately 3,000 hours of speech and transcriptions.

3.1 Dataset

During the dataset review, we found that the YODAS dataset contained many incorrect transcriptions (e.g., music and non-Japanese speech samples). To mitigate the negative impact of these incorrect samples, we refined the YODAS dataset. We integrated transcriptions of multiple hypotheses transcription generated from multiple ASR models to create a tuning dataset. Specifically, the following procedure was used to generate tuning data.

1. **Generation** We performed beam search decoding with multiple ASR models to generate multiple ASR hypotheses for each speech sample in ja100. This process yielded a set of hypotheses equal to the number of ASR models multiplied by the beam size. We set a beam size of 4.

2. **Language-based Filtering** We applied multi-step filtering for the YODAS dataset. First, we used Whisper to transcribe the speech; then, we applied the Compact Language Detector v3 (CLD3) [1] to filter non-Japanese language. Next, we excluded the transcriptions that did not contain Japanese-specific characters (i.e., *hiragana* or *katakana*). After language-based filtering, we filterd out uncertain transcription that contained repetition. Specifically, texts with bi-grams appearing more than six times were excluded.

3. **Combination** After filtering, we combined multiple ASR hypotheses using the Recognizer Output Voting Error Reduction (ROVER) (Fiscus, 1997).

4. **CER-based Filtering** To filter uncertain samples of ROVER results, we applied accuracy-based filtering. We measured the character error rate (CER) between the ROVER results and the original subtitles in YODAS. A high CER indicates that either one or both may be significantly inaccurate. For the ASR training, we constructed a development set of 2k samples of CER ≤ 0.3 data. No CER filtering was applied to the training set because no positive effect was observed in preliminary experiments. Finally, all ROVER results except the development set (1,614,110 segments) were treated as the training set. For the training of MT using the ASR data (described in §4.3), samples with CER ≤ 0.3 (693,304 segments) were used.

To compare the quality of the original subtitles and the ROVER results, we subjectively evaluated the two corresponding transcriptions of 100 randomly selected samples. As a result, we determined that the ROVER results were of higher quality.

3.2 Model

To create the tuning data, we used two pre-trained ASR models: Whisper large-v3[2] and kotoba-whisper-v1.1[3], a Japanese-specific ASR model.

3.3 Training

Using the tuning data created through the above procedure with the two ASR models, we separately fine-tuned each of these models. The training of the model was conducted using the AdamW optimizer, with parameters set as $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$. We employed a linear decay learning rate scheduler and set the warmup steps to 500. The model's parameters were saved every 4000 steps.

[1] https://github.com/google/cld3

[2] https://huggingface.co/openai/whisper-large-v3

[3] https://huggingface.co/kotoba-tech/kotoba-whisper-v1.1

The training was carried out with a batch size of 32 samples over a single epoch. We selected the best model based on the loss in the dev set.

3.4 Inference

During inference, we performed a beam search with a beam size of 4 and combined these four hypotheses using ROVER. For the post-processing of the ASR stage, we integrated punctuation and sentence segmentation into the transcription. We used the fine-tuned version of xlm-roberta[4] and Bunkai (Hayashibe and Mitsuzawa, 2020)[5] for punctuation insertion and sentence segmentation, respectively. Finally, the two types of hypotheses from the two ASR models were passed to MT.

In the data generation process for MT training (§4.3), ROVER was not performed and the top-1 hypothesis of the beam search was used.

4 Primary Translation Model

4.1 Dataset

We used two types of text corpora: monolingual and parallel data. Monolingual data are used for monolingual continual pre-training, while parallel data are used for parallel continual pre-training, sentence-level supervised fine-tuning (SFT), and paragraph-level SFT.

En-Ja We used the following monolingual corpora: Common Crawl (Kocmi et al., 2022), Leipzig Corpora (Goldhahn et al., 2012), News Crawl, and News Commentary (Kocmi et al., 2023). We also used JParaCrawl v3.0 (Morishita et al., 2022), News Commentary (Kocmi et al., 2023), the Kyoto Free Translation Task Corpus (KFTT) (Neubig, 2011), TED Talks (Barrault et al., 2020), and past WMT test data as the parallel data. Since JParaCrawl v3.0 is automatically created and contains a certain amount of noisy data, we filtered the corpus based on sentence embeddings. We employed LaBSE (Feng et al., 2022) to embed the source and target sentences and then filtered out the sentence pairs in which the similarities are not between 0.4 and 0.9.

Ja-Zh We used the following monolingual corpora: Leipzig Corpora (Goldhahn et al., 2012), News Crawl, and News Commentary (Kocmi et al., 2023). In order to obtain parallel data for continual pre-training, we used JParaCrawl Chinese v2.0 (Nagata et al., 2024). Since this corpus also contains noisy data, we filtered them using the same method as in the En-Ja task. For sentence-level SFT, we used ASPEC-JC (Nakazawa et al., 2016) and Flores-200 (NLLB Team et al., 2022) as training and development sets. In addition to the data for sentence-level SFT, we used News Commentary, WIT3 (Cettolo et al., 2012), Global Voice, and Neulab TedTalks (Tiedemann, 2012) as parallel corpora with context information for paragraph-level SFT.

4.2 Model Selection

For the En-Ja task, we used the largest available LLM in the constrained track, `Llama-2-13b`[6] (Touvron et al., 2023). For the Ja-Zh task, we used `TowerBase-13B-v0.1` [7] (Alves et al., 2024), a model based on Llama-2-13b that has been continually pre-trained with monolingual and parallel data.

Additionally, we developed and deployed a Transformer (Vaswani et al., 2017) model trained from scratch. As training data, we used JParaCrawl v3.0 for the En-Ja task and JParaCrawl Chinese v2.0 for the Ja-Zh task. The model configuration and hyperparameters are detailed in Table 1.

4.3 LLM Training Procedure

We conducted a three-stage training process based on research conducted on translation models using LLMs (Guo et al., 2024; Kondo et al., 2024). In the first stage, we performed continual pre-training with monolingual data. In the second stage, we conducted continual pre-training with parallel data. Finally, in the third stage, we carried out supervised fine-tuning. The detailed model configuration and hyperparameters are given in Table 1.

Monolingual Continual Pre-Training It has been reported that LLMs primarily pre-trained in English, such as Llama-2, have lower translation accuracy for languages other than English (Xu et al., 2024). Therefore, we performed continual pre-training using monolingual data to enhance the

[4] https://huggingface.co/1-800-BAD-CODE/xlm-roberta_punctuation_fullstop_truecase

[5] https://github.com/megagonlabs/bunkai

[6] https://huggingface.co/meta-llama/Llama-2-13b-hf

[7] https://huggingface.co/Unbabel/TowerBase-13B-v0.1

[8] https://github.com/facebookresearch/fairseq

[9] https://github.com/huggingface/transformers

Transformer Enc-Dec model	
Subword Size	32,000
Architecture	Transformer (big)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-8$)
LR Scheduler	Inverse Square root decay
Warmup Steps	4,000
Max Learning Rate	1e-3
Dropout	0.3
Gradient Clipping	1.0
Label Smoothing	0.1
Batch Size	512,000 tokens
Number of Updates	50,000 steps
Implementation	`fairseq`[8] (Ott et al., 2019)
Common Settings for All LLMs Training Phases	
Warmup Ratio	1%
Gradient Clipping	1.0
Weight Decay	1.0
Implementation	`transformers`[9] (Wolf et al., 2020)
Continual Pre-Training Settings	
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 1e-5$)
LR Scheduler	Cosine
Max Learning Rate (full / LoRA)	1.5e-4 / 2.0e-4
Batch Size	1,024 samples
Epoch	1
Context Length	2,048
Supervised Fine-tuning Settings	
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$)
LR Scheduler	Inverse Square root decay
Max Learning Rate	2.0e-4
Batch Size	1,024 samples
Epoch	3
LoRA Settings	
Rank / Alpha	16 / 32
Dropout	0.05
Target Modules	QKVO, FFN

Table 1: Model configuration and hyperparameters.

generation capabilities in languages other than English.

We used randomly sampled data from the monolingual corpora described in §4.1. For the En-Ja task, we created two models, ver1 and ver2, and trained them using randomly sampled data of 1B and 4B tokens, respectively. In contrast, for the Ja-Zh task, we trained only a single model with randomly sampled data of 1B tokens due to the lack of time and GPU resources.

Parallel Continual Pre-Training After completing monolingual continual pre-training, we performed continual pre-training using parallel data. Based on the findings of (Kondo et al., 2024), we used data where the source text is followed by its translation.

For the En-Ja task, the ver1 model was trained using LoRA (Hu et al., 2022), while the ver2 model was trained with full weights. Additionally, ver1 was trained using only the sentence-level parallel data from JParaCrawl v3.0, whereas ver2 utilized JParaCrawl v3.0 along with TED and News Commentary as pseudo-paragraph data.

Supervised Fine-Tuning After completing continual pre-training in monolingual and parallel data, we performed supervised fine-tuning using LoRA. The prompts applied to the training data were the same as those used in ALMA (Xu et al., 2024), and the same prompts were used during inference. Note that loss in the prompt outputs was excluded during training (Xu et al., 2024; Kondo et al., 2024).

Additionally, for domain adaptation, we performed SFT using data from each specific domain. For the En-Ja task, the ver1 model was fine-tuned using TED Talks, KFTT, and past WMT test data. In contrast, the ver2 model was fine-tuned with the same three datasets as ver1, plus two additional settings: using only the news domain data and using only the social domain data each from past WMT test data. Note that the past WMT test data used for SFT training consisted of the WMT20 development and test data, with the other test data from WMT21 to WMT23. For WMT21, both En-Ja and Ja-En directions were included, while WMT22 and WMT23 were composed only of the Ja-En direction. Additionally, the development data for all SFT were the WMT22 En-Ja data. As a result, we obtained a total of eight fine-tuned models for En-Ja. For Ja-Zh, we also performed SFT with synthetic data to enhance robustness against errors in the transcription for the speech domain. These data were constructed by forward translation from audio data using ASR and Transformer models.

5 Reranking

To enhance translation quality, we applied reranking to the candidate sentences. We conducted a comparative analysis of various methods and strategies, as described in §5.1 and §5.3, on the candidate generated by the methods described in §5.2.

5.1 Methods

The reranking approach is used to obtain the final output $\hat{y}$ from the set of candidate sentences $\mathcal{C}$ generated by the methods described in §5.2.

Quality Estimation (QE) This approach involves evaluating the candidates using reference-free quality estimation techniques, such as COMET-QE (Rei et al., 2021, 2022, 2023) and sentence embedding-based similarity, and subsequently selecting the candidate with the highest score, as follows:

$$\hat{y} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \sum_{i=1}^{m} \lambda_i \mathrm{QE}_i(x, c), \qquad (1)$$

where $\mathrm{QE}_i(\cdot,\cdot)$ is a reference-free quality estimation function and λ_i represents its weight, subject to $\sum_{i=1}^{m} \lambda_i = 1$.

Minimum Bayes Risk (MBR) decoding MBR decoding (Fernandes et al., 2022) employs reference-based metrics to rank translation candidates. It aims to identify the translation that maximizes expected utility while equivalently minimizing the risk (Meister et al., 2020; Eikema and Aziz, 2020) as follows:

$$\hat{y}_i = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} \frac{1}{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} \mathrm{RefMetric}(c_i, c_j), \qquad (2)$$

where $\mathrm{RefMetric}(\cdot,\cdot)$ is a reference-based metric. Note that MBR decoding scores the candidate using reference-based metrics by treating all candidates as reference texts without using an actual reference text.

MBR after QE (QE→MBR) This approach integrates QE with MBR decoding (Fernandes et al., 2022). The scores produced by the quality estimation procedure determined the top-n sample set from candidate set $\mathcal{C}$ as $\mathcal{C}_{\text{top-n}}$. Subsequently, MBR is applied to $\mathcal{C}_{\text{top-n}}$.

5.2 Candidate Generation

We generated five candidates for each model by varying the sampling methods during generation. For the speech domain in Ja-Zh, we had two extra transcriptions from our ASR models in addition to the official one. As a result, we generated five candidates for these two transcriptions and LLM models in the same manner. For models based on Llama-2-13b and TowerBase-13B-v0.1, the five methods were as follows: 1. greedy decoding (no sampling), 2. beam search with a beam size of 4, 3. temperature of 0.9, 4. temperature of 0.5, and 5. temperature of 0.3. For methods 3, 4, and 5, parameters other than temperature were set with top_p at 0.6 and top_k at 50. We also used the top-5 candidates from beam search for the Transformer with a beam size of 6. As a result, a total of 45 candidate sentences were generated for the En-Ja task using the eight SFT models described in §4.3, along with the Transformer model, making a total of nine models.

Furthermore, for each SFT model, we employed two approaches to generate candidates.

Sentence-Level Generation First, we used pySBD[10] (Sadvilkar and Neumann, 2020) to split the original paragraph-level test data into sentences, and then we performed sentence-level inference to generate sentence candidates $\mathcal{C}_{sent}$.

Paragraph-Level Generation We used the paragraph data directly as model input for generating paragraph candidates $\mathcal{C}_{para}$.

5.3 Reranking System

For the two types of candidates mentioned in §5.2, we used three reranking strategies and one fusion method that integrates all three.

Synthesized Paragraph Reranking In each sentence-level inference result, we concatenated the sentences that originally belonged to the same paragraph in order and then performed reranking on the synthesized paragraph.

Individual Sentence Reranking We performed sentence-level reranking on the sentence candidates $\mathcal{C}_{sent}$ and then reconstructed the paragraphs from the final reranked results.

Full Paragraph Reranking The paragraph candidates $\mathcal{C}_{para}$ were used as the objects of reranking, directly generating paragraph-level results.

Multi-Attribute Candidate Reranking We established a larger set of multi-attribute candidates $\mathcal{C}_{mac}$ according to the three reranking strategies mentioned above:

- Synthesized paragraph candidates by concatenating the sentences in order from sentence candidates $\mathcal{C}_{sent}$.
- Paragraph data reconstructed on the results obtained by different reranking methods from Individual Sentence Reranking.
- Paragraph candidates $\mathcal{C}_{para}$ generated by Paragraph-Level Generation.

[10]https://github.com/nipunsadvilkar/pySBD

	CER (YODAS)	COMET (WMT test)
Whiper large-v3	7.7	0.4598
+ FT	4.8	0.4601
kotoba-whisper-v1.1	12.6	0.4407
+ FT	5.0	0.4518
Official transcription	-	0.7278

Table 2: ASR performances and their translation accuracies. Second column is CER results on the evaluation data of the YODAS dataset. Third column is COMET results on the speech domain of this year's WMT test set.

Then, $\mathcal{C}_{mac}$ was used for paragraph-level reranking.

6 Experiment and Analysis

6.1 Results of ASR

The second column of Table 2 shows the ASR results (with and without fine-tuning on the YODAS dataset) for the Ja-Zh speech translation. Note that this evaluation was not done in combination with the ROVER system. We confirmed that fine-tuning improved the recognition performance on the YODAS dataset. The third column of Table 2 shows the translation results[11] for the WMT test set. Fine-tuning resulted in a relative improvement of 2.5% for kotoba-whisper-v1.1, but no significant improvement was observed for Whisper-large-v3, even through it demonstrated high ASR performance before fine-tuning. Moreover, our models performed worse than the official transcriptions. We trained the ASR models using relatively short audio samples, whereas the audio samples in the test set were longer than 30 seconds. This gap between the training and test conditions likely contributed to the degradation in speech recognition accuracy. In addition, we prepared ASR models for a wide range of topics, domains, and noise levels for open-domain speech input. For this purpose, we used the YODAS dataset instead of datasets such as TED, CSJ, and Libri, which contain clean speech with human transcriptions. However, this strategy did not turn out to be suitable for the WMT test set. In fact, when we listened to the speech from the test set, the SNR was high and clean. This gap may have also contributed to the degradation. These findings will be leveraged for future improvements.

[11] We used `wmt22-comet-da`. During this evaluation, we used the official transcription as the source text for all hypotheses because it would be the most accurate transcription. https://huggingface.co/Unbabel/wmt22-comet-da

Model	Input	COMET22
Ver1	Sentence	**0.8218**
	Paragraph	0.7666
Ver2	Sentence	**0.8352**
	Paragraph	0.8349

Table 3: COMET Scores of Sentence-Level and Paragraph-Level SFT on WMT23 En-Ja test data

Scoring Function	COMET22
LaBSE-cos	0.8364
Comet-QE20	0.8797
Comet-QE21	0.8837
CometKiwi22	0.8821
CometKiwi23-xl	0.8819
0.5×Comet-QE20 + 0.5×LaBSE-cos	0.8835
0.8×Comet-QE21 + 0.2×LaBSE-cos	**0.8856**
0.9×CometKiwi22 + 0.1×LaBSE-cos	0.8824
0.9×CometKiwi23-xl + 0.1×LaBSE-cos	0.8830
MBR ratio	COMET22
QE (Top 10%)	0.8911
QE (Top 20%)	0.8940
QE (Top 30%)	0.8949
QE (Top 40%)	0.8950
QE (Top 50%)	**0.8955**
QE (Top 60%)	**0.8955**
QE (Top 70%)	0.8954
QE (Top 80%)	0.8953
QE (Top 90%)	0.8953
100%	0.8953

Table 4: COMET Scores of QE and MBR decoding on WMT23 En-Ja test data. The 45 candidates used were generated by the methods in §5.2. MBR decoding was performed after QE with the best scoring function, `0.8×Comet-QE21 + 0.2×LaBSE-cos`.

6.2 Sentence-Level versus Paragraph-Level in SFT

In the SFT experiments using past WMT test data, we evaluated whether sentence-level or paragraph-level source texts achieved better accuracy by assessing them with COMET (`wmt22-comet-da`) on the WMT23 En-Ja test data. For paragraph-level training, the data were reconstructed from sentence-level to paragraph-level based on the .xml files provided by WMT. Table 3 shows the results, indicating that sentence-level inputs achieved higher accuracy than those of paragraph-level inputs. Therefore, for subsequent SFT, we used only sentence-level inputs.

6.3 Results of Quality Estimation

To identify the scoring function in Eq.(1) that yields the highest translation accuracy, we compared ten

ID	System	MetricX ↓	CometKiwi ↑
(a)	Synthesized Para	2.8830	0.7260
(b)	Individual Sent	2.8100	0.7273
(c)	Full Para	2.7263	0.7260
(d)	Multi-Attribute	**2.6321**	**0.7310**

Table 5: Results of Reranking Systems on WMT24 En-Ja test data. Systems (a)~(c) used 45 candidates, while System (d) used 100 candidates, consisting of 45 from $\mathcal{C}_{sent}$, 45 from $\mathcal{C}_{para}$, and 10 results obtained by Individual Sentence Reranking using the 10 methods listed in Table 4. All of the system results are based on Top 50% MBR decoding after QE with the best scoring function, `0.8×Comet-QE21 + 0.2×LaBSE-cos`.

different scoring functions based on the findings in the paper. We used COMET-QE and LaBSE cosine similarity for scoring functions and evaluated them with COMET on the WMT23 En-Ja test data. Since the WMT23 test data are sentence-level, we used the 45 candidate sentences generated through paragraph-level generation, where each sentence was directly input, as described in §5.2. Additionally, the reranking system utilized Full Paragraph Reranking, as described in §5.3. Table 4 shows the results, indicating that `0.8×wmt21-comet-qe + 0.2×LaBSE-cos` achieved the highest accuracy. Therefore, this scoring function was adopted for subsequent experiments and finally the submitted system.

6.4 Reslults of MBR after QE

We investigated the proportion of MBR that achieved the highest accuracy under the same conditions as in §6.3. Table 4 shows the results, indicating that accuracy was maximized at 50%. Therefore, in subsequent experiments and the submitted system, the proportion of MBR was set to 50%.

6.5 Results of Reranking Systems

Table 5 shows the results of the reranking system on WMT24 En-Ja. We used MetricX-23-XL (Juraska et al., 2023) and CometKiwi-DA-XL (Rei et al., 2023) as evaluation metrics, consistent with the WMT24 preliminary report (Kocmi et al., 2024b). From these results, it was found that the Multi-Attribute Candidate Reranking achieved the highest accuracy. Therefore, we adopted Multi-Attribute Candidate Reranking for the submitted system.

7 Conclusion

In this paper, we described our system for the WMT'24 General Translation Task. We developed ASR models for the speech domain in Ja-Zh and used Transformer and LLMs for the translation models. We trained LLMs using a three-stage training process: Monolingual Continual Pre-training, Parallel Continual Pre-Training, and Supervised Fine-Tuning. Finally, we applied reranking method and strategies to the translation candidates generated by the translation models. Our analyses confirmed the effectiveness of our reranking method and strategies for paragraph-level translation.

Acknowledgments

This work was done mainly under a collaborative research agreement between NTT and Tsukuba University. Additionally, this work partially used computational resources of Pegasus provided by the Multidisciplinary Cooperative Research Program in the Center for Computational Sciences, University of Tsukuba.

References

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv:2402.17733*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. 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.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on 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 (Volume 1: Long Papers)*, 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.

J.G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Jiaxin Guo, Hao Yang, Zongyao Li, Daimeng Wei, Hengchao Shang, and Xiaoyu Chen. 2024. A novel paradigm boosting translation capabilities of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 639–649, Mexico City, Mexico. Association for Computational Linguistics.

Yuta Hayashibe and Kensuke Mitsuzawa. 2020. Sentence boundary detection on line breaks in Japanese. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 71–75, Online. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. 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, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinþór Steingrímsson, and Vilém Zouhar. 2024a. Findings of the WMT24 general machine translation shared task:
the LLM era is here but mt is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, USA. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popovic, Mariya Shmatova, Steinþór Steingrímsson, and Vilém Zouhar. 2024b. Preliminary wmt24 ranking of general mt systems and llms. *arXiv:2407.19884*.

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, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Minato Kondo, Takehito Utsuro, and Masaaki Nagata. 2024. Enhancing translation accuracy of large language models through continual pre-training on parallel data. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 203–220, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. 2024. Yodas: Youtube-oriented dataset for audio and speech. *arXiv:2406.00899*.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022. 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.

Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. 2024. A japanese-chinese parallel corpus using crowdsourcing for web mining. *arXiv:2405.09017*.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv:1902.01382*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

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, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. 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. 2022. 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.

Nipun Sadvilkar and Mark Neumann. 2020. PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.