RFMT at WMT 2026 General Translation Task

Ryosuke Matsuda[1], Keito Kudo[1,2], Ryo Fujii[1,3], Takumi Ito[1,4], Makoto Morishita[1,3], Jun Suzuki[1,2]
[1]Tohoku University, [2]RIKEN, [3]Future Corporation, [4]Machine Learning Solutions

Abstract

We participated in the constrained (open-weight) track of the WMT 2026 General MT Task for the English-to-Japanese and Simplified Chinese-to-Japanese directions. Our system builds on Marco-MT-Algharb, a strong open-weight translation model from the previous WMT General MT task. We further fine-tuned this model to generate more natural Japanese translations. To this end, we constructed a **C**ontext-**A**ware **JA**panese **L**inguistic **A**cceptability dataset (CAJALA). During annotation, human annotators were shown multiple paraphrased variants of a segment within its document-level context and asked to select the most natural one. We then back-translated the CAJALA dataset to create parallel data for Direct Preference Optimization (DPO). We also trained the model to perform fill-in-the-middle (FIM) translation, in which the model reconstructs target-side segments from their surrounding document. At inference, we use Minimum Bayes Risk (MBR) decoding, followed by selective FIM post-editing of low-quality segments.

1 Introduction

This paper describes RFMT's submission to the WMT 2026 General MT Shared Task (Kocmi et al., 2026). We participated in the constrained (open-weight) track for the En→Ja and Zh→Ja directions. Our system features two key components: (1) we constructed CAJALA, a dataset that provides context-aware human preferences for specific segments within Japanese documents, and further trained our model on it; and (2) we explored fill-in-the-middle (FIM) translation for post-editing.

In document-level translation, a translation that appears natural at the single-segment level may still fail to maintain the document's anaphora, style, terminology, or discourse flow (Maruf et al., 2021; Lyu et al., 2021). Colloquial and non-standard expressions also affect naturalness, especially in domains such as news, social media, and literature (Michel and Neubig, 2018; Fujii et al., 2020; Le et al., 2021; Karpinska and Iyyer, 2023). Furthermore, translating technical terms and proper nouns requires not only contextual naturalness but also consistency (Hasler et al., 2018; Dinu et al., 2019).

To improve the naturalness of Japanese translations, we collected human preferences for Japanese expressions within their document-level context and used them to train the model. Furthermore, we investigated the effectiveness of FIM translation for correcting local errors generated during standard left-to-right decoding.[1]

2 System Overview

Figure 1 provides an overview of our system. Our system builds on Marco-MT-Algharb (Wang et al., 2025b,a). We fine-tuned this base model to specialize in Japanese translation and to support FIM translation. Furthermore, we trained it to accept auxiliary information via prompting, such as the source domain, target styles (e.g., polite or plain forms), and glossary entries.

Our inference pipeline consists of four stages:
Stage 1: Preprocessing. We identify the domain, segment the text, and parse inputs in JSON or HTML formats.
Stage 2: Translation. We translate the document segment by segment using MBR decoding.
Stage 3: FIM Post-editing. We detect and re-translate low-quality segments using FIM.
Stage 4: Post-processing. We reconstruct the document and restore its original format.

3 CAJALA Dataset

CAJALA is a dataset that annotates the acceptability of segments within Japanese documents.

[1]Our model is publicly available at https://huggingface.co/tohoku-nlp/RFMT.

Proceedings of the Eleventh Conference on Machine Translation, pages 1344–1371
October 28-29, 2026 ©2026 Association for Computational Linguistics

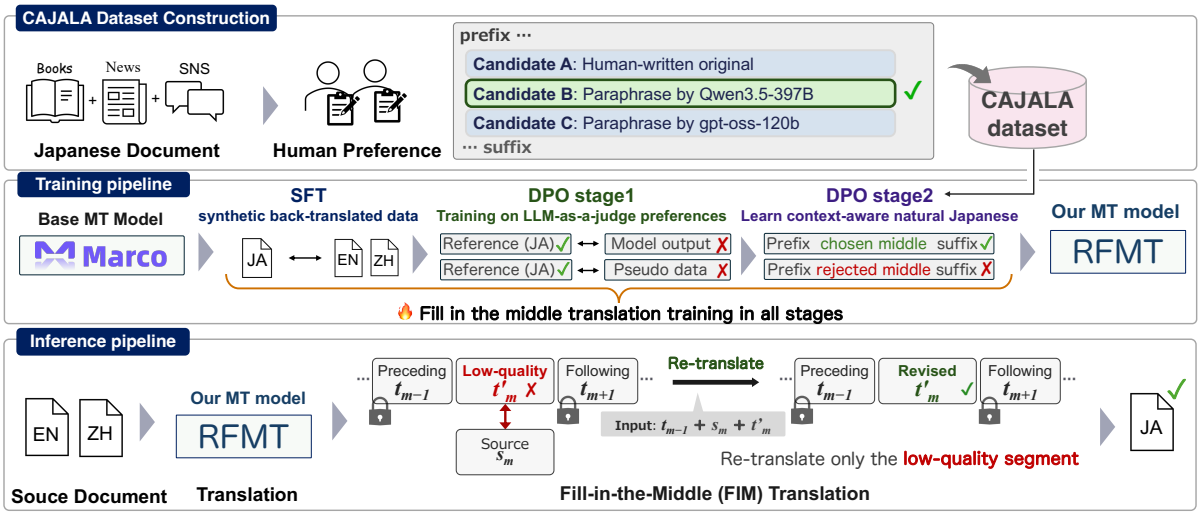

Figure 1: Overview of CAJALA dataset construction, RFMT training with SFT and DPO, and inference with FIM post-editing of low-quality segments.

Specifically, we collect human preferences based on the naturalness and consistency of Japanese expressions within document-level contexts. Each sample consists of a Japanese document and three variants of one segment: the original segment from an existing corpus and two paraphrases generated by LLMs. To construct the dataset, we used crowdsourcing. We showed human annotators three segment variations (the two paraphrases and the original text) alongside their surrounding context. We then asked the annotators to select the variation that best fits the overall flow of the document.

We obtained the original Japanese documents from a subset of the ja_fineweb-2 split in the llm-jp-corpus-v4 (LLM-jp, 2025). We used an LLM to classify these documents by domain and retained samples identified as News, SNS, or Literature, as described in Appendix B.1. We used an LLM-as-a-judge to identify segments that allowed meaningful alternative expressions and retained them for crowdsourcing, as described in Appendix A.3.

3.1 Paraphrase Generation and Human Annotation

We generated candidate segments by instructing LLMs to paraphrase the original Japanese expressions into more natural alternatives. We used `Qwen3.5-397B-A17B-FP8` (Qwen Team, 2026) and `gpt-oss-120b` (OpenAI, 2025) for this task. These two generated paraphrases and the original segment formed the three candidates. See Appendix A.2 for more details.

During human annotation, we presented these three segment variations with the same surrounding context and asked annotators to select the most acceptable segment. We instructed them to consider the acceptability of each candidate and its consistency in style and terminology with the surrounding context. Appendix A.4 outlines our quality control measures for the annotation process.

Number of Valid Annotators per Sample		
Item	Count	Proportion
1 annotator	125	1.70%
2 annotators	1,537	20.93%
3 annotators	5,683	77.37%
Candidate Selections		
Candidate	Selection Count	Selection Rate
`human`	6,741	33.29%
`Qwen`	7,111	35.12%
`gpt-oss`	6,396	31.59%

Table 1: CAJALA dataset statistics on the number of samples by valid annotator count and the selection counts and rates for each candidate.

3.2 Dataset Statistics

We collected 7,345 unique samples, each evaluated by up to three annotators. Table 1 shows the statistics of the selected candidates and the number of annotators. We calculated these selection rates based on a total of 20,248 valid responses in the public release. Appendix A.5 details the sample counts per domain.

4 Synthetic Dataset

We constructed synthetic parallel data to adapt the model for translation into Japanese.

4.1 Back-Translation Dataset

Except for the WMT 2025 test set (Kocmi et al., 2025), we constructed all parallel training data by back-translating (Sennrich et al., 2016) Japanese monolingual corpora. For these monolingual corpora, we used the llm-jp-corpus-v4 (LLM-jp, 2025), alongside transcriptions from the Japanese subsets of CC Audio (Asai et al., 2025) and YODAS2 (Li et al., 2023). Specifically, we constructed the parallel dataset as follows:

Domain Identification. An LLM predicted the domain of each document in the llm-jp-corpus-v4. We retained samples classified as News, Literature, or SNS, following the domains focused on in the previous year's General MT Track.

Noise Removal. An LLM identified and corrected speech recognition errors in the video transcriptions (CC Audio, YODAS2).

Back-translation. An LLM back-translated these preprocessed corpora into English or Chinese.

Cleaning. We applied rule-based filtering using source-target n-gram overlap, language identification, and document length. We evaluated translation quality and validity using an LLM-as-a-judge.

Terminology Extraction. We extracted bilingual terminology pairs from the translated sentences. We used these pairs later for glossary prompting and dictionary creation.

Style Identification. We classified the target Japanese text into the polite style (*desu/masu*) or the plain style (*da/dearu*). We used this classification later to train the model for style prompting.

For the FIM training samples, we performed two additional processing steps:

Target-side Segmentation. We conservatively divided the target documents into roughly sentence-level segments, merging overly short segments with adjacent ones to prevent unnatural splits.

Source-side Segment Alignment. An LLM identified the source-side segment that corresponded to the randomly selected target-side segment.

See Appendix B for details on the specific prompts and LLMs we used, along with the token counts of the final dataset.

4.2 Glossary Construction

We constructed a dictionary of terms specifically for glossary prompting. It consists of a large number of word-level parallel pairs, primarily targeting low-frequency and domain-specific terms. To build this glossary, we applied multi-stage rule-based and LLM-based filtering to the parallel pairs extracted during the dataset construction process. See appendix E for more details.

5 Model Adaptation

As our base model, we used Marco-MT-Algharb (Wang et al., 2025a,b), which ranked first in the En→Ja constrained track of the WMT 2025 General MT Task (Kocmi et al., 2025). Starting from this model, we first conducted supervised fine-tuning (SFT) on synthetic translation data to specialize it for Japanese translation and train it to follow prompting strategies. We then performed direct preference optimization (DPO) (Rafailov et al., 2023) using additional synthetic preference data and CAJALA to improve translation quality. In parallel, we included FIM training, in which the model reconstructs a missing target-side segment from its surrounding context.

5.1 Supervised Fine-tuning

Starting from Marco-MT-Algharb, we performed additional training using the back-translation dataset constructed in § 4.1. Appendix D provides the SFT training configuration.

5.2 Preference Learning

We applied DPO in two stages.

Stage 1. We trained the model on a mixture of three types of preference data. For **LLM-as-a-judge preferences**, we used `gemma-4-31B-it` (GemmaTeam, 2026) to compare the output of a model under development with the reference provided in the dataset. We defined the translation judged as higher quality by the LLM as the chosen response, and the other as the rejected response. For **synthetic repetition**, we artificially added repetition to the reference and used the modified version as the rejected response and the original reference as the chosen response. This data was intended to reduce repetitive outputs. For **synthetic untranslated content**, we replaced part of the reference suffix with the corresponding suffix from the source text. The modified version was used as the rejected response, and the original reference

was used as the chosen response. This data was intended to reduce untranslated source content in the model output. See Appendix D for more details.

Stage 2. We trained the model on the back-translated CAJALA dataset. We constructed preference pairs from samples with a majority candidate and excluded samples without a majority. To construct the training pairs, we back-translated the chosen responses into English and Chinese as the source sentences. See Appendix D.1 for details.

5.3 FIM Translation Training

Inspired by FIM training (Bavarian et al., 2022), we use FIM to enable localized post-editing while preserving document context. During training, we omit a target-side segment and train the model to reconstruct it from the corresponding source-side segment and both preceding and following target-side context. Unlike standard left-to-right decoding, which only conditions on preceding target context (Ding et al., 2021), FIM translation allows the model to revise a specific segment using context on both sides without regenerating the surrounding translation. Figure 2 shows the input-output format used for FIM translation training. For each training sample, we randomly selected one segment from the source document and enclosed it with the special tokens `<|fim_start|>` and `<|fim_end|>`. We then provided the target document in which the target-side segment corresponding to the selected source-side segment was replaced with `<|fim_skip|>` as the first **Assistant:** response. The omitted target-side segment was provided as the second **Assistant:** response. At inference time, we replace the target-side segment to be retranslated with `<|fim_skip|>` and provide the resulting incomplete target document together with the source document. We then prompt the model to generate the missing target-side segment. This format trains the model to reconstruct a missing target-side segment from its surrounding target-side context.

6 Inference

6.1 Preprocessing

For each source document, we perform three preprocessing steps: segment splitting (parsing structured input), domain identification, and style identification. These steps enable segment-level translation and allow us to incorporate auxiliary information into prompts. See Appendix C for details.

Input-Output format example

```
Human: Please translate the following text into
japanese.If a domain, glossary or style is
specified, please take it into account when
translating.Also, please skip translating the parts
enclosed by <|fim_start|> and <|fim_end|>.

Domain: News
Style: だ・である調
Glossary: - semiconductor => 半導体
---
Source text:
```
The Ministry of Economy, Trade and Industry
expanded chip subsidies.<|fim_start|>The framework
covers half of factory costs.<|fim_end|>The measure
strengthens supply chains.
```
<|im_end|>
Assistant:
経済産業省は半導体への補助金を拡大した。
<|fim_skip|>この措置はサプライチェーンを強化する。
<|im_end|>
Human: Please translate the skipped parts.
Assistant:
<|fim_bos|>この制度は建設費の半額を補助する。<|im_end|>
```

Figure 2: Input-output format for the trained model.

6.2 Translation

We translate each source document at the segment level. During translation, we use the following prompting and decoding strategies.

Context-Aware Incremental Translation. To maintain consistency within each document, we use context-aware incremental translation (Kudo et al., 2024). We translate the document segment by segment, starting from the first segment. When translating the second and subsequent segments, we provide the preceding source-side segments together with the current source-side segment as input. On the target side, we forcefully decode the hypothesis segments generated up to that point and ask the model to generate the continuation. We repeat this process until all segments have been translated. When the input exceeds the model's context length, we remove the oldest context first.

Prompting with Auxiliary Information. We provide three types of auxiliary information in the prompt. **Domain prompting:** We provide the domain information assigned during preprocessing in the prompt. **Style prompting:** We provide the style information assigned during preprocessing in the prompt. **Glossary prompting:** If the source document contains a term included in the glossary, we add the corresponding translation pair to the prompt. To match source-side glossary terms against the source document, we use longest non-overlapping substring matching.

	Model Adaptation		Inference		Score ↑				
	SFT	DPO	MBR	FIM	JSON	News	Social	Speech	Overall
`Base`					$87.9_{\pm 4.6}$	$48.1_{\pm 9.6}$	$55.7_{\pm 5.3}$	$65.5_{\pm 3.4}$	$62.9_{\pm 2.9}$
`SFT`	✓				$56.9_{\pm 13.5}$	$44.4_{\pm 5.5}$	$46.5_{\pm 6.5}$	$59.0_{\pm 4.0}$	$55.2_{\pm 3.2}$
`DPO-1`	✓	✓			$85.0_{\pm 11.7}$	$68.2_{\pm 9.8}$	$68.8_{\pm 6.1}$	$73.0_{\pm 3.6}$	$72.2_{\pm 2.9}$
`DPO-2`	✓	✓			$77.1_{\pm 8.6}$	$66.6_{\pm 7.8}$	$66.3_{\pm 5.9}$	$72.9_{\pm 3.6}$	$71.1_{\pm 2.8}$
`DPO-2+MBR`	✓	✓	✓		$86.9_{\pm 8.3}$	$67.4_{\pm 7.3}$	$69.6_{\pm 6.0}$	$72.3_{\pm 3.5}$	$71.9_{\pm 2.9}$
`RFMT`	✓	✓	✓	✓	$86.9_{\pm 8.3}$	$67.4_{\pm 7.3}$	$70.4_{\pm 5.8}$	$72.3_{\pm 3.5}$	$72.1_{\pm 2.8}$
`Base`					–	$12.6_{\pm 5.5}$	$26.5_{\pm 4.4}$	$49.5_{\pm 4.6}$	$38.4_{\pm 3.7}$
`SFT`	✓				–	$67.3_{\pm 6.9}$	$41.5_{\pm 5.1}$	$47.4_{\pm 4.5}$	$48.9_{\pm 3.3}$
`DPO-1`	✓	✓			–	$73.5_{\pm 8.0}$	$48.0_{\pm 5.6}$	$63.9_{\pm 4.3}$	$61.5_{\pm 3.4}$
`DPO-2`	✓	✓			–	$76.4_{\pm 6.7}$	$44.8_{\pm 5.3}$	$62.8_{\pm 3.7}$	$60.4_{\pm 3.1}$
`DPO-2+MBR`	✓	✓	✓		–	$78.4_{\pm 6.3}$	$44.8_{\pm 5.8}$	$63.5_{\pm 3.9}$	$61.2_{\pm 3.3}$
`RFMT`	✓	✓	✓	✓	–	$78.4_{\pm 6.3}$	$44.8_{\pm 5.8}$	$63.5_{\pm 3.9}$	$61.2_{\pm 3.3}$

Table 2: Translation llm-as-a-judge scores for each system configuration by domain. The upper and lower panels show results for En→Ja and Zh→Ja, respectively. Values are reported as the mean ± 95% CI.

MBR Decoding. We generated 32 candidates with a temperature of 0.8 and applied MBR decoding (Kumar and Byrne, 2004; Freitag et al., 2022). We used xCOMET-Lite (Larionov et al., 2024) as the utility function for MBR.

6.3 FIM Post-editing

After completing the main translation process, we detect clearly low-quality segments (e.g., untranslated). We then post-edit the detected segments using FIM translation. The post-editing procedure consists of the following steps.

Error Detection. We use rule-based criteria to detect clearly low-quality segments. Specifically, we calculate the n-gram overlap between each hypothesis and its source-side segment to detect untranslated content, and check whether the hypothesis contains excessive repetition. Segments that meet either criterion are selected for post-editing. Preliminary experiments motivated this conservative selection strategy, as detailed in Appendix G.

FIM Translation. We retranslated the detected segments using the FIM format. We used MBR decoding as the decoding algorithm. We generated 32 candidates with a temperature of 1.2, `top_p` of 0.95, and `top_k` of 32, and applied MBR decoding. For the MBR utility function, we used a combined score from xCOMET-Lite (Larionov et al., 2024) and MetricX-24 XL (Juraska et al., 2024).

6.4 Post-processing

We reconstructed the translated results into full documents. We performed two steps: (1) we reverted samples parsed from JSON or HTML back to their original input formats, and (2) we concatenated the segments to form the final documents.

7 Experiments

7.1 Evaluation Metrics

We evaluated the systems on the WMT 2026 blind test set. We evaluated translation quality using an LLM-as-a-judge (Kocmi and Federmann, 2023). Specifically, we employed Gemini 3.1 Pro Preview (DeepMind, 2026), the latest model in the Gemini family. We selected this model because its predecessor, Gemini 2.5 Pro (Comanici et al., 2025), achieved the best performance in the automatic evaluation of the WMT 2025 Metrics Shared Task (Lavie et al., 2025). We provided the source text and the hypothesis as input and asked the model to assign an integer score from 0 to 100. We then averaged the scores over all samples to obtain the mean score for each language pair and system. See Appendix F for more details.

7.2 Results and Analysis

Table 2 reports the LLM-as-a-judge evaluation results for `Base`, `SFT`, `DPO-1`, `DPO-2`, `DPO-2+MBR`, and our final `RFMT` system. These configurations are cumulative, with each configuration adding a training or inference component to the preceding one. Specifically, `SFT` adds supervised fine-tuning to `Base`; `DPO-1` and `DPO-2` successively add DPO stages 1 and 2; `DPO-2+MBR` applies MBR decoding to `DPO-2`; and `RFMT` further applies selective FIM post-editing to `DPO-2+MBR`. For `RFMT`, we re-evaluated the four En→Ja outputs changed by FIM post-editing and recomputed the overall

	Source Excerpt	Before	After
(a) `Base` → `SFT` **Zh→Ja** `24382-jpn_Jpan` (*Score*: 0 → 66)	我仿佛可以像想象。… 我们俩就抱着一个饮水机到处走。	我仿佛可以像想象。… 我们俩就抱着一个饮水机到处走。	僕は想像できるような。… 僕らふたりはウォーターサーバーを抱えて歩き回る。
(b) `SFT` → `DPO-1` **En→Ja** `24158-jpn_Jpan` (*Score*: 20 → 95)	Topics have been - different types of product managers and the basic skill sets in developing requirements	Topics have been - different types of product managers and the basic skill sets in developing requirements	話題は- プロダクトマネージャーの種類と要件開発における基本スキル
(c) `DPO-1` → `DPO-2` **En→Ja** `24182-jpn_Jpan` (*Score*: 45 → 85)	… Okay. And you're going uphill. I mean, this is really good stuff here. I know that you're probably feeling it a bit. I'm struggling to keep you. But I'm not in my best form. I think the hill is very challenging. The hill is always challenging. …	… はい。そして上りですね。いや、これ、めっちゃいいもんやで。ちょっと気持ち入ってるんやろうな、わかるけど。キープに苦労してる。でも、僕は本調子じゃないからね。坂はすごく厳しいと思うよ。坂はいつも厳しい。…	… はい。そして上りですね。いや、これホントいいもんなんですよ。ちょっと感じてるんだろうな、ってのは分かってるんですけど。キープに苦労してます。でも、僕は本調子じゃないです。坂はすごく厳しいと思います。坂はいつだって厳しいです。…
(d) `DPO-2` → `DPO-2+MBR` **En→Ja** `24296-jpn_Jpan` (*Score*: 50 → 95)	… But just remember, we own the rights. Once you crack it, we still want this to work. Need this to work.	… でもな、権利はこっちにあるからな。お前が解明したら、やっぱりこっちはこれに期待してるんだよ。これに期待。	… ただ、権利は当社が所有しているってことはお忘れなくな。解明できたら、やっぱりこれ、稼働させてほしいからな。稼働させないと。
(e) `DPO-2+MBR` → `RFMT` **En→Ja** `24247-jpn_Jpan` (*Score*: 33 → 50)	they send an email saying "thanks for sending through the design"	they send an email saying "thanks for sending through the design"	「デザインを送って下さってありがとうございます」というメールを送る

Table 3: Qualitative examples for successive system configurations. We show a Zh→Ja example for (a) and En→Ja examples for (b)–(e). Score changes are shown below each sample ID. Red marks errors or untranslated content, and blue marks the improved expressions.

score while retaining the original scores for all other samples. Table 3 provides representative examples for each successive system configuration. Appendix H presents additional examples for the other language direction. In the following, we discuss the impact of each stage.

SFT. Compared with `Base`, `SFT` decreased the En→Ja score from 62.9 to 55.2, while increasing the Zh→Ja score from 38.4 to 48.9. We attribute the Zh→Ja improvement to the fact that `Base` had not been trained for Zh→Ja translation, whereas `SFT` introduced training data for this direction. As shown by example (a) 24382-jpn_Jpan in Table 3, `Base` left most of the Chinese source untranslated while `SFT` produced a Japanese translation, increasing the score from 0 to 66.

DPO Stage 1. Compared with `SFT`, `DPO-1` increased the scores by 17.0 points for En→Ja and 12.6 points for Zh→Ja. `DPO-1` included preference pairs in which target-language outputs were used as chosen responses and source-language outputs as rejected responses. As shown by the En→Ja example in Table 3, this training reduced untranslated source content in the output.

DPO Stage 2. From `DPO-1` to `DPO-2`, the scores changed little for both En→Ja and Zh→Ja. However, the qualitative example in Table 3 shows a substantial improvement in style consistency. In example 24182-jpn_Jpan, a translation containing an informal Kansai dialect and a mixture of plain and polite forms was revised into a consistent polite style, increasing the score from 45 to 85.

MBR Decoding. Applying MBR decoding to `DPO-2` resulted in little change in the average scores. However, the qualitative example in Table 3 shows a substantial local improvement. In example 24296-jpn_Jpan, "Need this to work" was revised from "これに期待" (*kore ni kitai*; "counting on this") to "稼働させないと" (*kadō sasenai to*; "need to get this working"), making the intended sense of necessity clearer and increasing the score from 50 to 95.

FIM Post-editing. FIM post-editing was eventually applied to only four En→Ja samples; the score increased for three samples and remained unchanged for one. No FIM post-editing was applied to Zh→Ja samples. The qualitative example in Table 3 shows that FIM post-editing made local corrections, including the translation of previously untranslated content.

8 Conclusion

We described the RFMT's submission to the WMT 2026 General Machine Translation Task for En→Ja and Zh→Ja translation. We adapted Marco-MT-Algharb through SFT, DPO, and FIM training using the human-annotated CAJALA dataset. Our results show that SFT, DPO, MBR decoding, and selective FIM post-editing somehow improve document-level Japanese translation.

Acknowledgments

We would like to thank the members of the Tohoku NLP Group for their cooperation and feedback throughout the course of this research. This work was supported by Moonshot R&D Grant Number JPMJMS2011-35 (fundamental research) and by JSPS KAKENHI Grant Number JP25KJ0615 and JP26H02511. A part of the computation was carried out using the computer resource offered under the category of General Projects by Research Institute for Information Technology, Kyushu University

Contributions

Ryosuke Matsuda led the project and conducted crowdsourcing for the construction of the CAJALA dataset, cleaned the synthetic dataset, performed additional training of the language models, and conducted post-evaluation.
Keito Kudo constructed the synthetic dataset, performed additional training of the language models, and developed the inference pipeline.
Ryo Fujii, **Takumi Ito**, and **Makoto Morishita** provided advice on the crowdsourcing process and guidance on the overall development direction.
Jun Suzuki supervised and coordinated the entire project.

References

Kohei Asai, Issa Sugiura, Wataru Nakata, Shuhei Kurita, Shinnosuke Takamichi, Satoshi Ogawa, and Ryuichiro Higashinaka. 2025. Building large-scale speech and audio dataset from common crawl. In *Acoustical Society of Japan 2025 Autumn Meeting*.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle. *Preprint*, arXiv:2207.14255.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Google DeepMind. 2026. Gemini 3.1 Pro: Model Card.

Shuoyang Ding, Marcin Junczys-Dowmunt, Matt Post, and Philipp Koehn. 2021. Levenshtein training for word-level quality estimation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6724–6733, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Ryo Fujii, Masato Mita, Kaori Abe, Kazuaki Hanawa, Makoto Morishita, Jun Suzuki, and Kentaro Inui. 2020. PheMT: A phenomenon-wise dataset for machine translation robustness on user-generated contents. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5929–5943, Barcelona, Spain (Online). International Committee on Computational Linguistics.

GemmaTeam. 2026. Gemma 4 technical report. *Preprint*, arXiv:2607.02770.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. In *Proceedings of the Eighth Conference on Machine Translation*, pages 419–451, Singapore. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov,

Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakougna, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025. Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets. In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413, Suzhou, China. Association for Computational Linguistics.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Sergey Dukanov, Anton Dvorkovich, Mark Fishel, Markus Freitag, Samuel Frontull, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Stan Kharevich, Philipp Koehn, Zheng Li, Jean Maillard, Christof Monz, Alexander Murauski, and 12 others. 2026. Findings of the WMT26 general machine translation shared task: Contrastive dynamic human evaluation at scale. In *Proceedings of the Eleventh Conference on Machine Translation*. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Keito Kudo, Hiroyuki Deguchi, Makoto Morishita, Ryo Fujii, Takumi Ito, Shintaro Ozaki, Koki Natsumi, Kai Sato, Kazuki Yano, Ryosuke Takahashi, Subaru Kimura, Tomomasa Hara, Yusuke Sakai, and Jun Suzuki. 2024. Document-level translation with LLM reranking: Team-J at WMT 2024 general translation task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 210–226, Miami, Florida, USA. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Daniil Larionov, Mikhail Seleznyov, Vasiliy Viskov, Alexander Panchenko, and Steffen Eger. 2024. xCOMET-lite: Bridging the gap between efficiency and quality in learned MT evaluation metrics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21934–21949, Miami, Florida, USA. Association for Computational Linguistics.

Alon Lavie, Greg Hanneman, Sweta Agrawal, Diptesh Kanojia, Chi-Kiu Lo, Vilém Zouhar, Frederic Blain, Chrysoula Zerva, Eleftherios Avramidis, Sourabh Deoghare, Archchana Sindhujan, Jiayi Wang, David Ifeoluwa Adelani, Brian Thompson, Tom Kocmi, Markus Freitag, and Daniel Deutsch. 2025. Findings of the WMT25 shared task on automated translation evaluation systems: Linguistic diversity is challenging and references still help. In *Proceedings of the Tenth Conference on Machine Translation*, pages 436–483, Suzhou, China. Association for Computational Linguistics.

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.

Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. 2023. Yodas: Youtube-oriented dataset for audio and speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

LLM-jp. 2025. LLM-jp Corpus v4. `https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v4`.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Xinglin Lyu, Junhui Li, Zhengxian Gong, and Min Zhang. 2021. Encouraging lexical translation consistency for document-level neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3265–3277, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A survey on document-level neural machine translation: Methods and evaluation. *ACM Comput. Surv.*, 54(2).

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

NVIDIA. 2025a. Nemo megatron bridge. `https://github.com/NVIDIA-NeMo/Megatron-Bridge`. GitHub repository.

NVIDIA. 2025b. Nemo rl: A scalable and efficient post-training library. `https://github.com/NVIDIA-NeMo/RL`. GitHub repository.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: your language

model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS), Red Hook, NY, USA. Curran Associates Inc.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Hao Wang, Linlong Xu, Heng Liu, Yangyang Liu, Xiaohu Zhao, Bo Zeng, Liangying Shao, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025a. Beyond single-reward: Multi-pair, multi-perspective preference optimization for machine translation. *arXiv preprint arXiv:2510.13434*.

Hao Wang, Linlong Xu, Heng Liu, Yangyang Liu, Xiaohu Zhao, Bo Zeng, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025b. Marco large translation model at wmt2025: Transforming translation capability in llms via quality-aware training and decoding. In *Proceedings of the Tenth Conference on Machine Translation (WMT 2025)*, pages 587–593, Suzhou, China. Association for Computational Linguistics.

A CAJALA Construction Details

A.1 Annotation Platform and Crowd Workers

Platform. We collected all human annotations through Yahoo! Crowdsourcing, a commercial crowdsourcing platform operated by Yahoo Japan Corporation. Figure 3 shows an example of the annotation interface.

Crowd Workers. A total of 2,048 unique workers participated in the annotation project. We recruited general users registered on the platform and did not require any specific professional qualifications. Workers were progressively filtered based on quality criteria across annotation rounds, as described in Appendix A.4.

Compensation. For each task, defined as selecting the most appropriate candidate from three options, we set the reward based on the expected time required to complete the task. The unit price was set so that the effective hourly wage exceeded the average minimum wage in Japan. Across the entire annotation project, we paid workers a total of 219,376 JPY, equivalent to approximately 1,371 USD at the time of the experiments.

A.2 Paraphrase Candidate Generation

We used `Qwen3.5-397B-A17B-FP8` (Qwen Team, 2026) and `gpt-oss-120b` (OpenAI, 2025) to generate paraphrased candidates. Figure 4 shows the prompt used for this task.

A.3 Candidate Filtering

We evaluated the validity of each MIDDLE segment using an LLM-as-a-judge with the prompt shown in Figure 5. We retained only samples that satisfied all of the following prompt-specified conditions: `decision=keep`, `issue_types=[]`, a score of 5 for all quality criteria, and `risk=1`.

A.4 Annotation Quality Control

To ensure annotation quality, we progressively selected workers over multiple rounds. In Round 0, we prepared six gold questions with clear correct answers. Each worker was presented with five questions: two randomly selected gold questions and three regular questions. We ranked workers by their accuracy on the gold questions and selected the top 911 workers to participate in Round 1.

Figure 3: Example of the annotation interface showing document-level context and multiple candidates.

Worker Selection after Round 1. We used accuracy on the gold questions to select workers from Round 0 to Round 1. From Round 1 onward, we selected workers for the next round based on a worker quality score computed from their responses and annotation behavior. The quality score consisted of four measures: reading time, agreement with other annotators, viewing rates for the candidates and document text, and candidate dwell time.

Reading Time. For each question, we denote the reading time in seconds by s and define the reading-time score r as

$$r = \begin{cases} 0, & s < 20, \\ s/40, & 20 \leq s < 40, \\ 1, & 40 \leq s. \end{cases} \tag{1}$$

Thus, responses completed in less than 20 seconds receive a score of 0, those taking at least 40 seconds receive a score of 1, and scores between these thresholds increase linearly with reading time. The worker-level score is the average over all questions answered by that worker.

Agreement with Other Annotators. For each question, we define the agreement rate as the proportion of other workers who selected the same candidate. When no other worker was available for comparison, we set the agreement rate to 0.5. For each round, we first averaged the agreement rates over all non-gold questions answered by each worker and then averaged these worker-level scores

across participating workers. Table 4 reports the resulting agreement rates.

Candidate and Document Viewing. For each question, let v denote the candidate-viewing rate and b the document-viewing rate. We define the viewing score q as

$$q = \frac{v + b}{2}. \tag{2}$$

The worker-level viewing score is the average of q over all questions answered by that worker.

Candidate Dwell Time. Let d denote the dwell time for a candidate in milliseconds and m the reference dwell time. We define the dwell-time score h as

$$h = \min\left(\frac{d}{m}, 1\right). \tag{3}$$

We used $m = 400$ ms as the reference dwell time. The worker-level dwell-time score is the average of h over all candidates viewed by that worker.

Overall Quality Score. Because the four measures have different scales, we ranked workers separately for each measure and converted the rankings to scores ranging from 0 to 100, with 100 assigned to the highest-ranked group and 0 to the lowest-ranked group. Workers with the same value for a measure received the same rank score.

Let P_{iR}, P_{iA}, P_{iQ}, and P_{iH} denote the rank scores of worker i for reading time, agreement, viewing, and dwell time, respectively. We define the final worker quality score as

$$\text{PoliteScore}_i = \frac{P_{iR} + P_{iA} + P_{iQ} + P_{iH}}{4}. \tag{4}$$

At the end of each round, we ranked workers in descending order of PoliteScore_i and selected the highest-ranked workers according to the number required for the next round. Workers who were not selected did not participate in subsequent rounds.

A.5 Domain Composition of Data Sources

Table 5 shows the domain composition of the 7,345 samples in the final CAJALA dataset after excluding samples with no valid annotations.

B Synthetic Dataset Details

B.1 Construction Process

This section details the construction process of our synthetic dataset.

Round	Workers	Agreement
0	2,048	80.22%
1	911	82.02%
2	437	82.06%
3	230	84.53%
4	109	84.56%
5	59	85.31%
6	32	85.94%

Table 4: Statistics for each round of human annotation. Agreement is computed only over non-gold samples with exactly three valid responses.

Domain	Samples
News	2,212
SNS	3,645
Literature	1,492

Table 5: Domain composition of the final CAJALA dataset.

Domain Classification. We used gemma-4-26B-A4B-it (GemmaTeam, 2026) to classify the domains of the llm-jp-corpus-v4 samples across 19 categories. Figure 6 shows the prompt used for this task.

Noise Removal. We used Qwen3-Swallow-30B-A3B-RL-v0.2[2] to identify and correct speech recognition errors in the video transcriptions (CC Audio, YODAS2). Figure 7 shows the prompt used for this task.

Back-translation. We back-translated the Japanese documents into English and Chinese using several LLMs: gemma-4-26B-A4B-it, gemma-4-31B-it, Qwen3.6-35B-A3B, and Qwen3.5-122B-A10B-FP8. Figure 8 and 9 show the prompts used for this task.

Cleaning. For rule-based filtering, we applied the rules detailed in Table 6. For the LLM-as-a-judge, we used gemma-4-26B-A4B-it and gemma-4-31B-it (GemmaTeam, 2026). The models first evaluated the translations for predefined error types, such as omissions, incorrect target language, and named entity errors. They then made a final binary judgment on the overall translation quality. Figure 10 shows the prompt used for this evaluation.

Terminology Extraction. We extracted bilingual terminology pairs from the parallel corpus

[2] https://huggingface.co/tokyotech-llm/Qwen3-Swallow-30B-A3B-RL-v0.2

Filter	Rejection criterion
Language mismatch	Discard samples whose source or target is not identified as the expected language of the pair.
Boilerplate text	Remove samples whose source or target contains a boilerplate phrase from a curated list of web-scraping artifacts (citation notices, paywall notices, copyright and RSS footers, site navigation, browser names), with a fixed probability.
Target length	Discard samples whose target exceeds the allowed character limit.
Newline ratio	Remove samples whose proportion of line breaks exceeds the allowed threshold.
Punctuation gap	Discard samples whose Japanese text contains a punctuation-free span longer than the allowed length.
Target number ratio	Remove samples whose proportion of numeric characters exceeds the allowed threshold.
Number consistency	Discard samples in which a number written in Arabic digits on one side does not appear on the other side.
n-gram overlap	Remove samples whose source and target share more character n-grams than allowed.
Repetition	Discard samples whose character n-gram repetition rate exceeds the allowed threshold within a local window.

Table 6: Rule-based filters and their rejection criteria applied to the synthetic parallel data. A sample is discarded if it meets any of the criteria above.

for use in glossary prompting and dictionary construction. For this extraction, we used gemma-4-26B-A4B-it and gemma-4-31B-it (GemmaTeam, 2026). Figure 11 and 12 show the prompts used for this task.

Style Identification. We split each document into sentences based on line breaks and Japanese punctuation marks (e.g., "。", "！", "？"). We then classified the ending of each sentence using string matching. Specifically, we defined ending patterns for the polite style (e.g., "です", "ます", "ました", "ません", "ください") and the plain style (e.g., "だ", "である", "だった", "だろう", "ない"). We counted a sentence as polite if it matched a polite pattern, and as plain if it matched only a plain pattern. We then assigned the overall document the style of its majority sentence type. To ensure reliability, we only assigned a style if the majority style appeared in at least two sentences, and the minority style appeared in at most one sentence. We left documents unclassified if they failed to meet this condition.

	Tokens
SFT	758,507,883
DPO Stage1 (Synthetic)	1,230,093
DPO Stage1 (LLM judge)	70,135,828
DPO Stage2 (CAJALA)	8,631,489

Table 7: Number of training tokens used in each stage.

Source-side Segment Alignment. We randomly selected one segment generated during the Target-side Segmentation phase and identified its corresponding source-side segment. For this alignment, we used gemma-4-26B-A4B-it and Qwen3.6-35B-A3B. Figure 13 and 14 show the prompts used for this task.

B.2 Dataset Size

Table 7 shows the number of tokens in the final dataset.

C Preprocessing Details

As preprocessing steps, we performed segment splitting, domain classification, and style selection.

Segment splitting. We first determine whether the document can be parsed as JSON or HTML, that is, whether it contains structured text. For structured documents, we parse the document using the corresponding parser and treat each extracted text span as a single segment. For documents not identified as structured text, we first split the input document into sentences using spaCy (Honnibal et al., 2020). We then merge adjacent sentences to form segments of at least 130 characters for En→Ja and 50 characters for Zh→Ja, avoiding unnaturally short segments.

Domain classification. For each sample in the blind test set, instructions on the translation style and format are provided together with multimodal resources. We use these instructions and resources to assign a domain according to the following four rules. **Video:** If the multimodal resources include a video, assign the "Streaming/Video/Audio" domain. **News:** If the instructions contain the word `news`, ignoring case, assign the "News" domain. **SNS:** If the instructions contain keywords related to social media, assign the "SNS" domain. **Other:** Otherwise, leave the domain unspecified.

Style selection. For samples classified as News, we instructed the model through the prompt to use

SFT	
Train Steps	11,573
Batch Size	65,536 tokens
Learning Rate Schedule	Cosine
Warmup Steps	550
Max Learning Rate	3×10^{-5}
Min Learning Rate	1×10^{-7}
Optimizer	AdamW (Loshchilov and Hutter, 2019) $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-8}$, weight_decay $= 0.01$
Gradient Clip	1.0
Implementation	`Megatron Bridge` (NVIDIA, 2025a)
DPO Stage 1	
Train Steps	11,573
Batch Size	16 preference pairs (32 sequences)
Maximum Sequence Length	16,384 tokens
Learning Rate Schedule	Linear warmup followed by constant LR
Warmup Steps	20
Warmup Initial LR	1×10^{-11}
Learning Rate	5×10^{-7}
Optimizer	Adam (Kingma and Ba, 2015) $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$,
Gradient Clip	1.0
DPO Coefficient β	0.1
Log-probability Reduction	Sum over response tokens
Implementation	`NeMo RL` (Megatron backend) (NVIDIA, 2025b)
DPO Stage 2	
Train Steps	574
Batch Size	16 preference pairs (32 sequences)
Maximum Sequence Length	16,384 tokens
Learning Rate Schedule	Linear warmup followed by cosine decay
Warmup Steps	20
Warmup Initial LR	1×10^{-11}
Maximum Learning Rate	5×10^{-7}
Minimum Learning Rate	0
Optimizer	Adam (Kingma and Ba, 2015) $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, weight_decay $= 0.01$
Gradient Clip	1.0
DPO Coefficient β	0.1
Log-probability Reduction	Sum over response tokens
Implementation	`NeMo RL` (Megatron backend) (NVIDIA, 2025b)

Table 8: Hyperparameters used during model training.

the plain declarative style. For all other samples, no style was specified.

D Training Configuration

Table 8 lists the training hyperparameters.

D.1 DPO Dataset Construction

LLM-as-a-Judge Preferences. To generate LLM-as-a-judge preferences, we used `gemma-4-31B-it` (GemmaTeam, 2026) to assign preferences between the output of an in-development model and the reference in the synthetic dataset. Figure 15 shows the prompt used for this evaluation.

CAJALA Preference Pairs. For each CAJALA sample, if one candidate received at least two votes, we treated it as the chosen candidate and each of the other two candidates as a rejected candidate, yielding two preference pairs. Samples without a majority candidate, such as those with a 1:1:1 vote split, were excluded from DPO training. For each pair, the document containing the chosen segment was back-translated into English and Chinese and used as the source for both the chosen and rejected responses.

E Glossary Extraction Details

As described in § 4.2, we built the glossary by applying multi-stage filtering to the bilingual terminology pairs extracted during the dataset construction process. First, we used gemma-4-26B-A4B-it (GemmaTeam, 2026) as an LLM-as-a-judge to evaluate the validity of the extracted pairs. Figure 16 shows the prompt used for this evaluation. Next, using the wordfreq (Speer, 2022) library, we retained low-frequency words with a Zipf frequency of 4.0 or lower for English, and 4.5 or lower for Chinese. We then applied rule-based filters to remove identical source-target pairs, ambiguous cases where a single source word mapped to multiple targets, and terms containing boundary punctuation. Furthermore, we matched the extracted terms against the source sentences at the token level for English and the character level for Chinese. When matches overlapped, we applied the leftmost-longest rule to retain the longest possible matching terms in the glossary. The final En↔Ja and Zh→Ja glossaries contained 215,000 and 90,633 pairs, respectively.

F LLM-as-a-Judge Evaluation

During the LLM-as-a-judge evaluation, we provided the model with the source language, target language, source-side segment, and target-side segment, and instructed it to score the translation quality on an integer scale from 0 to 100. Figure 17

shows the full prompt used for this evaluation. We excluded two samples for which the Gemini API returned a `PROHIBITED CONTENT` error.

G FIM Post-editing Strategy

During system development, we also explored strategies for applying FIM post-editing to a broader set of segments, rather than only to the segments detected by the rules described in § 6.3. Specifically, we performed quality estimation for each segment using MetricX (Juraska et al., 2024) and experimented with post-editing the segment with the lowest score in each document. However, we found that the improvements in translation quality achieved by FIM post-editing were limited. We therefore adopted a conservative strategy of correcting only translations that were clearly of low quality, aiming to ensure that post-editing reliably improved translation quality. As a result, only four translation outputs were modified by FIM post-editing. Table 9 shows examples in which translation quality degraded when post-editing was applied more aggressively based on MetricX scores.

H Additional Qualitative Examples

Table 10 provides qualitative examples for the language direction omitted from each comparison in Table 3. For `Base` → `SFT`, we show an En→Ja example, whereas the remaining comparisons show Zh→Ja examples. Table 11 shows three examples in which FIM post-editing changed the hypotheses of `DPO-2+MBR`.

I Total System Parameters

Table 12 lists the parameter counts for each model used in our system. Our total parameter count of 18.7B falls below the 20B threshold, satisfying the constraints of the Open weight (constrained) track.

	Source text	Before FIM post-editing	After FIM post-editing
(a) **En→Ja** 24296-jpn_Jpan	So this is my carry on. And then this bad boy, are. You gonna check it? I don't know. I can't decide. It's pretty heavy. But it's so smooth.	だからこれ、私のキャリーオン。それでこのやつは、アレ。預けるかい？ わかんないなー。決められないなー。結構重たいしなー。でもめちゃくちゃ滑るよ。	これ、俺のキャリーオン。そしてこの野郎、アレだ。チェック入れる？ わかんないな。決められないな。結構重いよな。でもめっちゃ滑る。
(b) **En→Ja** 24247-jpn_Jpan	Look, they gave us a little beach bag, a hat. Um. Everything I never want to forget. And some things I do. Oh my God, so cute. So cute.	ほら、小さなビーチバッグと帽子をもらった。えっと。忘れたくないこと全部。そして、忘れたいこともある。やだー、かわいい。かわいい。	ほら、ビーチバッグと帽子の小さいのをもらった。えっと。私が絶対に忘れたくないもの全部。それから、忘れたくないものも。あらやだ、かわいい。かわいい。

Table 9: Examples of samples whose translation quality degraded due to FIM post-editing. Blue marks the parts that were correctly translated in the original translations, while red marks obvious errors introduced by FIM post-editing.

Lang	Sample ID	Source Excerpt	Before	After
(a) Base → SFT				
En→Ja	24255-jpn_Jpan (*Score*: 50 → 100)	… All right. I didn't realize this involved frying, and I don't think I did this correctly because the recipe did not tell me how much breadcrumbs to add. … It's not really frying. Really confused. I just let the zucchini drop to the bottom. …	… はい、いいえ。これに揚げ物が含まれていると知りませんでしたし、レシピにパン粉の量が書かれていなかったので、正しい作り方ではなかったと思います。… これって、揚げ物じゃないんですよね。本当におかしいです。ただ、ズッキーニを底に落としちゃいました。…	… はい。揚げるって知らなかったし、パン粉の量が書いてなかったから、これ正しく出来てないと思う。… 揚げてるわけじゃないし。めっちゃ混乱。ズッキーニを下に落とすだけ。…
(b) SFT → DPO-1				
Zh→Ja	24141-jpn_Jpan (*Score*: 0 → 85)	… 打一次用接力的方式，嗯，这样子也比较安全啊，…	… 打一次用接力的方式，嗯，这样子也比较安全啊，…	… 一回打ってリレーで、うん、これも安全だな、…
(c) DPO-1 → DPO-2				
Zh→Ja	24218-jpn_Jpan (*Score*: 33 → 75)	… 有来自江门的家长得悉是次澳大校庆开放日活动，虽未能及时办理港澳通行证签证，但特意组团到横琴粤澳深度合作区校区过渡期办学点查询入学情况，充分体验了过渡期办学点作为展示澳大办学风采的窗口之一的重要性。…	… 今回の澳大創立記念オープンハウスの開催を知り、香港から来たという江門市の保護者は、香港マカオ往来通行証のビザ取得が間に合わなかったものの、わざわざグループで横琴・粤澳深度協力区キャンパス移行期開設地を訪れ、入学状況を問い合わせるなど、移行期開設地が澳大の教育の魅力を発信する窓口の一つとしての重要性を存分に体感したようだ。…	… 江門から来たある保護者は、今回の澳大創立記念日のオープンハウスイベントの開催を知り、香港・マカオ往来通行証のビザ取得が間に合わなかったものの、わざわざグループを組んで横琴粤澳深度協力区キャンパス移行期開設地を訪れ、入学状況を問い合わせ、移行期開設地が澳大の運営の様子を伝える窓口の一つとしての重要性を存分に体感した。…
(d) DPO-2 → DPO-2+MBR				
Zh→Ja	24406-jpn_Jpan (*Score*: 50 → 100)	基本上就是温泉组成的山洞。论为了拍照有多牺牲。很冷的。…	基本的に温泉の洞窟。写真撮るのにどれだけ論が犠牲になってるか。寒いよ。…	基本的に温泉の洞窟。写真撮るためにどれだけ犠牲になったかって話。寒いんだよ。…

Table 10: Additional qualitative examples for the language direction not shown in Table 3. Score changes are shown below each sample ID. Red marks errors or untranslated content, and blue marks the improved expressions.

Sample ID	Source Excerpt	Before	After
DPO-2+MBR → RFMT			
24028-jpn_Jpan (*Score*: 98 → 98)	"2257 Exempt"	2257 Exempt	2257 エグゼンプト
24154-jpn_Jpan (*Score*: 55 → 65)	project move out, spring break relay	project move out, spring break relay	project 引っ越し、春休みラリー
24383-jpn_Jpan (*Score*: 60 → 70)	i. ii.	i. ii.	i。ii。

Table 11: Additional examples for which the hypotheses produced by DPO-2+MBR and RFMT differ. A representative example is shown in Table 3(e). The LLM-as-a-judge scores before and after post-editing are shown below each sample ID. Source texts are excerpts relevant to the changes; red highlights issues before post-editing, and blue highlights the corresponding expressions after post-editing.

Model	Parameters
Translation Model	14,768,307,200
XCOMET-lite	283,731,987
MetricX-24 XL	3,742,619,648
Total	18,794,658,835

Table 12: Number of parameters for each model and the total.

Paraphrase Prompt

```
# 文章の一部分を文脈に合わせて自然な表現に書き換える

与えられた文章全体の文脈を踏まえ、指定された「対象部分」を、前後の流れに調和したより自然な日本語表現へと書き換えてください。

## 入力

以下の 3 つが与えられます。

- **文章全体**：対象部分を含む文章の全体。文脈・文体・語調を判断するための材料です。
- **対象部分**：文章全体の中に含まれる、書き換える対象となる箇所。
- **対象部分の翻訳**：書き換える対象部分を翻訳したもの。意味を保持しつつ、自然な日本語表現にするための参考情報です。


## 書き換えの方針

### 1. 意味を保持する
- 対象部分が伝えている内容・主張・ニュアンスを変えない。
- 情報を勝手に追加したり削除したりしない。
- 参考情報として与えられた翻訳結果を尊重し、意味が変わらない範囲で自然な表現にする。

### 2. 前後の文脈と調和させる
- **文体**：文章全体に合わせて、敬体（です・ます調）か常体（だ・である調）かを統一する。
- **語彙・トーン**：硬さ／柔らかさ、専門性のレベル、フォーマルさを周囲と揃える。
- **つながり**：直前の文からの流れ、直後の文への流れが滑らかになるようにする。
- **用語の一貫性**：文章全体で使われている表記・専門用語・固有名詞の言い回しに合わせる。

### 3. 自然さを高める
- 翻訳調・直訳調や、不自然な言い回しを、こなれた日本語に直す。
- 冗長な表現や回りくどい言い方を簡潔にする。
- 助詞の誤りや、係り受けの不明瞭さを正す。
- 同じ語句の不要な繰り返しを避ける。
- 読点の位置や語順を整え、読みやすくする。

### 4. 範囲を守る
- 書き換えるのは **対象部分のみ**。前後の文には手を加えない。
- 対象部分の長さや位置づけを大きく逸脱しない（必要以上に増減させない）。
- すでに十分自然で文脈にも合っている場合は、無理に変えず元の表現をそのまま出力してよい。

## 出力形式

- 書き換え後の **対象部分のテキストのみ** を、次のように ```txt``` のコードブロックで囲んで出力する。
- コードブロックの外側には、前置き・説明・理由などを一切付けない。
- 文章全体を再掲しない。

```txt
（ここに書き換え後の対象部分）
```

タスク
上記の方針に従って、与えられた文章全体と対象部分をもとに、対象部分を自然な表現に書き換えてください。

```
【文章全体】
<<<|FULL_TEXT|>>>

【書き換え対象部分】
<<<|MIDDLE_TEXT|>>>

【対象部分の翻訳】
<<<|MIDDLE_TRANSLATED|>>>
```
```

Figure 4: Prompt used for paraphrasing

MIDDLE Suitability Judgment Prompt

```
# MIDDLE 妥当性判定タスク

text infilling 翻訳の preference アノテーション用データを作成します。
日本語文書から選ばれた MIDDLE が、**パラフレーズ候補を作って人間が「前後文脈に最も自然に入るもの」を選ぶタスク**に向いているか
判定してください。

このタスクでは、パラフレーズ候補は生成しません。
MIDDLE 自体を `keep` / `human_review` / `drop` のどれにするかだけを判定してください。

## 判定で見ること

次の観点を総合して判定してください。

1. MIDDLE が自然な日本語か。
2. PREFIX / SUFFIX と話題・文体・接続が自然につながるか。
3. 意味を保ったまま、自然な言い換え候補を 2 個以上作れそうか。
4. 候補間で「どれが文脈に最も自然か」の差が出そうか。
5. SOURCE と明らかに矛盾していないか。

SOURCE は補助情報です。
日本語側の自然さと前後文脈との整合性を最優先してください。
SOURCE が空、または対応判断が難しい場合、明らかな矛盾がなければ `source_consistency` は 3 にしてください。

## keep したい例

- 日本語として自然で、ノイズや UI 文言が混ざっていない。
- PREFIX / SUFFIX と自然につながる。
- 文体、語順、接続、助詞、文末、語彙などに言い換え余地がある。
- 意味・数量・固有名詞を保ったまま、2〜4 個程度の自然な候補を作れそう。

## drop したい例

- MIDDLE が空、null、または欠損している。
- URL、数字、商品名、コード、メンション、ハッシュタグ、絵文字、記号が中心。
- ログイン案内、引用 UI、エラーメッセージ、HTML、メタ情報などのノイズが中心。
- 定型文すぎて、自然な言い換え候補を複数作れない。
- PREFIX / SUFFIX とつながらない、または文体が合わない。
- 前後文脈を見ても意味や指示対象が曖昧すぎる。
- SOURCE と MIDDLE の対応が明らかに怪しい。
- 言い換えると意味ズレしやすい。

## スコア

各スコアは 1〜5 の整数で付けてください。
`risk` だけは、値が高いほど悪いスコアです。

- `fluency`: MIDDLE 単体の日本語としての自然さ。
  - 1: 不自然・破綻・ノイズ中心
  - 3: 読めるが少し不安
  - 5: 自然な日本語
- `context_fit`: PREFIX / SUFFIX との自然なつながり。
  - 1: つながらない
  - 3: おおむねつながるが不安
  - 5: 自然につながる
- `source_consistency`: SOURCE との整合性。
  - 1: 明らかに怪しい
  - 3: 判断に迷う
  - 5: 問題なさそう
- `paraphrase_potential`: 自然な言い換え候補を複数作れそうか。
  - 1: ほぼ作れない
  - 3: 1〜2 個なら作れそう
  - 5: 複数作りやすい
- `annotation_value`: 人間が preference を付けるデータとして価値があるか。
  - 1: 価値が低い
  - 3: 人手確認が必要
  - 5: 価値が高い
- `risk`: ノイズ、曖昧さ、意味ズレ、固有名詞・数字中心などのリスク。
  - 1: リスクが低い
  - 3: 少し不安
  - 5: リスクが高い
```

Figure 5: Prompt used to assess whether a middle segment was suitable for constructing CAJALA datasets (Part 1 of 2).

MIDDLE Suitability Judgment Prompt (continued)

```
## 判定ラベル

- `keep`: そのまま採用してよい。
  - 目安: `fluency >= 4`, `context_fit >= 4`, `paraphrase_potential >= 4`, `annotation_value >= 4`, `risk <= 2`
- `human_review`: 判断に迷うため、人手確認に回す。
  - 日本語や文脈は悪くないが、SOURCE 対応・曖昧さ・意味ズレ・ノイズに不安がある。
- `drop`: 除外する。
  - 空、ノイズ中心、定型的すぎる、文脈に合わない、言い換え価値が低い、SOURCE 対応が怪しい。

## issue_types

問題がある場合だけ、以下から少数選んでください。
問題がない場合は `[]` にしてください。

- `missing_middle`: 空、null、欠損
- `noise_or_non_text`: URL、UI 文言、コード、HTML、メタ情報などのノイズ
- `entity_or_symbol_heavy`: 数字、固有名詞、商品名、メンション、絵文字、記号が中心
- `context_or_style_mismatch`: PREFIX / SUFFIX と文脈・文体が合わない
- `source_mismatch`: SOURCE との対応が怪しい
- `low_paraphrase_value`: 言い換え余地や preference の価値が低い
- `ambiguous_or_drift_risk`: 曖昧すぎる、または意味ズレしやすい
- `other`: その他

## 出力形式

必ず以下の JSON オブジェクトのみを出力してください。
Markdown のコードブロックや説明文は出力しないでください。
`decision` は `"keep"`, `"human_review"`, `"drop"` のいずれか 1 つです。
MIDDLE が空、null、欠損、またはプレースホルダ未置換に見える場合は、`has_middle` を `false` にし、`decision` は `"drop"` にしてください。

{
  "has_middle": true,
  "scores": {
    "fluency": 1,
    "context_fit": 1,
    "source_consistency": 1,
    "paraphrase_potential": 1,
    "annotation_value": 1,
    "risk": 1
  },
  "decision": "keep",
  "issue_types": [],
  "reason": "判定理由を日本語で簡潔に記載"
}

## 入力

【ドメイン】
<<<|DOMAIN|>>>

【原文 SOURCE】
<<<|SOURCE_TEXT|>>>

【対応する SOURCE 側セグメント（空の場合は無視）】
<<<|ALIGNED_TRANSLATION|>>>

【日本語 文脈】
=== PREFIX（参照のみ）===
<<<|PREFIX|>>>

=== MIDDLE（判定対象）===
<<<|MIDDLE|>>>

=== SUFFIX（参照のみ）===
<<<|SUFFIX|>>>
```

Figure 5: Prompt used to assess whether a middle segment was suitable for constructing CAJALA datasets (continued).

Domain Classification Prompt

```
# 文書ドメイン分類タスク

あなたは、与えられた日本語テキストの文書ドメインを分類するアシスタントです。

## タスク

テキストを読み、当てはまるカテゴリを 1 つだけ選んでください。

1. **SNS への投稿**: X(Twitter)、Facebook、Instagram、TikTok などのソーシャルメディアへの投稿。
2. **掲示板**: 5ちゃんねる、なんJ、ふたばちゃんねる、Redditなどの掲示板への書き込み。匿名性が高く、スレッド形式のレスのやり取りや独特のスラング・アスキーアートを含むことが多い。
3. **Q&A**: Yahoo!知恵袋、教えて!goo、OKWAVE、Stack Overflowなどの質問・回答サイトのコンテンツ。質問文と回答文がセットになっているか、明確に質問・回答のいずれかと判別できる形式。
4. **対話**: 会話・チャットログ・インタビュー・座談会・対談・脚本など、複数話者のやり取りで構成されたテキスト。発言者の交代や話者ラベルが明確な形式。
5. **ニュース**: 新聞社・通信社・テレビ局などの報道機関による事実報道記事。「いつ・どこで・誰が・何を」を客観的に記述する報道スタイル。
6. **プレスリリース**: 企業・団体が公式に発表する報道発表資料。新製品発表、業績発表、人事発表、提携発表など、発信元が明示された公式アナウンス。
7. **ブログ・コラム**: 個人・ライターによるブログ記事、コラム、オピニオン記事、エッセイ、note記事など。書き手の主観・経験・意見が中心となる読み物。
8. **文学・web小説**: 小説、詩、戯曲、随筆、web小説、ライトノベル、二次創作など、創作的・物語的なテキスト。地の文と会話文を含む叙述形式。
9. **百科事典・解説**: Wikipedia、コトバンク、ニコニコ大百科などの百科事典記事、用語解説、概念の中立的な説明文。事実を網羅的・体系的に整理した記述。
10. **教育・学術**: 教科書、講義資料、学術論文、研究レポート、学習教材、問題集、参考書など、教育・学術目的で書かれたテキスト。
11. **技術関連・マニュアル・ソースコード**: 技術文書、APIリファレンス、製品マニュアル、取扱説明書、技術ブログ、ソースコード、設定ファイル、プログラミングのコード例・解説など。
12. **商品・EC**: Amazon、楽天市場などのECサイトの商品ページ、商品説明、スペック表、カタログなど、商品の紹介・販売を目的とするテキスト。
13. **レビュー・口コミ**: 商品、サービス、店舗、作品、施設などに対するユーザーによるレビュー、口コミ、評価コメント、感想投稿。
14. **企業・団体・行政**: 企業・団体・行政機関の公式サイトのコーポレートページ、会社概要、IR情報、行政からの告知・通達・広報など、組織の公式情報。
15. **求人・不動産・旅行などの情報サービス**: 求人情報、不動産物件情報、旅行・宿泊・観光情報、グルメ情報、イベント情報など、情報サービスサイトに掲載される定型的な情報。
16. **医療・法律・金融など専門情報**: 医療、法律、税務、金融、保険など、専門分野の情報提供記事・解説・Q&A・コラム。専門家による執筆または監修を伴うことが多い。
17. **広告・SEO・スパム**: 広告コピー、アフィリエイト目的の記事、SEO狙いの薄いコンテンツ、まとめサイトのコピペ記事、スパム的な宣伝文章など。
18. **アダルトサイト**: 成人向けコンテンツを扱うサイトのテキスト。アダルト動画・画像サイトの紹介文や作品説明、出会い系・風俗関連サイト、官能小説、性的内容を含む広告・口コミなど。
19. **その他**: 上記のいずれにも該当しない文書、または複数カテゴリの判別が困難な文書。

## 出力形式

- 出力は `番号. カテゴリ名` の形式で 1 行のみ出力してください。
- それ以外の内容は出力しないでください。

出力例:
```
5. ニュース
```

---

## 入力
次のテキストを分類してください:
```txt
<<<|TEXT|>>>
```
```

Figure 6: Prompt used for domain classification.

Noise Removal Prompt

```
# 日本語文字起こしテキストの校正

## 役割

あなたは日本語の文字起こし（音声認識）テキストを校正する編集者です。話し言葉ならではの言い回しや雰囲気はそのまま残しつつ、フィラーや言い淀みを取り除き、認識ミスを直し、文章として自然に読めるように手直しを行います。

## タスク

入力された文字起こしテキストに対して、以下のルールに従って校正してください。話し言葉を書き言葉に変換するのではなく、あくまで話し言葉のまま読みやすく整えることが目的です。

## 修正ルール

1. フィラーや言い淀み（「えー」「あのー」「まあ」「えっと」「なんか」「そのー」など）を削除する。
2. 言い直しや無意味な繰り返し（同じ語句のダブり、言いかけて止めた断片など）を整理し、自然につながるようにする。
3. 音声認識による誤変換（同音異義語の取り違え、語の聞き取り間違いなど）を、文脈から判断して正しい表記に修正する。
4. 句読点を適切に補い、読点・句点を整える。また、不自然な句読点の付け方を修正する。余計な空白やスペースを削除する。
5. 文の区切りや改行を整え、内容のまとまりごとに段落を分ける。
6. 明らかな脱字・衍字（よけいな文字）や、意味の通らない崩れた箇所を最小限に整える。

## 維持すること（書き換えない）

- 話し言葉特有の言い回し・口調・語尾（「〜だよね」「〜なんですよ」「〜じゃん」など）はそのまま残す。
- 口語的な省略や倒置など、意味の伝わる表現はそのままにする。
- 方言・くだけた表現を標準的な書き言葉に直さない。

## 厳守事項

- 発言の意味内容を変えない。要約・追加・独自解釈の挿入をしない。
- 話者の意図や主張、話し方の個性を勝手に補完・脚色・標準化しない。
- 元のテキストに存在しない情報を付け加えない。
- 判断に迷う箇所は、無理に修正せず元の表現をそのまま残す。

## 出力形式

修正後のテキストのみを、以下のように ```txt``` コードブロックで囲んで出力してください。コードブロックの前後に説明・注釈・前置きを一切加えないこと。

```txt
（修正後のテキスト）
```

ただし、入力内容が日本語でない・空欄等の問題があった場合には、

```txt
N/A
```
とだけ出力してください。

入力テキスト

以下のテキストを校正してください。

```txt
<<<|TEXT|>>>
```
```

Figure 7: Prompt used for noise removal.

Back Translation Prompt for Ja→EN

```
# Translation into English
Please translate the following text into natural English.

## Translation Instructions
- Produce natural, fluent, and easy-to-read English.
- Wrap the output in a ```txt``` code fence.
- Do not output any additional explanation other than the translation itself.

## Input / Output Example
### Source Text
```txt
半導体は AI の開発に必要不可欠です。
```

Translated Output (English)
```txt
Semiconductors are essential for the development of AI.
```

Task (Source Text)
```txt
<<<|SOURCE_TEXT|>>>
```
```

Figure 8: Prompt used for Japanese-to-English back-translation.

Back Translation Prompt for Ja→Zh

```
# 翻译为简体中文
请将以下文本翻译为自然的简体中文。

## 翻译要求
- 输出自然、流畅、易于阅读的简体中文。
- 将输出内容用 ```txt``` 代码块包裹。
- 除译文本身外，请勿输出任何额外的说明。

## 输入 / 输出示例
### 原文
```txt
半導体は AI の開発に必要不可欠です。
```
译文（简体中文）
```txt
半导体对于人工智能的开发是必不可少的。
```
任务（原文）
```txt
<<<|SOURCE_TEXT|>>>
```
```

Figure 9: Prompt used for Japanese-to-Chinese back-translation.

Translation-pair Quality Judgment Prompt

```
# 翻訳ペア品質判定
あなたは翻訳データセットをクリーニングするための厳格な品質判定者です。
次の原文と翻訳文を比較し、翻訳ペアをデータセットに採用してよいか判定してください。

## 判定基準
- `OK`: 原文の主要な意味が翻訳文に十分反映されている。
- `NG`: データセットから除外すべき重大な問題がある。

【NG となる重大な問題とエラータイプ（error_types）の対応】
- `omission`: 原文の重要情報が抜けている
- `mistranslation`: 原文と完全に異なる意味に訳している
- `entity_number_error`: 数字、固有名詞、日付などを誤っている
- `addition`: 原文にない重要情報を追加している
- `wrong_language`: 翻訳文が指定された対象言語ではない
- `truncated`: 文章が途中で切れている
- `translation_failure`: 翻訳に失敗している
- `other`: 上記以外の重大な欠陥がある

## 出力形式
必ず JSON オブジェクトのみを出力してください。
{
  "reason": "判定の理由",
  "label": "OK または NG",
  "error_types": []
}

## 判定対象データ
【言語情報】
- 原文言語: <<<|SOURCE_LANG|>>>
- 翻訳先言語: <<<|TARGET_LANG|>>>
【テキスト】
=== 原文 ===
<<<|SOURCE_TEXT|>>>
=== 翻訳文 ===
<<<|TRANSLATION_TEXT|>>>
```

Figure 10: Prompt used to evaluate the quality of translation pairs after back-translation.

Terminology Extraction Prompt for En→Ja

```
# Technical Terminology Translation Pair Extraction Task

## Role
You are an expert in extracting technical terminology translation pairs from parallel Japanese and English texts.

## Task
The **Japanese** text and **English** text presented below are in a parallel translation relationship. Carefully
read both texts and extract the **technical terminology translation pairs** contained within them.

## Extraction Rules

### 1. Faithfulness
- Extract the expressions appearing in the texts **exactly as they are written**.
- Do **not perform any processing** such as normalization of notation, completion of okurigana, conversion of
character types, adjustment of word endings, or restoration to base forms.
- Preserve the exact surface form as actually used in the texts.

### 2. Basis of Correspondence
- Extract only terms that **actually exist in both the Japanese and English texts** and correspond to each other as
translations.
- Do not supplement or fabricate translation pairs based on general knowledge or speculation.
- Terms that appear in only one of the languages are excluded from extraction.

### 3. Extraction Targets
- Target technical terms (technical vocabulary, academic terms, industry-specific concept words) and proper nouns.
- Exclude general everyday words, function words, grammatical elements, numerical values, and general-purpose verbs
and adjectives.
- Extract up to five pairs of technical terms and proper nouns.

### 4. Duplicate Handling
- If the same translation pair appears in multiple locations, output it only once.

## Output Format

**The response must be exactly one valid JSON object conforming to the following format.**

- Do not include any text other than the JSON (no preamble, explanation, annotation, conclusion, blank lines,
comments, etc.).
- If the extraction result is zero, output `{"term_pairs": []}`.

```json
{
  "term_pairs": [
    {"ja": "機械学習", "en": "machine learning"},
    {"ja": "教師あり学習", "en": "supervised learning"},
    {"ja": "ラベル付きデータ", "en": "labeled data"}
  ]
}
```

Input

Japanese
```
<<<|JAPANESE_TEXT|>>>
```

English
```
<<<|ENGLISH_TEXT|>>>
```
```

Figure 11: Prompt used to extract bilingual terminology pairs from the Japanese-English parallel corpus.

Terminology Extraction Prompt for Zh→Ja

```
# Technical Terminology Translation Pair Extraction Task

## Role
You are an expert in extracting technical terminology translation pairs from parallel Chinese (Simplified) and
Japanese texts.

## Task
The **Chinese (Simplified)** text and **Japanese** text presented below are in a parallel translation relationship.
Carefully read both texts and extract the **technical terminology translation pairs** contained within them.

## Extraction Rules

### 1. Faithfulness
- Extract the expressions appearing in the texts **exactly as they are written**.
- Do **not perform any processing** such as normalization of notation, completion of okurigana, conversion of
character types, adjustment of word endings, or restoration to base forms.
- Preserve the exact surface form as actually used in the texts.

### 2. Basis of Correspondence
- Extract only terms that **actually exist in both the Chinese and Japanese texts** and correspond to each other as
translations.
- Do not supplement or fabricate translation pairs based on general knowledge or speculation.
- Terms that appear in only one of the languages are excluded from extraction.

### 3. Extraction Targets
- Target technical terms (technical vocabulary, academic terms, industry-specific concept words) and proper nouns.
- Exclude general everyday words, function words, grammatical elements, numerical values, and general-purpose verbs
and adjectives.
- Extract up to five pairs of technical terms and proper nouns.

### 4. Duplicate Handling
- If the same translation pair appears in multiple locations, output it only once.

## Output Format

**The response must be exactly one valid JSON object conforming to the following format.**

- Do not include any text other than the JSON (no preamble, explanation, annotation, conclusion, blank lines,
comments, etc.).
- If the extraction result is zero, output `{"term_pairs": []}`.

```json
{
  "term_pairs": [
    {"zh": "机器学习", "ja": "機械学習"},
    {"zh": "监督式学习", "ja": "教師あり学習"},
    {"zh": "标注资料", "ja": "ラベル付きデータ"}
  ]
}
```

Input

Chinese (Simplified)
```
<<<|CHINESE_TEXT|>>>
```

Japanese
```
<<<|JAPANESE_TEXT|>>>
```
```

Figure 12: Prompt used to extract bilingual terminology pairs from the Japanese-Chinese parallel corpus.

Source-side Segment Prompt for En → Ja

```
# Parallel-Text Extraction Task

## Role
You are an **extractor** that, given a **Japanese–English parallel translation pair**, precisely pulls out the
English text corresponding to a specified passage. Your job is **not translation**, but the **identification and
extraction of English text that already exists**.

## Task Overview
1. As input, you are given the **Japanese version** of a document and its corresponding **English version** (a
parallel translation pair).
2. In addition, you are given **one segment** (a few consecutive sentences) taken from the Japanese version.
3. You must identify **which part of the English version** this Japanese segment corresponds to, and extract the
sentence(s) written in the English version **exactly as they appear**.

## Rules You Must Strictly Follow
- **Do not translate on your own.** Translating the Japanese segment into English is prohibited. You must instead
locate the text that already exists in the provided English version and output it by copying it verbatim.
- The English text you extract must be output **without altering the original wording, punctuation, capitalization,
or any other aspect of the source text** in the English version. You must not summarize, paraphrase, or edit it.
- Select a range of English text that covers the content of the Japanese segment **completely and without excess**.
If the corresponding English spans multiple sentences, extract the **entire consecutive range**.
- The sentence boundaries and ordering may differ between Japanese and English. Even if the word order or number of
sentences does not correspond one-to-one, extract the **consecutive range on the English side that corresponds in
meaning**.
- If the corresponding English text **does not exist** in the English version, output only `N/A`. Do not fabricate
English text by guessing.
- Output **only** the extracted English text (or `N/A`), enclosed in a code block beginning with ```` ```txt ```` .
Do not add any explanation, preamble, or supplementary remarks whatsoever.

## Input (Example)
### Japanese (full text)
```txt
私たちの会社は2010年に設立されました。設立当初は社員がわずか3名でした。現在では従業員数が500名を超えています。私たちは持続可能
な社会の実現を目指しています。
```
English (full text)
```txt
Our company was founded in 2010. At the beginning, we had only three employees. Today, our workforce exceeds 500
people. We aim to realize a sustainable society.
```
Japanese Segment to Extract
```txt
設立当初は社員がわずか 3 名でした。現在では従業員数が 500 名を超えています。
```
Output (Example)
```txt
At the beginning, we had only three employees. Today, our workforce exceeds 500 people.
```

Input (Actual)
Now, perform the task on the following input.

Japanese (full text)
```txt
<<<|TARGET_DOCUMENT|>>>
```
English (full text)
```txt
<<<|SOURCE_DOCUMENT|>>>
```
Japanese Segment to Extract
```txt
<<<|TARGET_SEGMENT|>>>
```
```

Figure 13: Prompt for Source-side Segment (En → Ja)

Source-side Segment Prompt for Zh → Ja

```
# 中日双语对照文本抽取任务

## 角色
你是一个**抽取器**。当给定一对**日文与中文的对照译文**时，你需要精确地抽取出与指定段落相对应的中文文本。你的任务**不是翻译**，
而是**识别并抽取出已经存在的中文文本**。

## 任务概述
1. 作为输入，你会得到某篇文档的**日文版本**，以及与之相对应的**中文版本**（两者互为对照译文）。
2. 此外，你还会得到从日文版本中截取出来的**一个片段**（连续的若干句子）。
3. 你需要判断该日文片段**对应于中文版本中的哪一部分**，并将中文版本中所写的句子**原封不动地**抽取出来。

## 必须严格遵守的规则
- **不得自行翻译。** 禁止将日文片段翻译成中文。你必须在所提供的中文版本中找到已经存在的文本，并将其逐字逐句、原样复制后输出。
- 抽取出的中文文本必须保持中文版本原文的样貌，**不得改动其字词、标点、书写形式等任何方面**。不得进行概括、改写或编辑。
- 所选取的中文文本范围应当**不多不少地**覆盖该日文片段的内容。如果对应的中文跨越多个句子，请抽取出**整个连续的范围**。
- 日文与中文的句子边界及顺序可能并不相同。即使语序或句子数量并非一一对应，也要抽取出**在语义上相对应的、位于中文一侧的连续范围
**。
- 如果中文版本中**不存在**与之相对应的中文文本，则只输出 `N/A`。不得凭猜测捏造中文文本。
- 输出**仅限于**抽取出的中文文本（或 `N/A`），并用以 ```txt 开头的代码块加以包裹。不得添加任何说明、前言或补充性的文字。

## 输入（示例）
### 日文（全文）
```txt
私たちの会社は2010年に設立されました。設立当初は社員がわずか3名でした。現在では従業員数が500名を超えています。私たちは持続可能
な社会の実現を目指しています。
```
中文（全文）
```txt
我们公司成立于 2010 年。成立之初只有三名员工。如今员工人数已超过 500 人。我们致力于实现可持续发展的社会。
```
待抽取的日文片段
```txt
設立当初は社員がわずか 3 名でした。現在では従業員数が 500 名を超えています。
```
输出（示例）
```txt
成立之初只有三名员工。如今员工人数已超过 500 人。
```

输入（实际数据）
现在，请针对以下输入执行该任务。

日文（全文）
```txt
<<<|TARGET_DOCUMENT|>>>
```
中文（全文）
```txt
<<<|SOURCE_DOCUMENT|>>>
```
待抽取的日文片段
```txt
<<<|TARGET_SEGMENT|>>>
```
```

Figure 14: Prompt for Source-side Segment (Zh → Ja)

Domain Classification Prompt

```
# Translation Quality Pairwise Evaluation

## Role
You are an expert evaluator of Japanese translation quality.

## Task
You are given a **source text** and **two Japanese translations** (Translation A and Translation B) of that source
text. Carefully compare the two translations and judge **which one is the better translation** of the source text.

## Evaluation Criteria
Assess each translation comprehensively from the following perspectives:

### 1. Adequacy
- How accurately and completely the meaning of the source text is conveyed.
- Penalize mistranslations, omissions, additions of information not present in the source, and unsupported
speculation.

### 2. Fluency

- Whether the Japanese reads naturally and grammatically.
- Penalize unnatural phrasing, grammatical errors, unnatural word order, awkward word choices, and
machine-translation-like expressions such as repeatedly including the subject where it would normally be omitted in
natural Japanese.

### 3. Terminology and Proper Nouns
- Whether technical terms and proper nouns are translated correctly and consistently.

### 4. Style and Register
- Whether the tone, formality, and register are appropriate for the source text.

## Judgment Rules
- Base your judgment **only on the source text and the two translations presented**. Do not rely on outside
information or speculation.
- Compare the two translations relatively and rate their relative quality on the **5-point scale** defined below.
- If both translations are of equal quality and there is no meaningful difference, judge it as a **tie**.

## Output Format

**The response must be exactly one valid JSON object conforming to the following format.**

- Do not include any text other than the JSON (no preamble, explanation, annotation, conclusion, blank lines,
comments, etc.).
- `verdict` must be one of the following five strings:
  - `"A_much_better"`: Translation A is clearly better.
  - `"A_slightly_better"`: Translation A is slightly better.
  - `"tie"`: The two translations are of equal quality.
  - `"B_slightly_better"`: Translation B is slightly better.
  - `"B_much_better"`: Translation B is clearly better.

```json
{
  "verdict": "A_much_better"
}
```

Input

Source
```
<<<|SOURCE_TEXT|>>>
```

Translation A
```
<<<|TRANSLATION_A|>>>
```

Translation B
```
<<<|TRANSLATION_B|>>>
```
```

Figure 15: Prompt used by the LLM-as-a-judge to assign preferences.

Glossary Filtering Prompt

```
# 対訳判定タスク

与えられた 2 つの異なる言語のフレーズが、互いに正しい翻訳（対訳関係）になっているかを判定してください。

## 判定基準

- 慣用句の自然な意訳は許容する。
- 意味の欠落や食い違い、固有名詞・数量・日時などの事実の齟齬、肯定/否定の逆転、部分訳や無関係な内容がある場合は **False**。
- **2 つのフレーズが同じ言語で書かれている場合は、翻訳が行われていないため、意味が一致していても必ず False とする。**

## 出力

`True` または `False` の 1 単語のみを出力してください。説明や補足は一切書かないでください。

## 入力

- **フレーズ A**: <<<|SOURCE_TERM|>>>
- **フレーズ B**: <<<|TARGET_TERM|>>>
```

Figure 16: Prompt used for glossary filtering.

LLM-as-a-Judge Scoring Prompt

```
Score the following translation from {source_lang} to {target_lang} on a scale from 0 to 100, where a score of 0
means a broken or poor translation; 33 indicates a flawed translation with significant issues; 66 indicates a good
translation with only minor issues in grammar, fluency, or consistency; and 100 represents a perfect translation in
both meaning and grammar.
Answer with only a whole number representing the score, and nothing else.

{source_lang} source text:
{source_seg}
{target_lang} translation:
{target_seg}
```

Figure 17: Prompt used by the LLM-as-a-judge to score translation quality.