Targeted Source Text Editing for Machine Translation: Exploiting Quality Estimators and Large Language Models

Hyuga Koretaka[†*] **Atsushi Fujita**[‡] **Tomoyuki Kajiwara**[†]
[†]Graduate School of Science and Engineering, Ehime University, Ehime, Japan
[‡]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
[†]{koretaka@ai., kajiwara@}cs.echime-u.ac.jp
[‡]atsushi.fujita@nict.go.jp

Abstract

To improve the translation quality of "black-box" machine translation (MT) systems, we focus on the automatic editing of source texts to be translated. In addition to the use of a large language model (LLM) to implement robust and accurate editing, we investigate the usefulness of targeted editing, i.e., instructing the LLM with a text span to be edited. Our method determines such source text spans using a span-level quality estimator, which identifies actual translation errors caused by the MT system of interest, and a word aligner, which identifies alignments between the tokens in the source text and translation hypothesis. Our empirical experiments with eight MT systems and ten test datasets for four translation directions confirmed the efficacy of our method in improving translation quality. Through analyses, we identified several characteristics of our method and that the segment-level quality estimator is a vital component of our method.

1 Introduction

In the last decade, the quality of machine translation (MT) outputs has been significantly improved as a result of the advancements of neural MT (NMT) and large language models (LLMs) and the accumulation of parallel data in the community. A number of new techniques for further improving translation quality, i.e., reducing translation errors, have been presented at conferences; however, proprietary MT services have tended to remain state of the art, presumably thanks to undisclosed technologies and massive in-house data. Such "black-box" systems are, in general, difficult to adapt for users' niche use cases in which texts with specific content domains or text styles are to be translated.

To obtain better translations using black-box MT systems, several strategies have been proposed. One such strategy is "pre-editing," i.e., editing given source texts to improve their translation by an MT system of interest. Although studies on automatic pre-editing have long been conducted (§2), two issues remain. First, existing methods have only limited editing ability. Various types of source text editing can potentially improve its translatability (Miyata and Fujita, 2017, 2021). However, in past studies, researchers have addressed only specific linguistic complexities or performed the regeneration of entire texts indiscriminately. Another issue is that researchers have performed pre-editing without referring to the actual translation generated by the MT system, despite the proven effectiveness of editing source text with reference to actual translation errors (Uchimoto et al., 2006; Resnik et al., 2010). Different MT systems have different error tendencies; thus, editing expressions that the MT system can translate well would result in new translation errors.

In this study, we automate "targeted source text editing" using (a) quality estimation models to determine what to edit on the basis of the translation errors caused by a target MT system and to search for a better translation and (b) LLMs to realize various types of editing as in Ki and Carpuat (2025). In our method (§3), a given source text is translated by the MT system, and errors in the output are identified by a span-level quality estimator. Then, using an LLM, our method edits the source text with a text span annotated as the source of the severest translation error. Guiding the editing process with a trigger error does not guarantee that the MT system can translate the edited text better (Miyata and Fujita, 2017, 2021). Thus, our method searches for a better translation by repeating text editing and MT, relying on a segment-level quality estimator.

Our empirical experiments with eight MT systems and ten test datasets for four translation directions confirmed the efficacy of our method in improving translation quality (§5). Our analyses

[*] This work was done during an internship of the first author at NICT.

Proceedings of the Tenth Conference on Machine Translation, pages 200–219
November 8-9, 2025

also revealed several characteristics of our method, including the diverse impact depending on the MT system and dataset, the necessity of improving the segment-level quality estimator, and controlled editing realized by tailored instruction (§6).

2 Previous Work

MT systems and services have gradually pervaded our lives, and have been incorporated into the human-centered translation production process adopted by translation/language service providers (ISO/TC37, 2017). Before they became sufficiently practical, researchers examined several approaches to human–MT interaction. Uchimoto et al. (2006) proposed the editing of source texts motivated by translation errors; the method was later named "targeted paraphrasing" by Resnik et al. (2010). Inspired by previous studies on targeted paraphrasing, Miyata and Fujita (2017, 2021) conducted manual investigations into the pre-editing strategy for exploiting black-box services based on statistical and neural MT. Through incrementally performing source text editing in four content domains referring to MT outputs, they found that most (80%–100%) of the source texts could eventually be edited so that they will lead to no translation errors and that human editors have performed diverse types of edits, not only limited to paraphrasing, that can improve translation quality.

Automatic "pre-editing" methods, which have been studied for three decades, can be classified into two groups. The first group focuses on specific linguistic phenomena that are difficult to translate, such as low frequency words, subject ellipsis, and long sentences, and avoids them relying on a set of rewriting rules based on morpho-syntactic information, corpus statistics, and neural language models (Shirai et al., 1993; Kim and Ehara, 1994; Yamaguchi et al., 1998; Shirai et al., 1998; Yoshimi, 2001; Mirkin et al., 2013; Štajner and Popovic, 2016; Štajner and Popović, 2018; Koretaka et al., 2023). However, each of these methods only covers a specific type of editing, among the diverse promising ones. Early methods are difficult to replicate for other source languages because of their heavy reliance on hand-crafted rules and resources.

Another line of research has attempted to regenerate entire source texts by regarding the task as monolingual translation and applying data-driven sequence-to-sequence decoding methods (Sun et al., 2010; Nanjo et al., 2012; Mirkin et al., 2013; Mehta et al., 2020). The performance of this approach is dominated by the characteristics and quantity of training parallel data. Since no parallel data have been specifically tailored for the purpose of pre-editing, except for small collections for manual analyses and evaluation, researchers have used monolingual parallel data that exhibit other monolingual tasks, such as text simplification and text revision, or synthetic parallel data generated by back-translating bilingual parallel data and round-trip translation of monolingual data. This approach has been proven effective for rule-based and statistical MT systems (Sun et al., 2010; Nanjo et al., 2012; Mirkin et al., 2013); in contrast, it does not necessarily work for NMT systems (Koretaka et al., 2023). Recently, Ki and Carpuat (2025) examined the utility of LLMs for source text editing. They compared several editing strategies and identified that the instruction for text simplification and selection based on quality estimation are effective.

Unlike manual investigations (Miyata and Fujita, 2017, 2021), both of the aforementioned groups of methods do not refer to the translation errors caused by the MT system of interest. Although a linguistically motivated pre-edit should be helpful in general, excessive editing of translatable expressions would introduce new translation errors.

3 Targeted Source Text Editing

Unlike existing "pre-editing" methods, we propose the editing of given source texts to avoid actual translation errors that the target MT system makes, i.e., the automation of the manual investigation process (Miyata and Fujita, 2017, 2021). Algorithm 1 shows the overall procedure of our method.

Given a source text src_0, our method first translates it using an MT system of interest T, and evaluates the quality of the generated hypothesis hyp_0 with reference to src_0 using a quality estimator Q. The pair of hyp_0 and its score initializes the best result (steps 1–3). The open list of errors to be edited is also initialized with translation errors in hyo_0 identified by an error detector E (steps 4–5).

Then, for pre-defined iterations N or until no errors remain (step 7), our method repeats the following steps: (a) identify the severest error and corresponding source text span (steps 8–9, §3.1), (b) edit the identified source span (step 10, §3.2), (c) translate the edited source text and evaluate the new hypothesis (steps 11–12, §3.3), and (d) search for the best translation (steps 8, 13–17, §3.4).

Algorithm 1: Proposed Error-Informed Source Text Editing Method.

Input : Original source text src_0, Translator T, Quality estimator Q, Error detector E, Maximum iteration N, Aligner A, Paraphraser P
Output : Best translation $best\_hyp$

1 $hyp_0 = T.translate(src_0)$
2 $best\_score = Q.evaluate(src_0, hyp_0)$
3 $best\_hyp = hyp_0$
4 $errs = E.detect\_errors(src_0, hyp_0)$
5 $cands = [\langle src_0, hyp_0, errs\rangle]$ `// open list`
6 $i = 1$
7 **while** $i < N \wedge cands \neq [\,]$ **do**
8 $\quad \langle src, hyp, err\rangle, cands = select\_one\_error(cands)$
9 $\quad src^{ann} = A.propagate\_error(src, hyp, err)$
10 $\quad src_i = P.paraphrase(src^{ann})$
11 $\quad hyp_i = T.translate(src_i)$
12 $\quad score_i = Q.evaluate(src_0, hyp_i)$
13 $\quad$ **if** $score_i > best\_score$ **then**
14 $\quad\quad best\_score = score_i$
15 $\quad\quad best\_hyp = hyp_i$
16 $\quad\quad errs = E.detect\_errors(src_i, hyp_i)$
17 $\quad\quad cands.append(\langle src_i, hyp_i, errs\rangle)$
18 $\quad i = i + 1$
19 **return** $best\_hyp$

3.1 Identification of the Source Text Span

In general, a translation hypothesis can contain multiple errors derived from dispersed source text spans. Following the incremental amelioration approach (Miyata and Fujita, 2017, 2021), we focus on the severest error and corresponding source text span (steps 8–9) in each iteration.

To this end, we rely on an error detector or a span-level quality estimator E, which identifies error spans with a severity score (steps 4 and 16). E may not jointly detect source text spans each corresponding to an error in the hypothesis. We thus identify such spans by aligning the source text tokens and those in the hypothesis using an aligner A (step 9). Then, we determine the source span that aligns with the severest error. Other tuples of an error span, its corresponding source text span, and its severity score are stored in the open list (steps 5 and 17) for future iterations (§3.4).

3.2 Targeted Text Editing Using an LLM

Given a source text annotated with a text span, our method then attempts to edit the span. Since the annotated span can be linguistically diverse, from a single symbol or word to the entire source text, we require a robust editor that can realize diverse types of text editing without introducing linguistic errors and semantic changes.

To perform such monolingual text editing, we use a decoder-only LLM, assuming that it has learned diverse linguistic phenomena from massive text data and is well-instructed for various text-editing tasks. In addition to the source text and the text span to be edited, it would be useful to instruct the LLM on the text editing task, such as its sub-steps and constraints, with some examples if possible. To better control its output, instructing the LLM with prompt formatting through few-shot examples is a promising approach (He et al., 2024). However, we need to prepare countermeasures against irregular outputs, such as control sequences and an off-target format.

3.3 Translation and Evaluation

The edited source text src_i is translated by the MT system T (step 11), and the quality of the generated hypothesis hyp_i is evaluated by the quality estimator Q (step 12). Text editing is not necessarily successful; it may fail to edit the annotated error source, thereby resulting in semantic drift in src_i and/or severer errors in hyp_i. Therefore, the quality is evaluated with respect to the original source text src_0 rather than the corresponding source text src_i as in Ki and Carpuat (2025).

3.4 Search for the Best Translation

We search for a better translation through repeating targeted source text editing, hypothesis generation with the MT system, and quality assessment. Given the high computational cost for LLM-based text editing, it is not feasible to traverse the entire search space. Therefore, we conduct depth-first search as in past manual investigations (Miyata and Fujita, 2017, 2021): our method performs source text editing greedily as long as quality improves. If an edit is confirmed to be detrimental by the quality estimator, it discards the edited text and selects the second severest error of the previous version of the source text. If no error remains, it backtracks to one more previous version of the source text. This is implemented in the "`select_one_error()`" function (step 8).

4 Preliminary Experiments

We determined the detailed settings using a small set of Japanese-to-English translation examples and one MT system. Henceforth, resources used in our experiments, including datasets, pre-trained model checkpoints, and tools, will be presented in this manner. See Appendix A for their details.

For this purpose, we used an NMT model pre-trained on JParaCrawl (Morishita et al., 2022) (the big model) and four sets of Japanese–English parallel data: (a) the development data of ASPEC (Nakazawa et al., 2016) (abstracts of scientific papers), (b) the test data of WMT22 (Kocmi et al., 2022) (mixture of several domains), (c) the test data of MTNT (Michel and Neubig, 2018) (users' posts on social media), and (d) the development data of the Kyoto Free Translation Task (benchmark splits of the Japanese–English Bilingual Corpus of Wikipedia's Kyoto Articles; henceforth, KFTT). First, we translated the Japanese side of these datasets into English using the MT model and Fairseq (Ott et al., 2019), and detected translation errors using a span-level quality estimator, XCOMET-XL (Guerreiro et al., 2024). We then randomly extracted 25 text pairs from each dataset that contained at least one "critical" or "major" error. Referring only to the sampled 100 ($= 25 \times 4$) text pairs, we determined the details of our method, including the combination of the span-level quality estimator and word aligner (Appendix C.1), the LLM for source text editing (Appendix C.2), and the prompt template (Appendix B.1).

To identify text spans in the source texts to be edited, our method propagates errors identified by XCOMET-XL, relying on optimal transport implemented by OTAlign (Arase et al., 2023). We determined the alignments between the tokens in the source text and hypothesis using contextual token embeddings obtained by InfoXLM-Base (Chi et al., 2021), uniform distribution as the mass for each token, cosine distance between token embeddings as the cost function, Sinkhorn algorithm, and 0.1 as the weight for the entropy-based regularizer.

For source text editing, we used Llama-3.1-Swallow-70B-Instruct-v0.1 (henceforth, Llama-Swallow) (Fujii et al., 2024), an LLM trained on massive text data of the source language, i.e., Japanese, and devised a prompt template for it, including the way of specifying the source text span to be edited and providing a one-shot paraphrase example. Such an example was randomly sampled from the 71 paraphrase examples in Japanese taken from a taxonomy of paraphrases.[1] The instruction was formatted by applying a chat template using the Language Model Evaluation Harness.

To select the best translation hypothesis among those derived from different versions of source texts, we also used XCOMET-XL, a reference-free quality estimation metric.

Through the preliminary experiments, we observed that the translation quality achieved by our method saturated up to five iterations (Appendix C). This is fewer than the 5.4–21.8 iterations required to obtain an acceptable translation in a manual investigation (Miyata and Fujita, 2021) because of the limited ability of our method compared with humans. The advancement of each component, as well as their tight integration, should improve the ability of our method.

5 Evaluation

To confirm the efficacy of our method on translation quality, we conducted experiments. Although most components of our method are multilingual, we had only Japanese and English speakers to write prompt templates for source text editing and translation with LLMs. Therefore, we evaluated the applicability of our method on Japanese-to-English (Ja→En), Japanese-to-Chinese (Ja→Zh), English-to-Japanese (En→Ja), and English-to-Chinese (En→Zh) translation directions.

5.1 Settings

5.1.1 Configuration of the Proposed Method

The configuration of our method mostly follows the details that we determined in our preliminary experiments (§4). For both translation error detection and translation quality estimation, we used XCOMET-XL, with one exception: we regarded text spans annotated as "critical," "major," or "minor" as errors, thereby extending the target. We used OTAlign and InfoXLM-Base for propagating erroneous spans to the source text.

Source text editing in our method is a monolingual task. Considering that an LLM trained specifically for the source language should perform better than other LLMs (Appendix C.2), we used Llama-Swallow for the Ja→En and Ja→Zh tasks, and Llama-3.1-70B-Instruct (henceforth, Llama) (Grattafiori et al., 2024) for the En→Ja and En→Zh tasks. We used 71 Japanese and 44 English examples available in the aforementioned paraphrase taxonomy[1] as the pool for the one-shot demonstration. The prompt templates are shown in Appendix B.1.

We set the number of maximum iterations, i.e., N in Algorithm 1, to 5, following our preliminary experiments (§4).

[1] https://paraphrasing.org/paraphrase.html

5.1.2 MT Systems

We applied our method to eight MT systems: four NMT and four LLM-based systems. Although we chose publicly available checkpoints for the sake of reproducibility, we regarded them as black boxes with the aim of simulating applications of our method to proprietary MT systems and services.

The NMT systems were NLLB-200-3B (henceforth, NLLB) (NLLB Team et al., 2022) and three sized-variants of the Ja→En and En→Ja specific models trained on JParaCrawl (Morishita et al., 2022), labeled as small, base, and big.

The four LLM-based MT systems were based on two LLMs, i.e., Llama and Llama-Swallow (see Appendix B.2 for their prompt templates), and two methods for selecting a translation example from a reference parallel corpus. One is BM25 (Robertson and Zaragoza, 2009), implemented in bm25s (Lù, 2024), which searches the parallel corpus for a text pair whose source side is most similar to the given source text. To this end, the source text to be translated and the corresponding side of the parallel corpora were tokenized with MeCab (Kudo et al., 2004) and Moses tokenizer (Koehn et al., 2007) for Japanese and English, respectively. The other method, called "vector," identifies such a text pair relying on sentence embeddings. We used LaBSE (Feng et al., 2022) as the sentence encoder and Faiss (Douze et al., 2024) for search, where we indexed the source side of the parallel corpora using product quantization with the number of subquantizers of 96 and the number of bits per subvector index of 8. As the reference parallel corpora, we used the official training data of WMT23 (Kocmi et al., 2023) consisting of 33.9M and 55.2M text pairs for Japanese–English and Chinese–English pairs, and Japanese–Chinese JParaCrawl (Nagata et al., 2024) consisting of 4.6M text pairs.

5.1.3 Test Datasets

For Ja→En, we used four datasets: [a] the test data of ASPEC (Nakazawa et al., 2016), [b] the test data of WMT23 (Kocmi et al., 2023) (mixture of several domains), [c] the test data of MTNT19 (Li et al., 2019) (users' posts on social media), and [d] the test data of KFTT. For Ja→Zh, we used [e] the test data of ASPEC. For En→Ja, we used four datasets: [f] the test data of the Asian Language Treebank (Riza et al., 2016) (news articles; henceforth, ALT), [g] the test data of WMT23, [h] the test data of MTNT19, and [i] the test data of IWSLT 2017 (Cettolo et al., 2017) (TED talks; henceforth, IWSLT). We also used [j] the test data of IWSLT 2017 for En→Zh.

When these datasets were translated by the target MT systems, 61.6%–96.2% of the resulting translations contained at least one "critical," "major," or "minor" error (see Appendix D for the details). Note that our method processes only these "erroneous test subsets."

5.1.4 Other Methods Compared

We regarded translation of the original source texts, i.e., hyp_0 in Algorithm 1, as the baseline. In addition, we evaluated the following "non-targeted" methods. Unlike ours, they are unaware of the target MT system and attempt to pre-edit source texts irrespective of potential translation errors.

Word-Sub: We replicated the word-substitution method proposed by Koretaka et al. (2023), which generates N-best paraphrases by substituting one word, relying on a masked language model and cosine similarity between word embeddings. We used language-specific BERT models trained through whole-word masking (Tohoku-NLP BERT base Japanese and Google BERT large for English) and FastText word embeddings.

Seq2seq-B: Although it is proven ineffective (Koretaka et al., 2023), we trained a monolingual sequence-to-sequence model for each source language, following Wieting et al. (2017) (see Appendix G for the training details), and obtained N paraphrased texts using the model via beam search with a beam size of 12.[2]

LLM-NT (non-targeted): We obtained N versions of source texts through iterative paraphrasing with an LLM.[3] The only difference from our method is that the source texts were not annotated on the basis of translation errors. We thus derived the prompt templates for Llama-Swallow and Llama from those for our method (Appendix B.1) by removing the step-wise instruction for error-informed text editing while retaining the criteria to meet.

Each of the N paraphrased source texts and the original source text was translated separately by the given MT system, and the best translation among

[2]We also examined nucleus sampling with $top\_p = 0.95$ but it underperformed beam search in most configurations.

[3]With $N = 1$, this is similar to one of the MT-Agnostic rewriting methods examined by Ki and Carpuat (2025).

MT System	Editing Method	Ja→En				Ja→Zh	En→Ja				En→Zh	#+	#-	#w
		ASPEC [a]	WMT23 [b]	MTNT19 [c]	KFTT [d]	ASPEC [e]	ALT [f]	WMT23 [g]	MTNT19 [h]	IWSLT [i]	IWSLT [j]			
NLLB [1]	Baseline	81.18	74.80	68.59	64.28	83.76	87.99	81.41	77.23	82.83	79.51	-	-	0
	Word-Sub	**81.47***	**76.40***	**70.50***	**66.91***	**84.05***	**88.70***	**83.35***	**79.43***	**84.44***	**79.95***	10	0	1
	Seq2seq-B	81.08	**75.78***	**70.09***	**66.51***	83.55*	**88.86***	**83.29***	**79.25***	**84.16***	**79.97***	8	1	1
	LLM-NT	**81.58***	**76.73***	**71.32***	**66.52***	**84.10***	**88.85***	**84.56***	**81.35***	**85.09***	**80.68***	10	0	7
	Ours	**81.67***	**76.24***	**70.36***	**65.90***	**83.90***	**88.52***	**82.71***	**78.50***	**83.29***	**79.83***	10	0	1
JParaCrawl (small) [2]	Baseline	81.75	76.92	72.77	73.46	-	85.74	80.26	74.69	79.73	-	-	-	0
	Word-Sub	**81.95***	**77.85***	**73.44***	**73.68**	-	**86.44***	**80.51**	74.04*	**80.63***	-	5	1	0
	Seq2seq-B	81.66	**77.29***	72.72	73.14	-	**86.60***	**80.84***	**74.73**	**80.42***	-	4	0	0
	LLM-NT	**82.12***	**78.25***	**74.17***	**74.11***	-	**87.15***	**83.38***	**79.36***	**82.74***	-	8	0	5
	Ours	**82.27***	**78.57***	**74.08***	**74.19***	-	**86.74***	**82.34***	**77.29***	**81.88***	-	8	0	3
JParaCrawl (base) [3]	Baseline	82.30	77.75	72.67	74.52	-	86.77	80.96	75.22	80.43	-	-	-	0
	Word-Sub	**82.45***	**78.40***	**73.76***	**74.63**	-	**87.29***	**81.35***	73.77*	**81.13***	-	6	1	0
	Seq2seq-B	82.09*	**78.00***	**72.85**	74.32	-	**87.19***	**81.64***	**75.26**	**81.00***	-	4	1	0
	LLM-NT	**82.57***	**78.98***	**74.54***	**75.09***	-	**87.93***	**83.84***	**79.66***	**83.15***	-	8	0	4
	Ours	**82.71***	**79.15***	**74.49***	**75.30***	-	**87.94***	**83.14***	**77.97***	**82.58***	-	8	0	4
JParaCrawl (big) [4]	Baseline	82.87	79.26	74.96	76.25	-	88.04	82.31	76.36	81.63	-	-	-	0
	Word-Sub	**82.96**	**79.61***	74.23*	**76.50**	-	**88.30***	**82.78***	75.70*	**82.21***	-	4	2	0
	Seq2seq-B	82.63*	79.26	74.35*	75.27*	-	**88.45***	**82.75***	**76.55**	**82.22***	-	3	3	0
	LLM-NT	**83.09***	**79.96***	**75.68***	**76.76***	-	**88.94***	**84.75***	**80.59***	**83.96***	-	8	0	6
	Ours	**83.06***	**80.01***	**75.72***	**76.50**	-	**88.91***	**84.12***	**79.32***	**83.19***	-	7	0	2
Llama (BM25) [5]	Baseline	81.61	80.45	75.26	76.96	86.57	89.62	85.51	82.32	81.64	81.86	-	-	0
	Word-Sub	**82.92***	**81.20***	**76.17***	**77.98***	**86.72**	**89.96***	**86.25***	**82.88***	**83.13***	**83.00***	9	0	6
	Seq2sec-B	**82.63***	**80.71**	**75.84***	**77.46**	86.34*	**89.96***	**86.23***	**83.04***	**82.91***	**83.07***	7	1	1
	LLM-NT	**82.56***	**81.13***	**76.32***	**77.71***	**86.69***	**89.63**	**86.12***	**83.31***	**83.89***	**83.17***	9	0	4
	Ours	**81.76***	**80.98***	**75.93***	**77.41***	86.57	**89.70**	**86.06***	**83.25***	**83.33***	**82.79***	8	0	0
Llama (vector) [6]	Baseline	81.98	80.67	74.75	76.82	86.37	89.86	85.92	82.44	81.85	82.21	-	-	0
	Word-Sub	**82.90***	**81.47***	**76.07***	**77.96***	**86.81***	**89.92**	**86.54***	**83.49***	**83.57***	**83.05***	9	0	6
	Seq2sec-B	**82.57***	**80.82**	**75.55***	**77.36***	86.31	**90.00**	**86.56***	**82.76**	**82.91***	**82.74***	6	0	2
	LLM-NT	**82.57***	**81.23***	**75.80***	**77.44***	**86.65***	89.70	**86.21**	**83.74***	**84.16***	**82.94***	8	0	2
	Ours	**82.16***	**81.08***	**75.54***	**77.00**	**86.38**	**90.00**	**86.38***	**83.10***	**83.10***	**82.78***	7	0	1
Llama-Swallow (BM25) [7]	Baseline	80.84	80.52	75.14	77.15	86.54	90.68	86.40	83.38	83.83	81.83	-	-	1
	Word-Sub	**82.06***	**81.60***	**76.32***	**78.45***	86.37*	**90.96***	**87.12***	**84.08***	**85.34***	**82.20***	9	1	2
	Seq2sec-B	**82.41***	**81.25***	**76.40***	**77.97***	86.02*	**90.92***	**87.01***	**83.93***	**85.04***	**82.47***	9	1	1
	LLM-NT	**82.08***	**81.66***	**76.44***	**78.09***	86.51	**90.80**	**87.16***	**84.38***	**85.69***	**82.52***	8	0	6
	Ours	80.76	**80.85***	**75.27**	**77.19**	86.20*	**90.92***	**87.01***	**84.05***	**85.14***	**82.21***	6	1	0
Llama-Swallow (vector) [8]	Baseline	81.40	80.93	74.62	77.69	86.48	90.85	86.75	83.46	84.34	81.76	-	-	1
	Word-Sub	**82.64***	**81.94***	**76.38***	**78.38***	86.36	**91.16***	**87.47***	**84.06***	**85.67***	**82.48***	9	0	7
	Seq2sec-B	**82.72***	**81.21**	**75.87***	**78.00**	86.05*	**91.05***	**87.33***	**83.96***	**85.05***	**82.30***	7	1	1
	LLM-NT	**82.36***	**81.65***	**76.27***	**78.25***	86.37	**90.96**	**87.22***	**84.29***	**85.62***	**82.46***	8	0	0
	Ours	81.38	**81.33***	**75.32***	77.58	86.07*	**90.96**	**87.16***	**84.30***	**85.43***	**82.01**	5	1	1

Table 1: COMET scores for the entire test sets. **Bold** indicates the improvement over the Baseline, underlining so does the best score among all the methods, the "#+" and "#-" columns show the number of test datasets for which the method achieved a significantly better or worse score (marked with "*") than the Baseline, respectively, and the "#w" column presents the number of datasets for which the method achieved the best score for each MT system.

$(N+1)$ hypotheses was selected by XCOMET-XL similarly to our method.[4]

5.1.5 Evaluation Metric

To evaluate the translation quality of MT outputs, we used the COMET score (Rei et al., 2020), specifically with the wmt22-comet-da checkpoint (Rei et al., 2022). We performed paired bootstrap resampling (Koehn, 2004) to test the statistical significance of the score difference from the baseline.

[4]XCOMET-XL achieved consistently better results than mBART used in Koretaka et al. (2023) in our preliminary experiments, although it was substantially slow.

5.2 Results

The COMET scores of all the methods in 74 test configurations are presented in Table 1. For the four NMT systems (the upper half), the LLM-based text editing methods, i.e., LLM-NT and ours, significantly improved translation quality, with only one exception. The word-substitution method also led to a significant gain in roughly 70% of configurations, outperforming the sequence-to-sequence method. For the LLM-based MT systems (the bottom half), the LLM-based text editing methods and the word-substitution method also achieved significant improvements. The word-substitution method

MT System	Editing Method	Ja→En				Ja→Zh	En→Ja				En→Zh	#+	#-	#w
		ASPEC [a]	WMT23 [b]	MTNT19 [c]	KFTT [d]	ASPEC [e]	ALT [f]	WMT23 [g]	MTNT19 [h]	IWSLT [i]	IWSLT [j]			
NLLB [1]	Baseline	79.71	72.94	67.92	66.49	84.01	87.48	82.25	78.79	83.72	79.70	-	-	0
	Word-Sub	**80.07***	**74.28***	**69.34***	**68.20***	**84.16***	**87.95***	**83.22***	**79.64***	**84.12***	79.69	9	0	0
	Seq2seq-B	79.48*	**73.62***	**68.70***	**67.54***	83.64*	**88.11***	**83.19***	**79.66***	**84.13***	**79.89**	7	2	0
	LLM-NT	**80.21***	**74.96***	**70.26***	**68.01***	**84.25***	**88.13***	**84.22***	**81.20***	**84.55***	**80.26***	10	0	5
	Ours	**80.43***	**75.11***	**70.48***	**68.44***	**84.16***	**88.15***	**83.90***	**80.44***	**84.33***	**80.09***	10	0	5
JParaCrawl (small) [2]	Baseline	80.38	74.70	71.54	73.45	-	84.98	79.25	73.80	78.18	-	-	-	0
	Word-Sub	**80.66***	**75.78***	**72.13***	**73.61**	-	**85.76***	**79.53**	73.05*	**79.18***	-	5	1	0
	Seq2seq-B	80.25	**75.10***	71.30	73.10*	-	**85.89***	**79.87***	73.77	**78.92***	-	4	1	0
	LLM-NT	**80.86***	**76.43***	**72.80***	**74.06***	-	**86.45***	**82.65***	**78.71***	**81.66***	-	8	0	4
	Ours	**81.10***	**76.82***	**73.16***	**74.24***	-	**86.11***	**81.62***	**76.74***	**80.76***	-	8	0	4
JParaCrawl (base) [3]	Baseline	81.04	75.38	70.84	74.48	-	85.91	79.87	74.51	78.93	-	-	-	0
	Word-Sub	**81.24***	**76.18***	**72.08***	**74.56**	-	**86.46***	**80.23***	72.98*	**79.72***	-	6	1	0
	Seq2seq-B	80.76*	**75.63**	**70.88**	74.25	-	**86.39***	**80.56***	74.51	**79.61***	-	3	1	0
	LLM-NT	**81.40***	**76.96***	**72.92***	**75.09***	-	**87.22***	**83.04***	**79.19***	**82.11***	-	8	0	3
	Ours	**81.60***	**77.21***	**73.12***	**75.32***	-	**87.27***	**82.37***	**77.66***	**81.51***	-	8	0	5
JParaCrawl (big) [4]	Baseline	81.61	77.10	73.20	76.34	-	87.08	81.00	75.62	79.82	-	-	-	0
	Word-Sub	**81.73**	**77.53***	72.17*	**76.58**	-	**87.38***	**81.50***	74.86*	**80.58***	-	4	2	0
	Seq2seq-B	81.27*	77.00	72.45*	75.41*	-	**87.56***	**81.49***	**75.76**	**80.56***	-	3	3	0
	LLM-NT	**81.91***	**78.02***	**73.90***	**76.82***	-	**88.14***	**83.76***	**80.11***	**82.76***	-	8	0	6
	Ours	**81.90***	**78.13***	**74.18***	**76.61**	-	**88.12***	**83.14***	**78.98***	**81.78***	-	7	0	2
Llama (BM25) [5]	Baseline	82.05	79.24	74.23	78.47	86.58	89.02	84.76	81.39	81.02	81.40	-	-	0
	Word-Sub	**82.41***	**79.69***	**74.51**	**78.85***	**86.62**	**89.25**	**85.32***	**81.81**	**82.34***	**82.29***	6	0	1
	Seq2seq-B	81.86*	78.80*	73.77	78.05*	86.21*	**89.35***	**85.37***	**81.93***	**82.11***	**82.38***	5	4	1
	LLM-NT	**82.25***	**79.70***	**74.84***	**78.58**	**86.65**	89.01	**85.21***	**82.29***	**83.18***	**82.44***	7	0	1
	Ours	**82.27***	**80.04***	**75.17***	**78.96***	86.57	**89.12**	**85.46***	**82.55***	**83.22***	**82.48***	8	0	7
Llama (vector) [6]	Baseline	82.10	79.48	73.75	78.25	86.61	89.23	84.89	81.52	81.22	81.66	-	-	0
	Word-Sub	**82.33***	**79.93***	**74.44***	**78.54**	**86.69**	**89.24**	**85.49***	**82.36***	**82.64***	**82.39***	7	0	2
	Seq2seq-B	81.77*	79.02*	**73.79**	77.75*	86.18*	**89.25**	**85.50***	**81.66**	**82.10***	**82.07***	3	4	1
	LLM-NT	**82.28***	**79.79***	**74.40***	**78.57**	**86.68**	88.91	**85.07**	**82.59***	**83.36***	**82.25***	4	1	3
	Ours	**82.36***	**80.10***	**74.87***	**78.45**	**86.62**	**89.43**	**85.48***	**82.36***	**82.87***	**82.32***	7	0	4
Llama-Swallow (BM25) [7]	Baseline	82.33	79.79	75.29	78.76	86.62	89.82	85.31	82.27	82.61	81.23	-	-	1
	Word-Sub	82.12*	**79.90**	74.90	**78.86**	86.29*	**90.20***	**86.07***	**82.75***	**84.30***	**81.50**	4	2	1
	Seq2seq-B	81.81*	79.39*	74.59*	78.30*	85.88*	**90.10***	**85.95***	**82.59**	**83.77***	**81.85***	4	5	1
	LLM-NT	**82.41**	**79.92**	75.08	**78.98**	86.49*	**89.94**	**86.14***	**83.10***	**84.63***	**81.78***	4	1	4
	Ours	82.20	**80.32***	**75.49**	**78.79**	86.25*	**90.15***	**86.13***	**83.14***	**84.40***	**81.66***	6	1	3
Llama-Swallow (vector) [8]	Baseline	82.29	80.03	74.50	78.85	86.59	90.00	85.75	82.13	83.18	81.20	-	-	1
	Word-Sub	82.24	**80.22**	**74.99**	78.78	86.29*	**90.32***	**86.46***	**82.73***	**84.64***	**81.81***	5	1	3
	Seq2seq-B	81.84*	79.40*	74.32	78.35*	85.93*	**90.24***	**86.29***	**82.59***	**83.88***	**81.66***	5	4	0
	LLM-NT	**82.37**	**80.28**	**75.23***	**78.97**	86.37*	**90.08**	**86.12***	**83.10***	**84.49***	**81.74***	5	1	2
	Ours	82.25	**80.67***	**75.55***	78.73	86.14*	**90.15**	**86.31***	**83.22***	**84.69***	**81.47**	5	1	4

Table 2: COMET scores for the erroneous test subsets (§5.1.3). See Table 1 for text decoration and symbols.

achieved the highest COMET score in more than 50% of configurations, whereas our method lagged behind it and LLM-NT.

Unlike existing non-targeted methods, our method edits only the erroneous test subsets (§5.1.3). Table 2 compares the COMET scores for these subsets, revealing the advantage of our method and diminished impact of the non-targeted methods. Our method achieved the highest COMET score in 34 out of 74 configurations, followed by LLM-NT which won in 28.

From these results, we conclude that the LLMs performed source text editing for MT more robustly and accurately than the existing methods. However, our targeted method is not yet incontestably superior over its non-targeted counterpart, i.e., LLM-NT. For instance, the prompt template developed with Ja→En examples had a minimal or negative impact on the Ja→Zh task, in particular with Llama-Swallow. In contrast, the equivalent prompt template manually translated into English worked fairly well, encouraging future applications of our method to other source languages.

6 Analyses

To better understand the characteristics of our method, we conducted several analyses, focusing on the erroneous test subsets (§5.1.3).

6.1 System-wise and Dataset-wise Impact

We hypothesize that the worse the quality of an MT output for the original source text is, the more

MT System	Ja→En				Ja→Zh	En→Ja				En→Zh
	ASPEC [a]	WMT23 [b]	MTNT19 [c]	KFTT [d]	ASPEC [e]	ALT [f]	WMT23 [g]	MTNT19 [h]	IWSLT [i]	IWSLT [j]
NLLB [1]	-0.381	-0.401	-0.416	-0.388	-0.324	-0.453	-0.540	-0.444	-0.340	-0.359
JParaCrawl (small) [2]	-0.368	-0.437	-0.339	-0.292	-	-0.375	-0.434	-0.340	-0.452	-
JParaCrawl (base) [3]	-0.392	-0.389	-0.385	-0.324	-	-0.493	-0.454	-0.373	-0.489	-
JParaCrawl (big) [4]	-0.393	-0.391	-0.382	-0.275	-	-0.479	-0.481	-0.437	-0.461	-
Llama (BM25) [5]	-0.353	-0.381	-0.322	-0.342	-0.269	-0.237	-0.400	-0.385	-0.633	-0.565
Llama (vector) [6]	-0.340	-0.380	-0.342	-0.231	-0.297	-0.264	-0.369	-0.347	-0.573	-0.486
Llama-Swallow (BM25) [7]	-0.228	-0.338	-0.335	-0.262	-0.215	-0.480	-0.466	-0.391	-0.683	-0.457
Llama-Swallow (vector) [8]	-0.281	-0.345	-0.347	-0.163	-0.157	-0.470	-0.394	-0.395	-0.619	-0.460

Table 3: Pearson product-moment correlation coefficients r between the baseline COMET score and its gain achieved by our method. See Appendix D for the number of segments in each configuration.

it should benefit from avoiding underlying translation errors by source text editing. To examine this, we calculated the correlation between the baseline COMET score and its gain. Table 3 summarizes segment-level correlation coefficients. Although there was moderate negative correlation for most configurations, we observed that others, such as the Ja→Zh ASPEC dataset translated by the LLM-based MT systems, did not follow the rule. In general, correlation for the LLM-based MT systems was weaker than those for the NMT systems, except for the two IWSLT tasks, and more diverse over the datasets. This implies that these LLMs had peculiar characteristics. For instance, some of the test datasets might have already been learned by them, unlike the NMT systems.

Figure 1 visualizes the correspondences between the baseline COMET score and its gain for each erroneous test subset. Our method had a stronger correlation than the other methods. One may consider that our method could be less impactful for very accurate MT systems, such as "black-box" proprietary systems. Despite this, we consider that our approach still has potential, because its advancement is orthogonal to the enhancements of those MT systems. For instance, we developed our method using only one NMT system and a small sample of Ja→En sentence pairs (§4), but it worked well for other translation directions (4f–4i) and stronger LLM-based MT systems (e.g., 5g–8i). Through our extensive experiments, we identified configurations where the current form of our method did not work well, such as the two ASPEC and KFTT tasks. We will conduct in-depth analyses to explore the reasons and address them in our future work.

6.2 Quality Estimator

We investigated whether the segment-level quality estimator, i.e., XCOMET-XL, was useful for search-

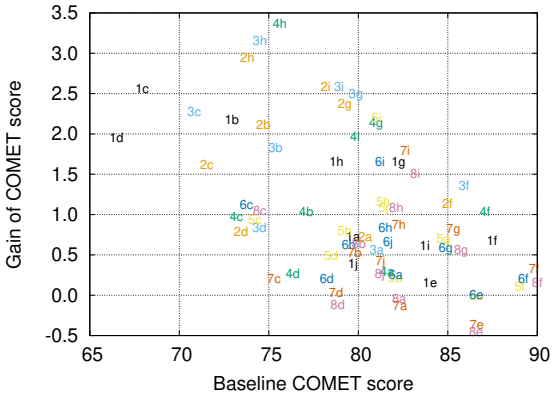

Figure 1: Baseline COMET score for the erroneous test subsets and its gain achieved by our method ($r = -0.522$). "1" to "8" and "a" to "j" are the indices of the MT systems and test datasets, respectively.

ing for the best translation. Figure 2 shows that the estimated quality monotonically improved during the search as intended; 1.91–8.48 and 2.46–11.05 points with $N = 1$ and $N = 5$, respectively. However, the final COMET score shown in Figure 3 did not follow the same trend, even though these two measures correlated well at the segment level in our experiments (0.292–0.762, Appendix E). For instance, in the two configurations where our method significantly deteriorated the COMET score, i.e., the two variants based on Llama-Swallow for the Ja→Zh ASPEC task (7e and 8e shown at top right of Figure 3), the correlation coefficient between the two measures was moderate (0.534 and 0.519). In contrast, the configurations with a weakest correlation, i.e., the JParaCrawl variants applied to the En→Ja IWSLT task (2e–4e, $r = 0.292$–0.296), achieved a 2.0–2.5 COMET point gain.

This discrepancy suggests that the segment-level quality estimator was a vital component, and thus requires further improvements to capture subtle differences between accurate translations.

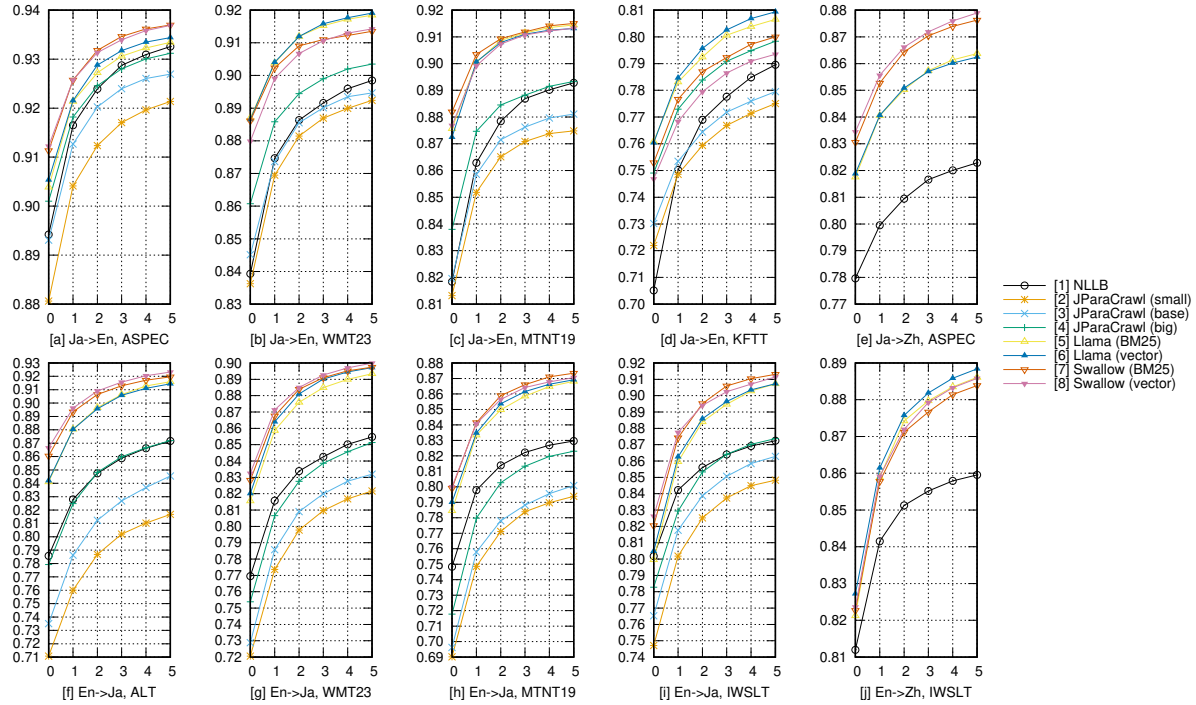

Figure 2: Translation quality estimated by XCOMET-XL without reference at each iteration of our method.

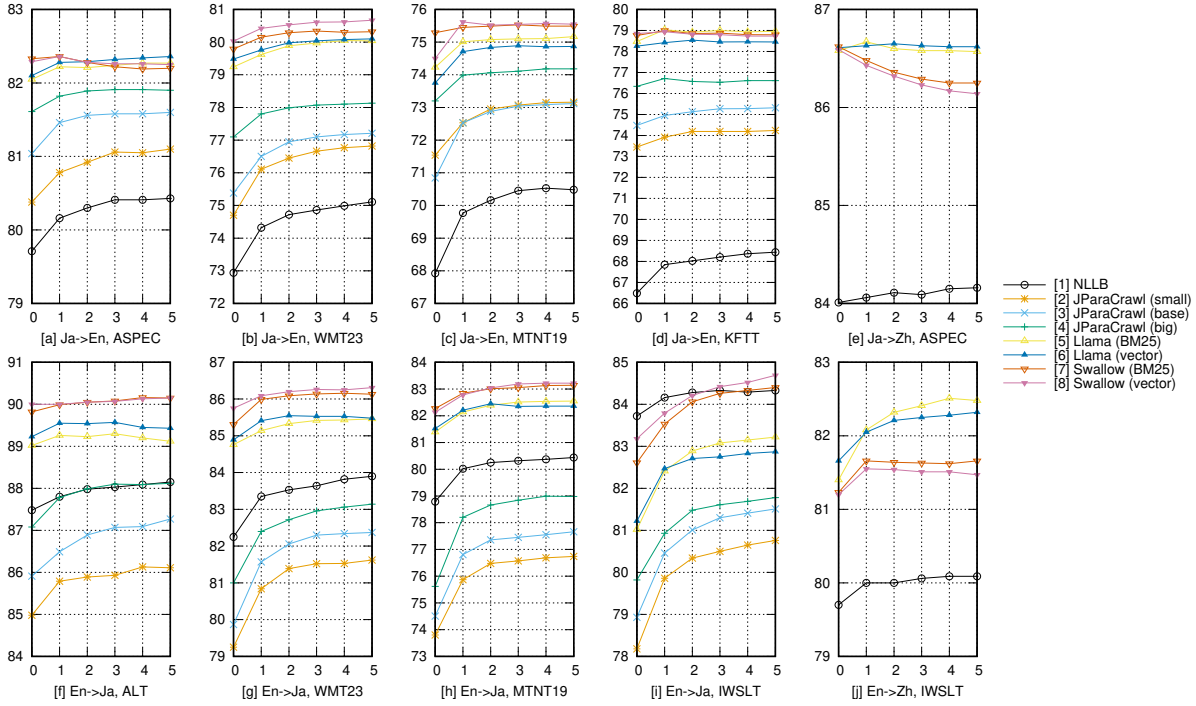

Figure 3: COMET score computed by wmt22-comet-da with a reference at each iteration of our method.

6.3 Source Text Editor

We quantified the degree of text editing performed by each method, with the translation error rate (TER) (Snover et al., 2006) at the dataset level. More specifically, we tokenized Japanese and English texts using MeCab (Kudo et al., 2004) and Moses tokenizer (Koehn et al., 2007), respectively, and computed TER using SacreBLEU (Post, 2018),[5] regarding the original and edited source texts as the reference and hypothesis, respectively.

Figure 4 shows that our method altered 9%–37% of linguistic tokens. The ratio was higher

[5]Signature: nrefs:1|case:lc|tok:tercom|norm:no|punct:yes|asian:no|version:2.5.1

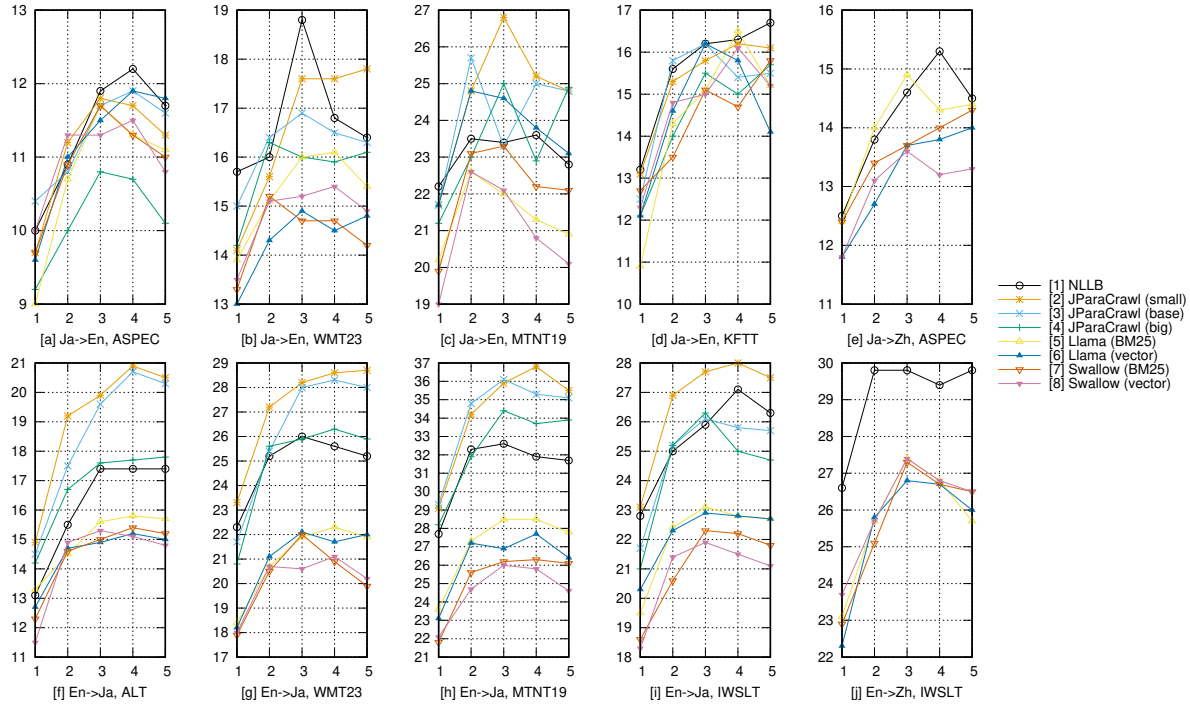

Figure 4: Translation edit rate (TER) between the original source text src_0 and each of its edited versions src_i generated by our method. Because of the search function, src_{i+1} was not necessarily obtained from src_i directly.

than those exhibited by the word-substitution method (9%–19%; Appendix F), except for the two ASPEC tasks (9%–16% vs. 17%–19%), and lower than those performed by the other two non-targeted methods (12%–76% and 19%–68% by the sequence-to-sequence and LLM-NT methods, respectively; Appendix F). We also observed that the ratio varied across the targeted MT systems.

The modest ratio of our method reflects the length distribution of the translation error spans and corresponding source text spans identified by XCOMET-XL and OTAlign. We thus consider that the LLMs properly conformed to the instruction for targeted text editing.

Note that we do not consider the ratio to be a good indicator of better translations.

7 Conclusion

As an approach to exploiting black-box MT systems, we focused on automatic and targeted source text editing. To overcome the two issues that remain in the literature, i.e., the limited ability of editing and unawareness of actual translation errors, we used LLMs, expecting that they would have sufficiently high competence to realize diverse types of edits that can improve translation quality (Miyata and Fujita, 2017, 2021) and a segment-level quality estimator as in a concurrent work (Ki and Carpuat, 2025), and implemented targeted paraphrasing (Uchimoto et al., 2006; Resnik et al., 2010) by harnessing a span-level quality estimator (error detector) and a word aligner.

Our experiments with eight MT systems and ten test datasets for four translation directions confirmed the efficacy of our method in improving translation quality, while the non-targeted counterpart (LLM-NT) also achieved a rivaling performance. Through the analyses, we identified that the impact of our method varied depending on the MT system and dataset, and that the segment-level quality estimator is the vital component that requires further improvements.

Future work includes improving each component of our method, while simplifying and speeding up the whole pipeline. Since our method focuses on source texts that lead to translation errors according to an error detector, applying LLM-NT to other error-free segments will be a straightforward way for improving translation for entire datasets. Only prompt templates are language dependent; hence, we plan to evaluate the applicability of our method to other translation tasks as well as other MT systems, including proprietary ones. We are also interested in assessing the applicability of the proposed method to other text-to-text tasks, including text summarization and text simplification.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. A part of these research results was obtained from the commissioned research (No. 22501) by NICT.

Limitations

As observed in our experiments, the gain of the COMET score achieved by our method and competing methods depended on the target MT systems and test datasets. In addition, the COMET score may evaluate the translation quality only from limited perspectives, heavily relying on a single reference translation. Thus, our results do not guarantee the same conclusions for other MT systems, datasets, and translation directions. For instance, our method may not work well for translating from/into low-resource languages, provided that the component models of our method, i.e., quality estimator, error detector, aligner, and paraphraser, have not been trained for those languages and thereby perform less accurately.

We used COMET (wmt22-comet-da) for evaluating the translation quality, following Freitag et al. (2022); they reported that it achieved the highest correlation with human rating among reproducible automatic metrics. However, recent work, such as Agrawal et al. (2024), demonstrated that XCOMET-XL surpassed COMET. Evaluating with XCOMET-XL may lead to different conclusions. The discrepancy between reference-free and reference-based metrics observed in Figures 2 and 3 could be resolved. On the other hand, the use of the same model for search and evaluation may lead to a bias. To confirm the gain in translation quality, human evaluation is indispensable.

We made large efforts to refine the prompt templates for text editing and translation with LLMs. However, there is still room for improvement. While our prompt templates (Appendix B.1) follow the spirit of "targeted paraphrasing" (Uchimoto et al., 2006; Resnik et al., 2010), a concurrent work (Ki and Carpuat, 2025) has demonstrated that the instruction for text simplification results in better translations than instructing on the paraphrasing task. The optimized templates for an LLM may not work well for other LLMs.

LLM-based source text editing requires substantially larger computes than existing methods that do not rely on LLMs. However, we assume that the latency with $N = 5$ and eight NVIDIA Tesla V100 GPUs is acceptable for the existing translation production process (ISO/TC37, 2017), where the manual post-editing step governs latency. We expect that recent advances in smaller language models and model compression will make our approach faster and consequently more feasible.

Ethics Statements

The resulted translations had, on average, a higher quality according to the COMET score. However, translation errors should remain. Therefore, the direct use of MT outputs could mislead potential users.

References

Sweta Agrawal, António Farinhas, Ricardo Rei, and Andre Martins. 2024. Can automatic metrics assess high-quality translations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14491–14502.

Yuki Arase, Han Bao, and Sho Yokoi. 2023. Unbalanced Optimal Transport for Unbalanced Word Alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3966–3986.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. *Preprint*, arXiv:2401.08281.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-Centric Multilingual Machine Translation. *The Journal of Machine Learning Research*, 22(1):4839–4886.

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*, pages 878–891.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. 2021. The Eval4NLP Shared Task on Explainable Quality Estimation: Overview and Results. In *Proceedings of the Second Workshop on Evaluation and Comparison of NLP Systems*, pages 165–178.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust. In *Proceedings of the Seventh Conference on Machine Translation*, pages 46–68.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities. In *Proceedings of the First Conference on Language Modeling*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent Machine Translation Evaluation through Fine-grained Error Detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does Prompt Formatting Have Any Impact on LLM Performance? *Preprint*, arXiv:2411.10541.

ISO/TC37. 2017. ISO 18587:2017 Translation Services – Post-editing of Machine Translation Output – Requirements.

Dayeon Ki and Marine Carpuat. 2025. Automatic Input Rewriting Improves Translation with Large Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10829–10856.

Yeun-Bae Kim and Terumasa Ehara. 1994. An Automatic Sentence Breaking and Subject Supplement Method for J/E Machine Translation. *IPSJ Journal*, 35(6):1018–1028. (in Japanese).

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, and 2 others. 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.

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*, pages 1–45.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Hyuga Koretaka, Tomoyuki Kajiwara, Atsushi Fujita, and Takashi Ninomiya. 2023. Mitigating Domain Mismatch in Machine Translation via Paraphrasing. In *Proceedings of the Tenth Workshop on Asian Translation*, pages 29–40.

Taku Kudo and John Richardson. 2018. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the First Shared Task on Machine Translation Robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102.

Xing Han Lù. 2024. BM25S: Orders of magnitude faster lexical search via eager sparse scoring. *Preprint*, arXiv:2407.03618.

Sneha Mehta, Bahareh Azarnoush, Boris Chen, Avneesh Saluja, Vinith Misra, Ballav Bihani, and Ritwik Kumar. 2020. Simplify-Then-Translate: Automatic Preprocessing for Black-Box Translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8488–8495.

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.

Shachar Mirkin, Sriram Venkatapathy, Marc Dymetman, and Ioan Calapodescu. 2013. SORT: An Interactive Source-Rewriting Tool for Improved Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 85–90.

Rei Miyata and Atsushi Fujita. 2017. Dissecting Human Pre-Editing toward Better Use of Off-the-Shelf Machine Translation Systems. In *Proceedings of the 20th Annual Conference of the European Association for Machine Translation*, pages 54–59.

Rei Miyata and Atsushi Fujita. 2021. Understanding Pre-Editing for Black-Box Neural Machine Translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1539–1550.

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.

Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. 2024. A Japanese-Chinese Parallel Corpus Using Crowdsourcing for Web Mining. *Preprint*, 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*, pages 2204–2208.

Hiroaki Nanjo, Yuji Yamamoto, and Takehiko Yoshimi. 2012. Automatic Construction of Statistical Preediting System from Parallel Corpus for Improvement of Machine Translation Quality. *IPSJ Journal*, 64(6):1644–1653. (in Japanese).

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, and 20 others. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. *Preprint*, arXiv:2207.04672.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

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

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2685–2702.

Philip Resnik, Olivia Buzek, Chang Hu, Yakov Kronrod, Alex Quinn, and Benjamin B. Bederson. 2010. Improving Translation via Targeted Paraphrasing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 127–137.

Hammam Riza, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng, Khin Mar Soe, Khin Thandar Nwet, Masao Utiyama, and Chenchen Ding. 2016. Introduction of the Asian Language Treebank. In *Proceedings of the 2016 Conference of the Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique (O-COCOSDA)*, pages 1–6.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Satoshi Shirai, Satoru Ikehara, and Tsukasa Kawaoka. 1993. Effects of Automatic Rewriting of Source Language within a Japanese to English MT System. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 226–239.

Satoshi Shirai, Satoru Ikehara, Akio Yokoo, and Yoshifumi Ooyama. 1998. Automatic Rewriting Method for Internal Expressions in Japanese to English MT

and Its Effects. In *Proceedings of the Second International Workshop on Controlled Language Applications*, pages 62–75.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the Seventh Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 Shared Task on Quality Estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 Shared Task on Quality Estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725.

Sanja Štajner and Maja Popovic. 2016. Can Text Simplification Help Machine Translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.

Sanja Štajner and Maja Popović. 2018. Improving Machine Translation of English Relative Clauses with Automatic Text Simplification. In *Proceedings of the First Workshop on Automatic Text Adaptation*, pages 39–48.

Yanli Sun, Sharon O'Brien, Minako O'Hagan, and Fred Hollowood. 2010. A Novel Statistical Pre-Processing Model for Rule-Based Machine Translation System. In *Proceedings of the 14th Annual Conference of the European Association for Machine Translation*.

Kiyotaka Uchimoto, Naoko Hayashida, Toru Ishida, and Hitoshi Isahara. 2006. Automatic Detection and Semi-Automatic Revision of Non-Machine-Translatable Parts of a Sentence. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 703–708.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 5998–6008.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285.

Masaya Yamaguchi, Nobuo Inui, Yoshiyuki Kotani, and Hirohiko Nishimura. 1998. Acquisition of Automatic Pre-edition Rules from Results of Pre-edition. *IPSJ Journal*, 39(1):17–28. (in Japanese).

Takehiko Yoshimi. 2001. Improvement of Translation Quality of English Newspaper Headlines by Automatic Pre-editing. *Machine Translation*, 16:233–250.

A Public Resources Used

Table 4 lists the links to the resources, including the datasets, pre-trained model checkpoints, and tools, used in our experiments (Sections 4–6).

B Prompt Templates

B.1 For Targeted Source Text Editing

Figure 5 presents the prompt templates used for our targeted source text editing. To perform this task, we used an LLM that is mainly trained on the language of interest, i.e., Llama-Swallow for Japanese and Llama for English, and provided prompts in the same language. Pairs of double brackets ("{{" and "}}") indicate placeholders. Given a source text to be edited, a tailored prompt is automatically instantiated by filling these placeholders.

B.2 For Translation

Figure 6 presents the prompt templates used for MT. In the same manner as for source text editing, we provided prompts in the language for which the LLM was recently and mainly trained on, i.e., Japanese for Llama-Swallow and English for Llama. The role of double brackets is the same as for source text editing.

C Preliminary Experiments

As described in §4, we explored a better way of determining the source text span to be edited and the LLM used for source text editing, using the sampled set of 100 parallel sentences and the JParaCrawl Japanese-to-English NMT model (big).

C.1 Source Text Span Detection Methods

To confirm the feasibility and impact of determining source text spans to be edited, we compared the following three methods.

Random: This method first specifies the beginning of the source text span randomly, and then the end from the remainder of the source text.

Datasets
Asian Language Treebank (ALT), https://huggingface.co/datasets/mutiyama/alt
ASPEC, https://jipsti.jst.go.jp/aspec/
IWSLT 2017, https://huggingface.co/datasets/IWSLT/iwslt2017
Japanese-English Bilingual Corpus of Wikipedia's Kyoto Articles, https://alaginrc.nict.go.jp/WikiCorpus/, 2.01
JParaCrawl, https://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/, v3.0
Kyoto Free Translation Task (KFTT), https://www.phontron.com/kftt/, 1.4
MTNT, https://github.com/pmichel31415/mtnt/, v1.1
MTNT19, https://pmichel31415.github.io/mtnt/, MTNT2019.tar.gz
WMT22 Test sets, https://github.com/wmt-conference/wmt22-news-systems, v1.1
WMT23 Test sets, https://github.com/wmt-conference/wmt23-news-systems, v0.1
Pre-trained model checkpoints
FastText word embeddings, https://fasttext.cc/docs/en/crawl-vectors.html
Google BERT large, https://huggingface.co/google-bert/bert-large-cased-whole-word-masking
InfoXLM-Base, https://huggingface.co/microsoft/infoxlm-base
JParaCrawl NMT Models, https://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/, based on v3.0
LaBSE, https://huggingface.co/sentence-transformers/LaBSE
Llama-3.1-70B-Instruct, https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
Llama-3.1-8B-Instruct, https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
Llama-3.1-Swallow-70B-Instruct-v0.1, https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-70B-Instruct-v0.1
Llama-3.1-Swallow-8B-Instruct-v0.1, https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.1
M2M100-418M, https://huggingface.co/facebook/m2m100_418M
NLLB-200-3.3B, https://huggingface.co/facebook/nllb-200-3.3B
Tohoku-NLP BERT base Japanese, https://huggingface.co/tohoku-nlp/bert-base-japanese-whole-word-masking
wmt22-comet-da, https://huggingface.co/Unbabel/wmt22-comet-da
XCOMET-XL, https://huggingface.co/Unbabel/XCOMET-XL
Tools
bm25s, https://github.com/xhluca/bm25s, 0.2.6
COMET, https://github.com/Unbabel/COMET, 2.2.5
Fairseq, https://github.com/facebookresearch/fairseq, v0.12.2
Faiss, https://github.com/facebookresearch/faiss, v1.7.2
Language Model Evaluation Harness, https://github.com/EleutherAI/lm-evaluation-harness, v0.4.3
MeCab, https://taku910.github.io/mecab/, 0.996
Moses tokenizer, https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl, RELEASE-4.0
OTAlign, https://github.com/yukiar/OTAlign, 79fefaa
SacreBLEU, https://github.com/mjpost/sacrebleu, v2.5.1
SentencePiece, https://github.com/google/sentencepiece, v0.2.0

Table 4: Public resources used in our experiments.

Direct: The error detector, XCOMET-XL, annotates only erroneous text spans in the hypothesis. This method examines whether it can directly annotate the source text spans by swapping the source text and hypothesis.

Propagation: A combination of XCOMET-XL for annotating erroneous text spans in the hypothesis and a word aligner, OTAlign, to propagate those spans to the source text.

Figure 7 shows the results with Llama-Swallow for Japanese source text editing (Appendix C.2). The "Propagation" method achieved the highest COMET score with any value of N up to 5, even though it should have involved prediction errors of both the error detector and word aligner. Interestingly, with the "Random" text spans, our method improved the COMET score up to 1.0 point. Even though the "Direct" assessment of the source text led to a higher COMET score than "Random," it lagged behind "Propagation."

We thus chose the "Propagation" method in our experiments in §5, whose prediction could become even more accurate if the two components are improved. In addition, in the literature on translation quality estimation, researchers have attempted to determine source text spans corresponding to translation errors (Specia et al., 2020, 2021; Fomicheva et al., 2021). Although this line of research is out of scope in the recent series of shared tasks, we believe it is worth considering, in particular for promoting source text editing.

C.2 LLMs for Text Editing

A number of LLMs are publicly available, but the extent to which they perform the source text editing task is unknown. We considered that instruction tuning is necessary. Hence, we compared the following four LLMs that differ in the existence of language-specific adaptation and model size.

Llama-Swallow-70B: The Llama model with 70B parameters, continually pre-trained on

```
日本語文の言い換え文を出力してください。
手順は次の通りです。
1. 日本語文の¶¶で囲まれた言い換え対象表現について、同じ意味を持つ異なる表現の言い換え候補を1個から5個挙げてください。
2. 手順1で挙げた言い換え候補の中から、日本語文の¶¶で囲まれた箇所を言い換えるのに、最も適切な候補を1つ選んでください。
3. 日本語文の¶¶で囲まれた箇所を、手順2で選ばれた最適な言い換え候補を使用して置換してください。この際、以下の基準を満たすように文脈に合わせて適切に調整してください。
・文脈に応じて、適切な動詞の活用形や助詞を使うこと。
・元の文の意味を正確に伝えること。
・文法的に正しい構文を持つこと。

入力例:
日本語文: {{paraphrase["example_original"]}}
言い換え対象表現: {{paraphrase["example_original_span"]}}
出力例:
{"言い換え文":"{{paraphrase["example_paraphrase"]}}"}

日本語文: {{paraphrase["annotated_src"]}}
言い換え対象表現: {{paraphrase["propagate_error_span"]}}
```

(a) Prompt template used for Llama-Swallow to edit Japanese source text.

```
Please output a paraphrased sentence for a given English sentence.
The procedure is as follows:
1. For the target expression for paraphrasing marked with ¶¶ in the English sentence, provide 1 to 5 paraphrase candidates with the same meaning and different expressions.
2. Select one among the paraphrase candidates generated in step 1 that is most appropriate for paraphrasing the part marked with ¶¶ in the English sentence.
3. Replace the part marked with ¶¶ in the English sentence with the paraphrase candidate selected in step 2. At the same time, please perform necessary adjustment to make it fit the context while meeting the following criteria.
・Use appropriate conjugation form of words and particles according to the context.
・Convey the original meaning of the sentence accurately.
・Maintain the grammatically correct structure of the sentence.

Input example:
English sentence: {{paraphrase["example_original"]}}
Target expression for paraphrasing: {{paraphrase["example_original_span"]}}
Output example:
{"paraphrased_sentence":"{{paraphrase["example_paraphrase"]}}"}

English sentence: {{paraphrase["annotated_src"]}}
Target expression for paraphrasing: {{paraphrase["propagate_error_span"]}}
```

(b) Prompt template used for Llama to edit English source text.

Placeholder	Content to be filled
paraphrase["example_original"]	Source text of the retrieved example
paraphrase["example_original_span"]	Targeted span in the above text
paraphrase["example_paraphrase"]	Paraphrased text of the retrieved example
paraphrase["annotated_src"]	Source text to be edited
paraphrase["propagated_error_span"]	Targeted span in the above text to be edited

(c) Placeholders in the prompt templates for text editing.

Figure 5: Prompt templates used for source text editing.

a massive text data in Japanese (Fujii et al., 2024).

Llama-Swallow-8B: A smaller model obtained in the same manner as above.

Llama-70B: The 70B parameter model not specially adapted to Japanese (Grattafiori et al., 2024).

Llama-8B: A smaller model obtained in the same manner as above.

We also evaluated manual source text editing performed by the first author, which cannot be the upper bound, but is a good reference. Note that other components, including source text span detection method and prompt templates, were the same as our final method.

Figure 8 demonstrates that the four LLMs and human editor had a clear order in the COMET score

```
{{source_language}}を{{target_language}}に翻訳してください。

入力例:
英語文: {{translation["example_src"]}}
出力例:
{"翻訳文":"{{translation["example_tgt"]}}"}

英語文: {{translation["src"]}}
```

(a) Prompt template used for translation with Llama-Swallow.

```
Please translate the {{source_language}} sentence into {{target_language}}.

Input example:
Japanese sentence: {{translation["example_src"]}}
Output example:
{"translation":"{{translation["example_tgt"]}}"}

{{source_language}} sentence: {{translation["src"]}}
```

(b) Prompt template used for translation with Llama.

Placeholder	Content to be filled
source_language	Source language (e.g., "Japanese")
target_language	Target language (e.g., "English")
translation["example_src"]	Source text of the retrieved translation example
translation["example_tgt"]	Target text of the retrieved translation example
translation["src"]	Source text to be translated

(c) Placeholders in the prompt templates for translation.

Figure 6: Prompt templates used for translation.

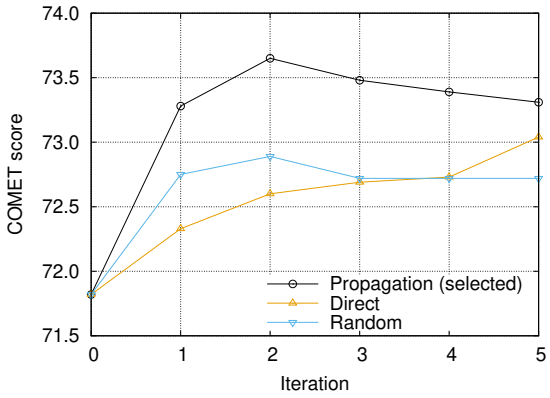

Figure 7: COMET scores achieved by different source text span detection methods.

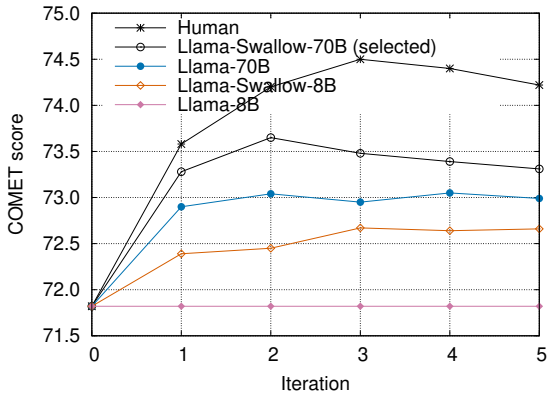

Figure 8: COMET scores achieved by different source text editors.

with any value of N up to 5. When we compared LLMs with the same sizes, Llama-Swallow outperformed Llama, which confirms the benefit of adaptation to the language of interest. We also found that smaller LLMs were not as good as their larger counterpart. Although some coincidences could exist, the non-adapted small LLM, i.e., Llama-8B, did not improve the COMET score at all. Following the results, we used Llama-Swallow-70B for editing source texts in Japanese, and analogously Llama-70B for English (§5.1.1).

At the time of this preliminary experiment, we required a sufficiently large model. However, recent advances in smaller language models will lead to a better balance of cost and benefit.

D Erroneous Test Subsets

Our method attempts to avoid translation errors. Thus, we mainly evaluated and analyzed our method focusing on the erroneous test subsets determined using XCOMET-XL (§5.1.3).

Table 5 summarizes the number of lines containing at least one "critical," "major," or "minor" error when translating with each MT system.

MT System	Ja→En				Ja→Zh	En→Ja				En→Zh
	ASPEC [a]	WMT23 [b]	MTNT19 [c]	KFTT [d]	ASPEC [e]	ALT [f]	WMT23 [g]	MTNT19 [h]	IWSLT [i]	IWSLT [j]
NLLB [1]	1,233	1,332	766	961	2,026	823	1,638	1,077	1,097	1,207
JParaCrawl (small) [2]	1,314	1,544	894	1,069	-	900	1,817	1,225	1,215	-
JParaCrawl (base) [3]	1,312	1,514	888	1,065	-	879	1,802	1,214	1,211	-
JParaCrawl (big) [4]	1,224	1,432	862	1,077	-	844	1,755	1,225	1,163	-
Llama (BM25) [5]	1,261	1,321	795	1,037	2,003	767	1,630	1,126	1,114	1,264
Llama (vector) [6]	1,253	1,331	789	1,040	1,979	748	1,612	1,110	1,101	1,272
Llama-Swallow (BM25) [7]	1,141	1,262	732	1,029	1,940	729	1,538	1,072	1,064	1,295
Llama-Swallow (vector) [8]	1,117	1,250	733	1,035	1,914	716	1,525	1,066	1,046	1,306
All	1,812	1,992	1,110	1,160	2,107	1,018	2,074	1,392	1,452	1,459

Table 5: Number of lines containing "critical," "major," or "minor" errors in the baseline translation detected by XCOMET-XL.

MT System	Ja→En				Ja→Zh	En→Ja				En→Zh
	ASPEC [a]	WMT23 [b]	MTNT19 [c]	KFTT [d]	ASPEC [e]	ALT [f]	WMT23 [g]	MTNT19 [h]	IWSLT [i]	IWSLT [j]
NLLB [1]	0.492	0.528	0.609	0.614	0.329	0.389	0.357	0.392	0.357	0.392
JParaCrawl (small) [2]	0.476	0.489	0.591	0.682	-	0.427	0.349	0.404	0.292	-
JParaCrawl (base) [3]	0.442	0.527	0.553	0.634	-	0.354	0.346	0.353	0.296	-
JParaCrawl (big) [4]	0.518	0.527	0.523	0.741	-	0.429	0.331	0.425	0.296	-
Llama (BM25) [5]	0.582	0.545	0.481	0.663	0.386	0.581	0.400	0.453	0.373	0.448
Llama (vector) [6]	0.530	0.576	0.578	0.637	0.368	0.617	0.441	0.440	0.466	0.436
Llama-Swallow (BM25) [7]	0.762	0.617	0.682	0.753	0.534	0.356	0.428	0.381	0.465	0.493
Llama-Swallow (vector) [8]	0.703	0.638	0.646	0.682	0.519	0.428	0.409	0.395	0.433	0.418

Table 6: Pearson product-moment correlation coefficients r between the COMET gain (wmt22-comet-da, with reference) and QE score gain (XCOMET-XL, without reference) both achieved by the proposed method. See Table 5 for the number of segments in each configuration.

E Correlation between the Estimated Quality and COMET Score

Table 6 summarizes the segment-level correlation coefficients between the gain of the COMET score and the gain of the estimated quality. At a glance, a moderate positive correlation existed for most configurations, and one might consider that this indicates that the estimated quality was beneficial to the search. However, as observed in §6.2, the estimated score does not always help the system make the correct decision.

F Source Text Edit Rate

Figures 9, 10, and 11 show the degree of text editing performed by the word-substitution, sequence-to-sequence, and LLM-NT methods, respectively, measured by TER computed in the same manner as for our method (§6.3). Compared with our method, the word-substitution method led to a lower TER, since it substituted only one word. The other two non-targeted methods indiscriminately affected the entire texts, which led to a substantially larger TER depending mainly on the dataset: the sequence-to-sequence method for Japanese and the LLM-NT method for English. LLM-NT demonstrated a saturation, indicating that it properly followed the instruction for retaining semantics and grammaticality.

G Details of the Sequence-to-Sequence Paraphrasing Models

Only the monolingual sequence-to-sequence pre-editing models were trained by us for our experiments. Our procedure below follows that in a previous study (Koretaka et al., 2023).

First, we generated synthetic monolingual parallel data from the bilingual parallel corpora used for retrieving translation demonstrations (§5.1.2). We randomly sampled text pairs from each corpus, aligning their sizes with the minimum corpus of 4.6M, and translated their non-targeted side into the language on the other side, using M2M100-418M (Fan et al., 2021) and beam search with a beam size of 12. The synthetic monolingual parallel data, 9.2M text pairs for each of Japanese and English, were composed of the pairs of the resulted translation (Ja'←{En,Zh} and En'←{Ja,Zh}) and the corresponding reference translation in the bilingual parallel corpus (Ja–{En,Zh} and En–{Ja,Zh}).

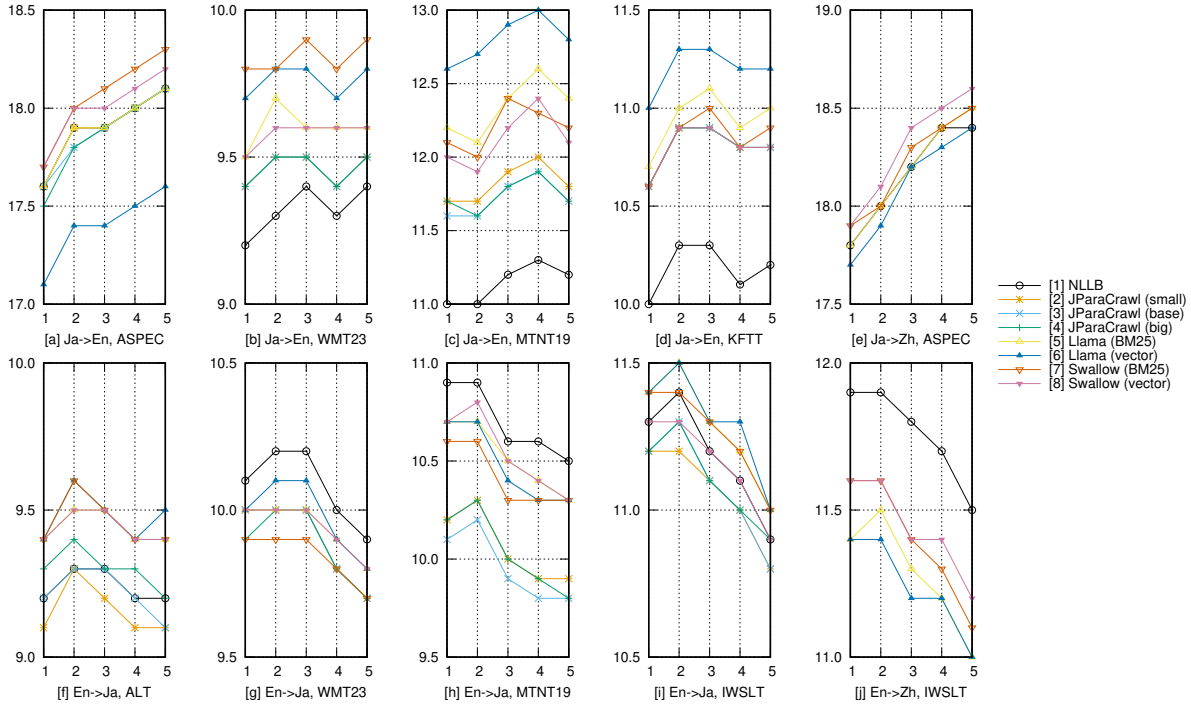

Figure 9: Translation edit rate (TER) between the original source text src_0 and each of its edited versions src_i generated by the Word-Sub method (the five-best outputs).

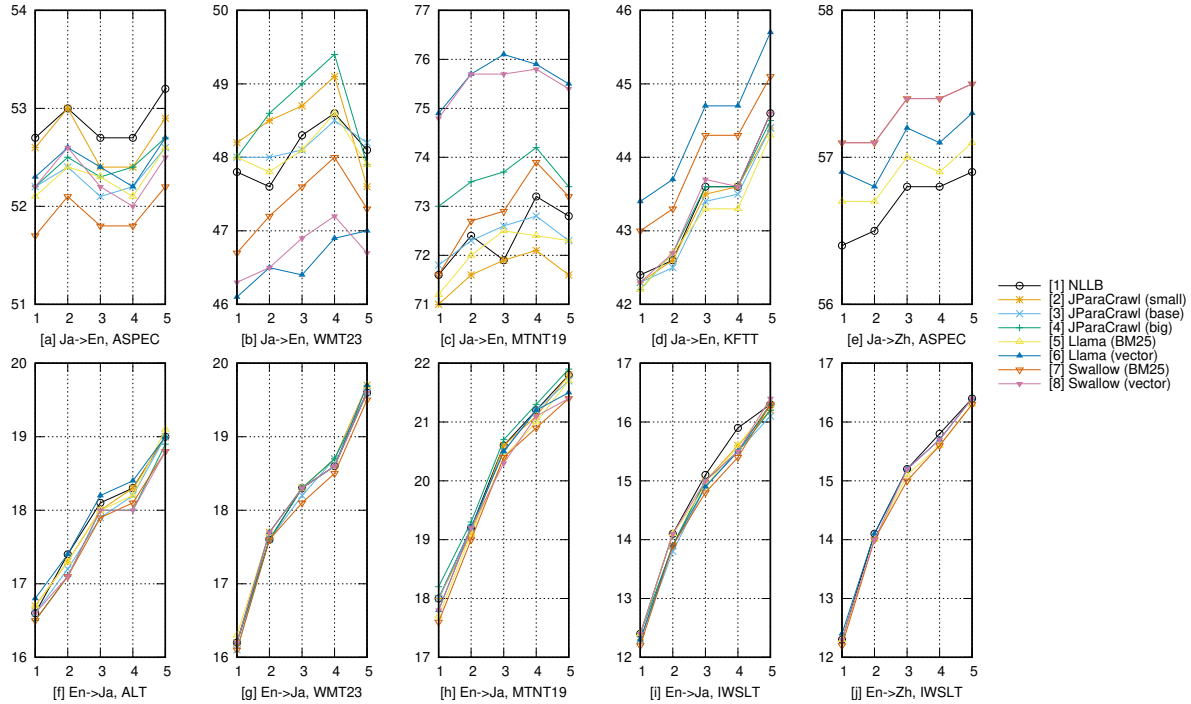

Figure 10: Translation edit rate (TER) between the original source text src_0 and each of its edited versions src_i generated by the Seq2seq-B method (the five-best outputs).

We then trained a Transformer Base model (Vaswani et al., 2017) for each of Japanese and English, regarding the synthetic side as the source, and using a joint vocabulary of 32k sub-words determined using SentencePiece (Kudo and Richardson, 2018) and Fairseq (Ott et al., 2019). We set the training hyper-parameters to the same values as Morishita et al. (2022), except for the number of updates of 60k and the lack of checkpoint averaging.

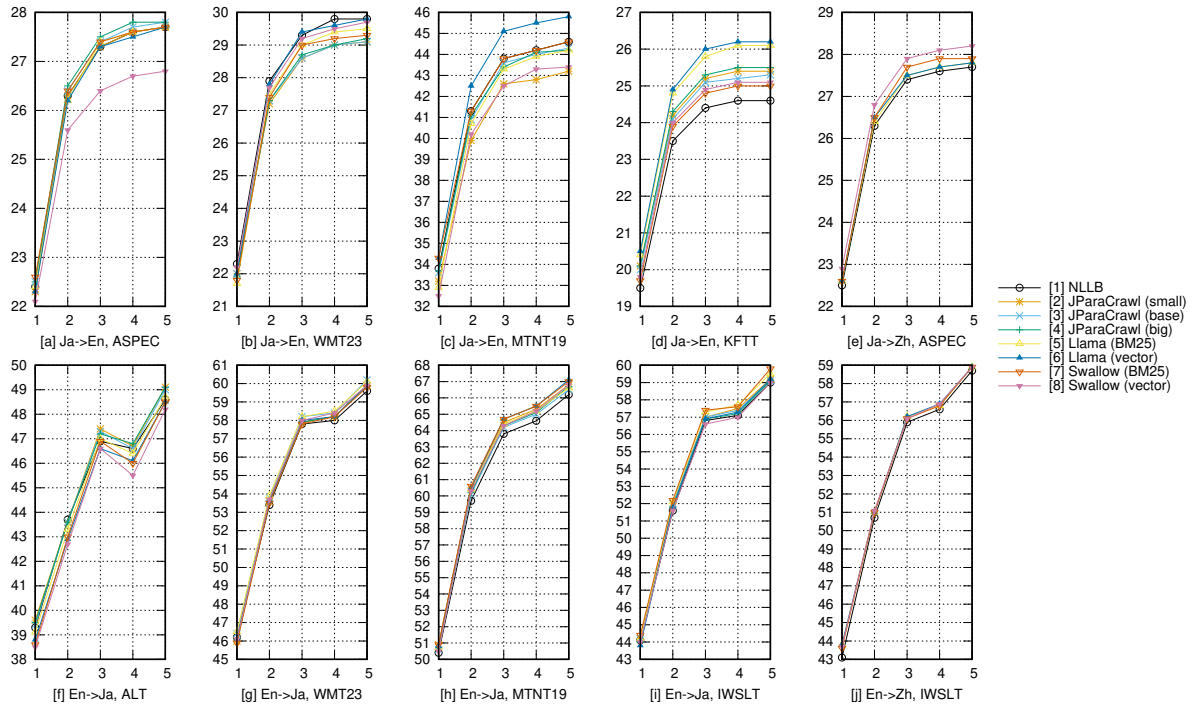

Figure 11: Translation edit rate (TER) between the original source text src_0 and each of its edited versions src_i generated by the LLM-NT method. Because of the iterative nature, src_{i+1} was always obtained from src_i directly.