Two Intermediate Translations Are Better Than One: Fine-tuning LLMs for Document-level Translation Refinement

Anonymous ACL submission

Abstract

Recent research has shown that large language models (LLMs) can enhance translation quality through self-refinement. In this paper, we build on this idea by extending the refinement from sentence-level to document-level translation, specifically focusing on document-to-document (Doc2Doc) translation refinement. Since sentence-to-sentence (Sent2Sent) and Doc2Doc translation address different aspects of the translation process, we propose fine-tuning LLMs for translation refinement using two intermediate translations, combining the strengths of both Sent2Sent and Doc2Doc. Additionally, recognizing that the quality of intermediate translations varies, we introduce an enhanced fine-tuning method with quality awareness that assigns lower weights to easier translations and higher weights to more difficult ones, enabling the model to focus on challenging translation cases. Experimental results across ten translation tasks with LLaMA-3-8B-Instruct and Mistral-Nemo-Instruct demonstrate the effectiveness of our approach. We will release our code on GitHub.

1 Introduction

Recent research has highlighted the ability of large language models (LLMs) to improve their outputs through self-refinement (Madaan et al., 2023). In machine translation, translation refinement aims to improve the quality of translations by refining intermediate results. For instance, Chen et al. (2024b) use GPT for translation refinement, designing simple prompts to support iterative enhancements. Similarly, Raunak et al. (2023) employ a chain of thought (CoT) strategy to provide natural language descriptions of suggested changes to the translation. Koneru et al. (2024) further expand the task by leveraging document-level context for better refining current sentences.

Different from above studies, in this paper we extend the translation refinement from sentence-level to document-level, refining the translations of all sentences within a document in one go. A document's translation can usually be generated either by a sentence-to-sentence (Sent2Sent) system or a document-to-document (Doc2Doc) system. However, Sent2Sent translation, which lacks document-level context, often faces discourse-related issues such as lexical inconsistency and coherence problems. For example, as shown in Figure 1, the word "跑步机/pao_bu_ji" in the source document is translated as both *running machine* and *treadmill* in the Sent2Sent translation. Additionally, translating "但即使/dan_ji_shi" as *even if* hurts coherence by ignoring the discourse relationship between sentences #2 and #3. On the other hand, while Doc2Doc translation can alleviate these discourse-related issues by incorporating both source- and target-side document-level context, it often suffers from under-translation, where phrases, clauses, or even entire sentences are omitted. For instance, in the Doc2Doc translation shown in Figure 1, the verb phrase "进入新领域/jin_ru_xin_ling_yu" from the source document is completely omitted in the translation. Taking Chinese-to-English

Source Document
#1 竞争就像是一台跑步机/pao_bu_ji。
#2 如果你呆在原地，就会被送下跑步机/pao_bu_ji。
#3 但即使/dan_ji_shi 你跑起来，你也无法真正跨出跑步机/pao_bu_ji，进入新领域/jin_ru_xin_ling_yu

Sent2Sent Translation
#1 Competition is like a running machine.
#2 If you stay where you are, you will be taken away from the treadmill.
#3 Even if you do run, you can't truly step outside the treadmill, into new territory.

Doc2Doc Translation
#1 Competition is like a treadmill.
#2 If you stand still, you get thrown off.
#3 But even if you run, you can never really get off the treadmill.

Our Translation Refinement
#1 Competition is like a treadmill.
#2 If you stand still, you get thrown off.
#3 But even if you run, you can't really step off the treadmill, into new territory.

Figure 1: An example of a source document and its Sent2Sent and Doc2Doc translations.

System	Coh.	LTCR	ALTI+
Sent2Sent	54.98	46.32	59.32
Doc2Doc	56.21	50.00	58.66

Table 1: Performance comparison between Sent2Sent and Doc2Doc Chinese-to-English translations.

document-level translation as example, Table 1 compares the performance between Sent2Sent and Doc2Doc translations of LLaMA3-8B-Instruct.[1] It shows that Doc2Doc translation achieves better performance in discourse-related metrics, Coherence and LTCR (Lyu et al., 2021; Dale et al., 2023b), while Sent2Sent translation is better in ALTI+ (Dale et al., 2023a) which detects hallucinated translation and undertranslation.

Therefore, we conjecture that refining document-level translation over two intermediate translations from both Sent2Sent and Doc2Doc systems can leverage the strengths of each, thereby mitigating the issues discussed above. For a *source* document, we prompt an existing LLM to generate Sent2Sent and Doc2Doc translations, referred to as the *sent2sent* and *doc2doc* translations, respectively. We then create a document-level refinement quadruple (*source, sent2sent, doc2doc, reference*), where *reference* serves as a naturally refined translation. When fine-tuning the LLM, we propose an enhanced fine-tuning with quality awareness that differentiates instances based on the difficulty of refinement by expanding above quadruple into a quintuple (*source, sent2sent, doc2doc, quality, reference*). The enhanced fine-tuning with quality awareness is aimed to address the varying difficulty of refining translations at sentence- and document-level. By incorporating a quality score as an additional factor during fine-tuning, it helps the model prioritize and output a better translation with differing refinement inputs. Extensive experiments on two popular LLMs show the effectiveness of our approach across ten $X \leftrightarrow$ En document-level translation tasks.

Overall, our main contributions in this work can be summarized as follows:

- We extend translation refinement from the traditional sentence-level to the document-level, and further expand it by refining two intermediate translations rather than just one.
- We introduce an enhanced fine-tuning with quality awareness, which differentiates instances based on the difficulty of refinement.
- Experimental results on two popular LLMs across ten $X \leftrightarrow$ En document-level translation tasks demonstrate that refining two intermediate translations outperforms refining from a single translation.

[1]Detailed experimental settings and the metrics can be found in Section 3.

2 Methodology

Unlike previous studies that fine-tune LLMs for translation tasks using sentence-level or document-level parallel datasets, our approach focuses on document-level translation refinement. The goal is to improve existing document-level translations by aligning them with a reference translation. Specifically, to harness the translation diversity between Sent2Sent and Doc2Doc translations, we introduce document-level translation refinement with two intermediates, with the reference as the target. A key distinction of our work emphasizes document-level translation refinement, rather than direct translation or sentence-level refinement, setting it apart from previous LLM-based translation or refinement.

As shown in Figure 2, we develop our document-level refinement LLMs in two steps:

- Fine-Tuning Data Preparation (Section 2.1): For each source-side document in the fine-tuning set, we generate two versions of its translation: one using Sent2Sent translation and the other using Doc2Doc translation.
- Enhanced Fine-Tuning with Quality Awareness (Section 2.2): Using the prepared fine-tuning data, we fine-tune LLMs in two stages: a naïve fine-tuning stage followed by the other stage with a quality-aware strategy.

Finally, Section 2.3 describes the inference.

2.1 Fine-Tuning Data Preparation

We use $(\mathbf{s}, \mathbf{r})$ to denote a document-level parallel in the fine-tuning data, where $\mathbf{s} = [s_1, \cdots, s_N]$, $\mathbf{r} = [r_1, \cdots, r_N]$, and N is the number of sentences in the document pair. Firstly, we use LLM $\mathcal{M}_S$ to generate sentence-level translation $\mathbf{y} = [y_1, \cdots, y_N]$ by translating sentences within $\mathbf{s}$ individually. This is done using the prompt template outlined in Figure 3 (a). Then we again use the LLM to generate document-level translation $\mathbf{z} = [z_1, \cdots, z_N]$ by viewing the sentences within the document as a

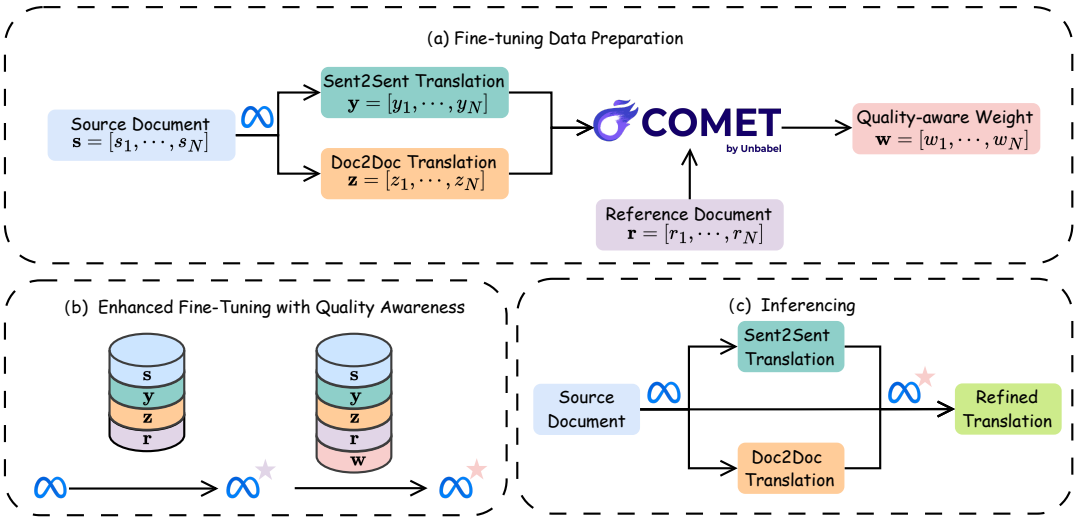

Figure 2: Illustration of our approach.

long sequence. As illustrated in the Figure 3 (b), we follow Li et al. (2024) to organize the sentences within a document by inserting markers # id between neighbouring sentences, which indicate their respective positions. Naturally, both $\mathbf{y}$ and $\mathbf{z}$ are of lower quality compared to the reference $\mathbf{r}$. Therefore, we use $\mathbf{r}$ as the target for refinement, as Feng et al. (2024a). Till now, we obtain a document-level refinement quadruple $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{r})$.

Sentence-level Quality-aware Weight. For two sentences s_i and s_j in document $\mathbf{s}$, the difficulty of refining their translations can vary, depending on the quality of their respective translations y_i/z_i and y_j/z_j. Based on the definition in Feng et al. (2024a), *easy* translations differ significantly from the reference, providing the most room for refinement. In contrast, *hard* translations are nearly perfect, with minimal differences, making them the most difficult to refine. As a result, we assign lower weights to easy translations and higher weights to hard translations. Specifically, for sentence s_i and its two translations y_i and z_i, we use reference-based sentence-level COMET to evaluate the translation quality and compute the weight as follows:

$$w_i = 1 + \lambda(\max(\mathtt{DA}(s_i, y_i, r_i), \mathtt{DA}(s_i, z_i, r_i)) - \epsilon), \quad (1)$$

where λ and ϵ are the hyper-parameters, and DA is computed using reference-based COMET[2] (Rei et al., 2022a). Consequently, we expand a document-level refinement quadruple into a quintuple $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{r})$, where $\mathbf{w} = [w_1, \cdots, w_N]$ represents sentence-level quality-aware weights.[3]

Preventing Position Bias. Figure 3 (c) shows the prompt template for document-level translation refinement. To prevent position bias, where LLMs might learn to refine translations based on specific positions (Liu et al., 2023), the placeholder <*hyp1*> can represent either the sentence-level translation $\mathbf{y}$ or the document-level translation $\mathbf{z}$, with the other translation in <*hyp2*>. This design generates two fine-tuning instances from the quintuple $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{r})$. For illustration, we refer to the quintuple as $(\mathbf{x}, \mathbf{h_1}, \mathbf{h_2}, \mathbf{w}, \mathbf{r})$, where $\mathbf{h_1}$ and $\mathbf{h_2}$ denote the two intermediate translations in the template.

2.2 Enhanced Fine-Tuning with Quality Awareness

For better leveraging the fine-tuning dataset, we propose an enhanced fine-tuning with quality awareness, where we fine-tune the LLM $\mathcal{M}_\mathcal{T}$ in two stages upon the same fine-tuning dataset. In the first stage, we perform naïve fine-tuning that does not make difference among fine-tuning instances while in the second stage, we continue to fine-tune the LLM with a quality-aware strategy. The prompt template for the fine-tuning in both stages is shown in Figure 3 (c).

Naïve Fine-Tuning. In this stage, the LLM $\mathcal{M}_\mathcal{T}$ is fine-tuned on the fine-tuning set $\mathcal{T}$ to minimize

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

[3] We provide comparison to two other weight variants in Appendix D.

the following cross-entropy loss function:

$$\begin{aligned}\mathcal{L}_1(\mathcal{T}) &= -\sum_{q\in\mathcal{T}} \log P\left(\mathbf{r}|\mathcal{P}\left(\mathbf{s},\mathbf{h_1},\mathbf{h_2}\right)\right) \\ &= -\sum_{q\in\mathcal{T}}\sum_{i=1}^{N} \log P\left(r_i|\mathcal{P}\left(\mathbf{s},\mathbf{h_1},\mathbf{h_2}\right),r_{<i}\right),\end{aligned} \tag{2}$$

where q denotes a quintuple $(\mathbf{x},\mathbf{h_1},\mathbf{h_2},\mathbf{w},\mathbf{r})$, $\mathcal{P}\left(\mathbf{s},\mathbf{h_1},\mathbf{h_2}\right)$ returns the prompt defined by the template, $r_{<i}$ represents the previous sentences before r_i in $\mathbf{r}$. In this stage, all sentences in the reference document $\mathbf{r}$ are assigned equal weights, specifically a weight of 1.

Quality-aware Fine-Tuning. In this stage, we continue to fine-tune $\mathcal{M}_T$ on $\mathcal{T}$ using a quality-aware strategy, achieved by assigning quality-aware weights to the sentences in the reference $\mathbf{r}$ when calculating the loss function:

$$\begin{aligned}\mathcal{L}_2(\mathcal{T}) &= -\sum_{q\in\mathcal{T}} \mathbf{w}\log P\left(\mathbf{r}|\mathcal{P}\left(\mathbf{s},\mathbf{h_1},\mathbf{h_2}\right)\right) \\ &= -\sum_{q\in\mathcal{T}}\sum_{i=1}^{n} w_i \log P\left(r_i|\mathcal{P}\left(\mathbf{s},\mathbf{h_1},\mathbf{h_2}\right),r_{<i}\right).\end{aligned} \tag{3}$$

Specifically, all tokens within a reference sentence r_i have the same weight w_i. And we refer to the fine-tuned LLM as $\mathcal{M}_T^*$.

2.3 Inferencing

Once fine-tuning the LLM $\mathcal{M}_T^*$ is complete, we use it to refine translations on the test sets. As shown in Figure 2 (c), we first prompt $\mathcal{M}_S$ to generate both Sent2Sent and Doc2Doc translations. Then, for each source document, the two intermediate translations are fed into $\mathcal{M}_T^*$ for refinement. During inferencing, quality-aware weights are not needed.

3 Experimentation

3.1 Experimental Settings

Datasets. Following Li et al. (2024), to avoid data leakage (Garcia et al., 2023), we utilize the latest News Commentary v18.1[4], which features parallel text with document boundaries. We conduct our experiments on five language pairs in both directions: English (En) ↔ German (De), English (En) ↔ Russian (Ru), English (En) ↔ Spanish (Es), English (En) ↔ Chinese (Zh), and English (En) ↔ French (Fr). For each language pair, 150 documents are randomly selected as the development set, and another 150 documents as the test set. See Table 8 in Appendix A for more details.

[4] https://www2.statmt.org/wmt24/translation-task.html

(a) Sent2Sent Translation

Translate this document from <*src_lang*> into <*tgt_lang*>.
Don't give any explanation.
<*src_lang*> Source: <*sent_src*>
<*tgt_lang*> Translation:

(b) Doc2Doc Translation

Translate this document from <*src_lang*> into <*tgt_lang*>.
Don't give any explanation.
Each sentence is separated by #id.
<*src_lang*> Source: <*doc_src*>
<*tgt_lang*> Translation:

(c) Translation Refinement

You are an expert in editing translations.
Given a <*src_lang*> source text and two <*tgt_lang*> translated versions, please produce an improved translated version by drawing upon the strengths of both initial translations.
Don't give any explanation.
Each sentence is separated by #id.
<*src_lang*> Source: <*doc_src*>
<*tgt_lang*> Translation 1: <*hyp1*>
<*tgt_lang*> Translation 2: <*hyp2*>
<*tgt_lang*> Translation Refinement:

Figure 3: Prompt template used for translation and refinement.

Models and Settings. We select LLaMA-3-8B-Instruct[5] (Meta, 2024) and Mistral-Nemo-Instruct[6] (MistralAI, 2024) as the foundation open-source LLMs for applying prompt engineering (i.e., $\mathcal{M}_S$) and quality-aware fine-tuning (i.e., $\mathcal{M}_T$).[7] During fine-tuning, we adopt QLoRA (Dettmers et al., 2023), a quantized version of LoRA (Hu et al., 2021). For the hyper-parameters in Eq. 1, we set λ to 3.75 and ϵ to 0.7, respectively. During inference, to ensure reproducibility, we set `do_sample` to false. For detailed fine-tuning and hyper-parameter settings, please refer to Appendix B and C.

Baselines. We compare our approach to several translation baselines:

- Sent2Sent: As described in Section 2.1, we prompt $\mathcal{M}_S$ to generate sentence-level translation. In a contrastive setting, we first fine-tune $\mathcal{M}_S$ at sentence-level translation and then obtain sentence-level translation, referred as $\text{Sent2Sent}_{\text{tuned}}$.

[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[6] https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407

[7] For simplicity, we treat $\mathcal{M}_S$ and $\mathcal{M}_T$ as the same LLM, unless otherwise specified.

- Doc2Doc: As described in Section 2.1, we prompt $\mathcal{M}_S$ to generate document-level translation. Similarly, Doc2Doc$_{\text{tuned}}$ refers to document-level translation from fine-tuned $\mathcal{M}_S$ at document-level translation.

- SentRefine$_{\text{sent}}$: It is sentence-level translation refinement by fine-tuning $\mathcal{M}_T$ on Sent2Sent, similar to Chen et al. (2024b).

- DocRefine$_{\text{sent}}$: It is document-level translation refinement by fine-tuning $\mathcal{M}_T$ on Sent2Sent, similar to Koneru et al. (2024).

- DocRefine$_{\text{doc}}$: It is also document-level translation refinement by fine-tuning $\mathcal{M}_T$ on Doc2Doc.

Note that SentRefine$_{\text{sent}}$, DocRefine$_{\text{sent}}$ and DocRefine$_{\text{doc}}$ all use one intermediate translation. Please refer to Table 10 in Appendix E for detailed prompts. Differently, our approach uses both Sent2Sent and Doc2Doc as intermediate translations. For all document-level translation or refinement output, we use `Bertalign` (Liu and Zhu, 2023) to recover sentence-level translation.

Evaluation Metrics. We report document-level COMET (d-COMET) scores proposed by Vernikos et al. (2022). Specifically, we apply reference-based metric `wmt22-comet-da`[8] (Rei et al., 2022a). For other tranditional evaluation metrics, including sentence-level COMET (s-COMET), document-level BLEU (d-BLEU), please refer to Appendix F.

Besides, we also report several additional metrics. 1) We follow Li et al. (2023) and Su et al. (2022) to compute coherence score (Coh.) using cosine similarity between the sentence embeddings of SimCSE (Gao et al., 2021) of the neighbouring sentences. 2) We report ALTI+ score (Ferrando et al., 2022; Dale et al., 2023a) to detect under-translation and hallucination issues in translation. 3) We follow Lyu et al. (2021) and compute LTCR score to measure lexical translation consistency. 4) We compute document-level perplexity (PPL) using GPT-2[9] (Radford et al., 2019). 5) We report BlonDe (Jiang et al., 2022), which evaluates discourse phenomena via a set of automatically extracted features (Deutsch et al., 2023). Except for ALTI+, these metrics are document-level discourse-related metrics. LTCR, BlonDe, and perplexity are computed only for the $X \rightarrow$ En translation direction, while the other two metrics are applicable to all translation directions.

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

[9] https://huggingface.co/openai-community/gpt2

3.2 Main Results

Table 2 presents the performance comparison in d-COMET. From it, we observe:

- Extending the translation unit from sentence-level to document-level improves overall performance, as Doc2Doc outperforms Sent2Sent. This aligns with findings from related studies (Karpinska and Iyyer, 2023). However, the fine-tuned LLMs exhibit different performance trends. LLaMA-3-8B-Instruct shows similar performance for both Sent2Sent$_{\text{tuned}}$ and Doc2Doc$_{\text{tuned}}$, while Mistral-Nemo-Instruct performs better with Doc2Doc$_{\text{tuned}}$ compared to Sent2Sent$_{\text{tuned}}$.

- Refining with a single input, whether from Sent2Sent or Doc2Doc, leads to higher COMET scores. However, this refinement shows little to no improvement over the performance of directly fine-tuned LLMs.

- Our refinement approach, based on the two intermediate translations Sent2Sent and Doc2Doc, significantly improves translation performance across all language pairs. It achieves COMET score improvements of 2.73 and 1.80 on LLaMA-3-8B-Instruct, and 2.21 and 1.79 on Mistral-Nemo-Instruct. Our approach also outperforms other baselines, including both refining with single translations and directly fine-tuning, demonstrating the effectiveness of our proposed approach.

- Lastly, disabling the quality-aware fine-tuning stage results in a performance drop, highlighting the effectiveness of our fine-tuning strategy. Additionally, compared to SentRefine$_{\text{sent}}$, DocRefine$_{\text{sent}}$, and DocRefine$_{\text{doc}}$, refinement using two intermediate translations outperforms refinements with just one.

Table 3 presents the performance on several additional metrics when LLaMA-3-8B-Instruct is used. The results show that, except for ALTI+, document-level translation and refinement systems outperform their sentence-level counterparts. By combining the strengths of Sent2Sent and Doc2Doc translations, our approach achieves the best performance across all five metrics.

System	X→En					En→X					Avg.
	De→	Es→	Ru→	Fr→	Zh→	→ De	→ Es	→ Ru	→ Fr	→ Zh	
LLaMA-3-8B-Instruct											
Sent2Sent	85.97	86.62	81.63	84.43	82.18	82.50	85.02	80.97	82.89	76.80	82.90
Sent2Sent$_{\text{tuned}}$	87.94	87.46	81.98	86.46	84.18	85.42	86.11	80.88	84.30	82.84	84.76
Doc2Doc	87.05	87.21	81.07	85.40	83.60	83.35	85.36	80.18	83.14	81.89	83.83
Doc2Doc$_{\text{tuned}}$	87.82	88.04	81.25	86.37	84.88	85.45	85.61	81.06	84.63	82.18	84.73
SentRefine$_{\text{sent}}$	83.70	87.99	82.64	85.98	84.08	85.21	86.34	83.74	84.57	82.93	84.72
DocRefine$_{\text{sent}}$	87.42	87.98	81.16	86.56	85.06	85.38	86.32	80.39	84.43	82.61	84.73
DocRefine$_{\text{doc}}$	87.71	88.06	82.73	86.32	84.99	85.07	86.49	83.16	84.73	82.70	85.19
Ours	**88.14**	**88.42**	**82.75**	**86.69**	**85.39**	**86.05**	**86.86**	**83.85**	**84.84**	**83.35**	**85.63**
- QA Fine-tuning	88.02	88.35	82.63	86.53	85.09	85.70	86.60	83.17	84.48	82.98	85.36
Mistral-Nemo-Instruct											
Sent2Sent	86.85	87.21	82.86	85.27	83.82	84.66	85.47	83.78	83.67	79.39	84.30
Sent2Sent$_{\text{tuned}}$	86.86	86.89	83.33	85.79	83.96	85.49	85.77	84.58	84.49	81.18	84.83
Doc2Doc	87.61	87.64	82.60	85.95	84.55	84.34	85.14	84.34	83.66	81.34	84.72
Doc2Doc$_{\text{tuned}}$	87.80	88.34	82.60	86.39	85.16	86.50	86.72	85.68	85.28	81.27	85.57
SentRefine$_{\text{sent}}$	87.73	88.23	83.87	86.23	84.71	86.36	86.48	85.63	85.06	81.27	85.56
DocRefine$_{\text{sent}}$	88.09	88.50	82.34	86.21	85.40	86.58	86.91	84.67	85.09	84.06	85.79
DocRefine$_{\text{doc}}$	88.13	88.37	81.65	86.41	85.20	86.44	86.95	83.90	85.11	83.86	85.61
Ours	**88.45**	**88.99**	**84.59**	**87.00**	**85.83**	**86.89**	**87.31**	**85.99**	**85.50**	**84.53**	**86.51**
- QA Fine-tuning	88.01	88.27	83.89	86.40	85.37	86.70	86.94	85.34	85.43	83.86	86.02

Table 2: Performance in document-level COMET (d-COMET) score. Bold scores represent the highest performance, while underlined scores indicate the second-best performance. *-QA Fine-tuning* indicates disabling the quality-aware fine-tuning stage.

System	Coh. ↑	ALTI+ ↑	LTCR ↑	PPL ↓	BlonDe ↑
Sent2Sent	56.17	42.57	57.23	32.86	48.49
Sent2Sent$_{\text{tuned}}$	56.23	42.94	60.45	30.34	58.61
Doc2Doc	62.28	40.04	61.25	31.85	51.30
Doc2Doc$_{\text{tuned}}$	63.42	42.99	64.99	31.58	57.86
SentRefine$_{\text{sent}}$	64.27	43.09	60.08	32.14	57.47
DocRefine$_{\text{sent}}$	64.95	43.00	63.62	30.13	58.69
DocRefine$_{\text{doc}}$	65.09	42.80	63.68	31.62	59.01
Ours	**67.12**	**43.53**	**66.57**	**26.51**	**59.86**
- QA Fine-tuning	66.07	43.06	65.98	31.64	59.57

Table 3: Averaged performance of LLaMA-3-8B-Instruct in additional metrics.

4 Discussion

In this section, we use LLaMA-3-8B-Instruct as the representative LLM, unless otherwise noted.

4.1 Refining Translations of GPT

To further evaluate our approach, we use our fine-tuned LLMs to refine translations from GPT-4o-mini (OpenAI, 2024). As shown in Table 4, both sentence-level and document-level refinements with one intermediate translation show limited improvement (i.e., #4/#5 vs. #2). In contrast, refining with two intermediate translations yields a 0.22 COMET score improvement (i.e., #6 vs. #2), suggesting that using two intermediate translations is more effective. Our two fine-tuned systems behave differently: LLaMA-3-8B-Instruct experiences a slight drop (85.62 to 85.46), while Mistral-Nemo-Instruct successfully improves performance from 85.62 to 86.31. For detailed s-COMET scores, please refer to Table 13 in Appendix F.

4.2 Effect of Enhanced Fine-tuning with Quality Awareness

Table 5 compares the performance on En↔De and En↔Zh directions for various fine-tuning strategies. It shows by removing either the naïve or the quality-aware fine-tuning stage decrease the performance. Meanwhile, replacing the quality-aware fine-tuning stage with naïve one may cause a performance drop, indicating that each stage in our enhanced fine-tuning with quality awareness contributes to the overall performance, which can effectively alleviate overfitting to further enhance generalization.

4.3 Effect of Preventing Position Bias

To prevent introducing position bias, <*hyp1*> in the prompt template can be either Sent2Sent or Doc2Doc translation. To examine its effect, we compare it with a version where <*hyp1*> is always set to Sent2Sent and <*hyp2*> is set to Doc2Doc.

#	System	X→En					En→X					Avg.
		De→	Es→	Ru→	Fr→	Zh→	→ De	→ Es	→ Ru	→ Fr	→ Zh	
1	GPT Sent2Sent	86.49	86.53	82.43	84.73	83.98	85.96	86.52	85.28	84.97	83.70	85.06
2	GPT Doc2Doc	87.00	87.12	83.71	85.64	84.75	86.30	86.76	85.59	85.23	84.07	85.62
3	GPT SentRefine$_{sent}$	86.86	86.89	83.37	83.70	83.33	85.32	86.43	85.42	84.30	83.99	84.96
4	GPT DocRefine$_{sent}$	87.03	87.26	83.23	85.77	84.29	86.57	87.04	86.04	85.40	84.07	85.67
5	GPT DocRefine$_{doc}$	87.04	87.29	83.27	85.63	84.41	86.37	87.03	86.14	85.43	83.93	85.62
6	GPT DocRefine$_{doc+sent}$	87.39	87.65	83.44	85.77	84.78	86.61	86.96	86.16	85.46	**84.13**	85.84
7	L-DocRefine$_{doc+sent}$	87.88	88.15	82.07	86.57	85.22	86.31	86.09	83.66	85.28	83.32	85.46
8	M-DocRefine$_{doc+sent}$	**88.14**	**88.22**	**84.39**	**86.73**	**85.48**	**86.88**	**87.20**	**86.20**	**85.69**	84.12	**86.31**

Table 4: Performance in d-COMET when refining translations from GPT-4o-mini. For the GPT-based refinement systems, we use the same prompt templates as those used in our approach, but without fine-tuning. L-* and M-* denote our fine-tuned LLaMA-3-8B-Instrcut and Mistral-Nemo-Instruct, respectively.

Stage1	Stage2	De→En	En→De	Zh→En	En→Zh
naïve	QA	**88.14**	**86.05**	**85.39**	**83.35**
naïve	-	88.02	85.70	85.09	82.98
QA	-	87.76	85.60	84.88	83.05
naïve	naïve	87.75	85.91	83.98	82.14

Table 5: Performance comparison when using different fine-tuning strategies. QA indicates quality-aware fine-tuning.

Our Approach	De→En	En→De
w/ preventing position bias	**88.14**	**86.05**
w/o preventing position bias	87.60	85.55

Table 6: Performance comparison with and without preventing position bias.

As shown in Table 6, preventing position bias leads to a significant boost in performance.

4.4 Comparison to Reranking and Reranking + Refining

To demonstrate the effectiveness of our approach in combining Sent2Sent and Doc2Doc translations, we compare it with two other strategies: 1) Reranking, which chooses the translation with the higher reference-free COMETKiwi score[10] (Rei et al., 2022b) for each source sentence (He et al., 2024; Farinhas et al., 2023); and 2) Reranking + Refining, which further refines the selected translation using DocRefine$_{doc}$ and DocRefine$_{sent}$.

As shown in Table 7, our approach outperforms the other two strategies in combining two intermediate translations. Furthermore, our approach benefits from the variety of intermediate translations, achieving the best performance when T1 and T2 are from Sent2Sent and Doc2Doc[11], respectively. This demonstrates that our approach effectively leverages the strengths of both translation types.

T1	T2	Strategy	De→En	En → De
S2S	D2D	Rerank	86.96	84.20
S2S	D2D	Rerank + Refine	87.74	85.56
S2S	D2D	Ours	**88.02**	**86.05**
S2S	S2S	Rerank	86.16	83.07
S2S	S2S	Rerank + Refine	87.63	85.58
S2S	S2S	Ours	**87.76**	**86.04**
D2D	D2D	Rerank	86.99	83.30
D2D	D2D	Rerank + Refine	87.50	85.65
D2D	D2D	Ours	**87.61**	**85.69**

Table 7: Comparison with *reranking* and *reranking + refining*. T1/T2 refers to intermediate translation 1/2.

4.5 GPT-based Error Annotating

Following Wu et al. (2024), we identify translation errors from both sentence-level and document-level perspectives. Please refer to Appendix H for our detailed prompts. Specifically, we use GPT-4o-Mini to detect sentence-level issues such as mistranslation, over-translation (including additions), and under-translation (including omissions). Additionally, we address document-level errors related to cohesion, coherence and inconsistent style (including the use of multiple terms for the same concept). Figure 4 shows the results for De→En translation. It highlights that: 1) our approach addresses all the issues observed in Doc2Doc translation; and 2) it improves most of the issues in Sent2Sent translation, with a trade-off in performance related to under-translation (including omissions). The two highlights suggest that our approach effectively combines the strengths of both Sent2Sent and Doc2Doc translations.

5 Related Work

5.1 LLM-based Translation Refinement

Current approaches to LLM-based translation refinement can be broadly categorized into two types: prompt engineering and supervised fine-tuning.

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

[11] To obtain different S2S (or D2D) translations, we set `do_sample` to true, `temperature` to 0.3 and `top_p` to 0.7.

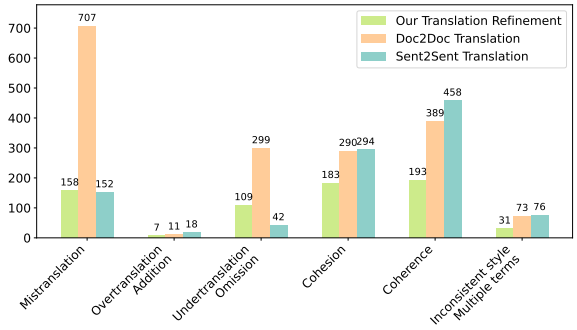

Figure 4: Counts of error types on De→En translation.

In the realm of prompt engineering, Chen et al. (2024b) propose a method where ChatGPT is iteratively prompted to self-correct translations. Raunak et al. (2023) investigate the use of GPT-4 to automatically post-edit translations produced by neural machine translation (NMT) systems. Feng et al. (2024b) introduce the *Translate-Estimate-Refine* framework, which employs LLMs for translation self-refinement. Xu et al. (2023) and Xu et al. (2024) also prompt LLMs to firstly generate an intermediate translation, and then provide self-feedback, which is used to optimize the final translation. Yang et al. (2023) explore human intervention in the inference process of LLM in MT tasks. Chen et al. (2024a) explore dual learning for translation tasks to enhance LLMs' self-reflective abilities, thereby improving translation performance. Berger et al. (2024) prompt LLMs to edit translations with human error markings. Farinhas et al. (2023) generate multi hypotheses, and then experiment on various ways to ensemble these hypotheses. All of these studies focus on sentence-level refinement.

In supervised fine-tuning approaches, Ki and Carpuat (2024) fine-tune LLMs using source sentences, intermediate translations, and error annotations. Alves et al. (2024) fine-tune LLMs for translation-related tasks, such as quality estimation (QE) and automatic post-editing (APE), and train a model called Tower-Instruct. Feng et al. (2024a) propose a hierarchical fine-tuning strategy, dividing fine-tuning instances into three groups based on refinement difficulty for multi-stage fine-tunin. These studies, like the prompt engineering approaches, also focus on sentence-level refinement. In contrast, Koneru et al. (2024) extend sentence-level refinement by incorporating document-level context. Our work builds on this idea, but goes further by focusing on document-to-document refinement, where we extend the refinement process from individual sentences to entire documents.

5.2 LLM-based Document-level Machine Translation

Current approaches to LLM-based document-level machine translation (DMT) can also be broadly categorized into two types: prompt engineering and supervised fine-tuning.

In prompt engineering, Wang et al. (2023) are the first to experiment with various prompt templates for performing DMT using GPT models. Karpinska and Iyyer (2023) analyze translation performance of GPT-3.5 on novel translation tasks, exploring how LLMs handle DMT. Cui et al. (2024) apply retrieval-augmented generation (RAG), leveraging contextual summaries to select the most relevant examples from a database, thereby improving translation quality by incorporating additional context. Additionally, Wang et al. (2024) introduce a document-level translation agent with a multi-level memory structure, improving consistency and accuracy by better handling long-range dependencies.

On the other hand, supervised fine-tuning approaches focus on enhancing LLMs' ability to perform DMT through targeted fine-tuning. For instance, Li et al. (2024) propose a mixed fine-tuning strategy that combines sentence-level fine-tuning instructions with document-level fine-tuning to improve overall translation performance. Wu et al. (2024) introduce a multi-stage fine-tuning approach, initially fine-tuning on non-English monolingual documents and then fine-tuning with parallel documents. Lyu et al. (2024) present a decoding-enhanced, multi-phase prompt tuning method, which enables LLMs to better model and utilize both inter- and intra-sentence context, thereby improving the adaptation of LLMs to context-aware NMT.

6 Conclusion

In this paper, we have proposed a novel approach to refine Doc2Doc translation by combining the strengths of both sentence-level and document-level translations. Our approach employs an enhanced fine-tuning with quality awareness to improve the performance of large language models (LLMs). Experimental results across ten document-level translation tasks show substantial improvements in translation quality, coherence, and consistency for a variety of language pairs.

Limitations

Our experiments are primarily conducted on a news dataset, which may not fully represent LLMs' performance in other specific domains and other non-English translation directions. Moreover, we train one model for one specific translation direction, leading to huge computational cost. The model may be biased to refining texts of a specific style and may perform worse when refining texts in other styles. Further research may enhance the multilingual performance of LLMs.

References

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

Nathaniel Berger, Stefan Riezler, Miriam Exel, and Matthias Huck. 2024. Prompting large language models with human error markings for self-correcting machine translation. *CoRR*, abs/2406.02267.

Andong Chen, Lianzhang Lou, Kehai Chen, Xuefeng Bai, Yang Xiang, Muyun Yang, Tiejun Zhao, and Min Zhang. 2024a. DUAL-REFLECT: Enhancing large language models for reflective translation through dual learning feedback mechanisms. In *Proceedings of ACL (Short Papers)*, pages 693–704.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2024b. Iterative translation refinement with large language models. In *Proceedings of EACL*, pages 181–190.

Menglong Cui, Jiangcun Du, Shaolin Zhu, and Deyi Xiong. 2024. Efficiently exploring large language models for document-level machine translation with in-context learning. In *Findings of ACL*, pages 10885–10897.

David Dale, Elena Voita, Loic Barrault, and Marta R. Costa-jussà. 2023a. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better. In *Proceedings of ACL*, pages 36–50.

David Dale, Elena Voita, Janice Lam, Prangthip Hansanti, Christophe Ropers, Elahe Kalbassi, Cynthia Gao, Loic Barrault, and Marta Costa-jussà. 2023b. HalOmi: A manually annotated benchmark for multilingual hallucination and omission detection in machine translation. In *Proceedings of EMNLP*, pages 638–653.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Proceedings of NeurIPS*, pages 10088–10115.

Daniel Deutsch, Juraj Juraska, Mara Finkelstein, and Markus Freitag. 2023. Training and meta-evaluating machine translation evaluation metrics at the paragraph level. In *Proceedings of WMT*, pages 996–1013.

António Farinhas, José de Souza, and Andre Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. In *Proceedings of EMNLP*, pages 11956–11970.

Zhaopeng Feng, Ruizhe Chen, Yan Zhang, Zijie Meng, and Zuozhu Liu. 2024a. Ladder: A model-agnostic framework boosting LLM-based machine translation to the next level. In *Proceedings of EMNLP*, pages 15377–15393.

Zhaopeng Feng, Yan Zhang, Hao Li, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2024b. Tear: Improving llm-based machine translation with systematic self-refinement. *CoRR*, abs/2402.16379.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In *Proceedings of the EMNLP*, pages 8756–8769.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of EMNLP*, pages 6894–6910.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *Proceedings of ICML*, pages 10867–10878.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024. Exploring human-like translation strategy with large language models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Yuchen Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Jian Yang, Haoyang Huang, Rico Sennrich, Ryan Cotterell, Mrinmaya Sachan, and Ming Zhou. 2022. BlonDe: An automatic evaluation metric for document-level machine translation. In *Proceedings of NAACL: HLT*, pages 1550–1565.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. In *Proceedings of WMT*, pages 419–451.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. In *Findings of ACL*, pages 4253–4273.

Sai Koneru, Miriam Exel, Matthias Huck, and Jan Niehues. 2024. Contextual refinement of translations: Large language models for sentence and document-level post-editing. In *Proceedings of NAACL: HLT*, pages 2711–2725.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of ACL*, pages 12286–12312.

Yachao Li, Junhui Li, Jing Jiang, and Min Zhang. 2024. Enhancing document-level translation of large language model via translation mixed-instructions. *CoRR*, abs/2401.08088.

Lei Liu and Min Zhu. 2023. Bertalign: Improved word embedding-based sentence alignment for chinese–english parallel corpora of literary texts. *Digital Scholarship in the Humanities*, 38:621–634.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172.

Xinglin Lyu, Junhui Li, Zhengxian Gong, and Min Zhang. 2021. Encouraging lexical translation consistency for document-level neural machine translation. In *Proceedings of EMNLP*, pages 3265–3277.

Xinglin Lyu, Junhui Li, Yanqing Zhao, Min Zhang, Daimeng Wei, Shimin Tao, Hao Yang, and Min Zhang. 2024. DeMPT: Decoding-enhanced multi-phase prompt tuning for making LLMs be better context-aware translators. In *Proceedings of EMNLP*, pages 20280–20295.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/.

MistralAI. 2024. Mistral nemo. https://mistral.ai/news/mistral-nemo/.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/research/gpt-4.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Vikas Raunak, Amr Sharaf, Yiren Wang, Hany Awadalla, and Arul Menezes. 2023. Leveraging GPT-4 for automatic translation post-editing. In *Findings of EMNLP*, pages 12009–12024.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022a. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of WMT*, pages 578–585.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of WMT*, pages 634–645.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Proceedings of NeurIPS*, pages 21548–21561.

Giorgos Vernikos, Brian Thompson, Prashant Mathur, and Marcello Federico. 2022. Embarrassingly easy document-level MT metrics: How to convert any pretrained metric into a document-level metric. In *Proceedings of WMT*, pages 118–128.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. In *Proceedings of EMNLP*, pages 16646–16661.

Yutong Wang, Jiali Zeng, Xuebo Liu, Derek F. Wong, Fandong Meng, Jie Zhou, and Min Zhang. 2024. Delta: An online document-level translation agent based on multi-level memory. *CoRR*, abs/2410.08143.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari. 2024. Adapting large language models for document-level machine translation. *CoRR*, abs/2401.06468.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2024. LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback. In *Findings of NAACL: HLT*, pages 1429–1445.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of EMNLP*, pages 5967–5994.

Xinyi Yang, Runzhe Zhan, Derek F. Wong, Junchao Wu, and Lidia S. Chao. 2023. Human-in-the-loop machine translation with large language model. In *Proceedings of MTSummit*, pages 88–98.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of ACL*, pages 400–410.

A Data Statistics

Table 8 shows the detailed statistics of our training, validation and test datasets for the ten translation directions.

Dataset	#Document Train/Valid/Test	#Sentence Train/Valid/Test
De ↔ En	8.4K/150/150	333K/5.9K/6.0K
Fr ↔ En	7.9K/150/150	310K/5.9K/5.8K
Es ↔ En	9.7K/150/150	378K/5.8K/5.8K
Ru ↔ En	7.3K/150/150	279K/5.7K/5.6K
Zh ↔ En	8.6K/150/150	342K/6.0K/5.9K

Table 8: Statistics of the datasets

B Fine-Tuning and Inferencing Settings

In fine-tuning, we set LoRA rank to 8 and LoRA alpha to 16. We apply LoRA target modules to both the query and the value components. All fine-tuning experiments are conducted on 4 NVIDIA V100 GPUs. We use the AdamW optimizer and learning rate scheduler of cosine, with an initial learning rate to 1e-4, warmup ratio of 0.1, batch size of 2, gradient accumulation over 8 steps. In both stages of quality-aware enhanced fine-tuning, we train 1 epoch. During inference, following Alves et al. (2024) and Koneru et al. (2024), we set `num_beams` to 3. Our implementation is based on LLaMA-Factory Framework[12] (Zheng et al., 2024).

C Effects of Hyper-Parameters

We use the combined En ↔ De validation sets to tune two hyper-parameters: λ and ϵ. First, we explore values of ϵ in the range from 0.5 to 0.9 with a step size of 0.1. Our experiments reveal that ϵ has a minimal effect on performance, and we ultimately set ϵ to 0.7.

Next, we search for an optimal value of λ within the range of 1.0 to 5.0, using a step size of 0.5. We observe that λ values between 2.5 and 4.0 yield better performance than other values. As a result, we narrow the search for λ to the range of 2.5 to 4.0 with a finer step size of 0.25. Figure 5 illustrates the learning curve for λ values between 1.0 and 5.0, showing that $\lambda = 3.75$ achives the best performance.

Based on these findings, we set $\lambda = 3.75$ and $\epsilon = 0.7$ for all experiments.

[12] https://github.com/hiyouga/LLaMA-Factory

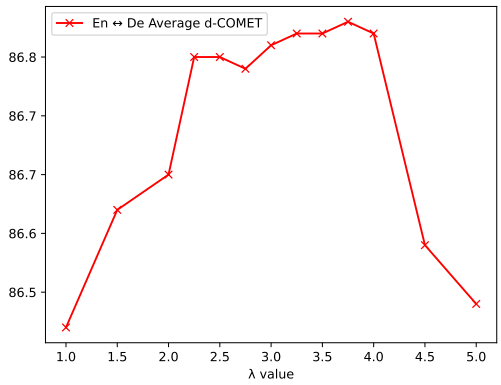

Figure 5: Performance curve on the En ↔ De validation sets for λ values ranging from 1.0 to 5.0. The optimal performance is achieved when $\lambda = 3.75$.

D Comparison to Other Two Weight Variants

In addition to using Eq. 1 to compute the sentence-level weight, we also compare it with two alternative weight variants:

- Variant 1: Instead of using the maximum DA score, we compute the weight based on h_i, which is the first translation in the prompt template (either y_i or z_i:):

$$w_i = 1 + \lambda(\texttt{DA}(s_i, h_i, r_i) - \epsilon). \quad (4)$$

- Variant 2: Rather than assigning a weight to each sentence, we assign a weight to each document. This document-level weight is computed as:

$$w = 1 + \lambda(\max(\texttt{avgDA}(s, y, r), \texttt{avgDA}(s, z, r)) - \epsilon), \quad (5)$$

where $\texttt{avgDA}(s, y, r)$ returns the averaged reference-based COMET score.

Table 9 compares the performance. It shows that our weight method outperforms the other two weight variants.

	De→En	En→De	Zh→En	En→Zh
Our	**88.14**	**86.05**	**85.39**	**83.35**
Variant 1	87.12	85.31	84.79	83.17
Variant 2	87.60	85.52	84.72	83.03

Table 9: Performance comparison when using different equations to calculate weights.

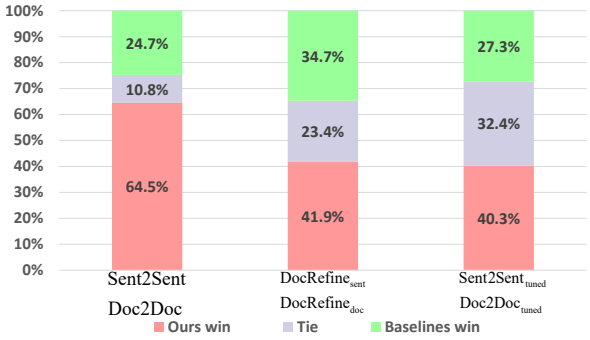

Figure 6: Comparison of our approach with the reranking variant.

E Translation Refinement Prompts

Table 10 presents the prompt we use for baselines, including SentRefine$_{Sent}$, DocRefine$_{Sent}$ and DocRefine$_{Doc}$. Note that we use the same prompt when we conduct DocRefine$_{Sent}$ and DocRefine$_{Doc}$.

F Experimental Results in s-COMET and d-BLEU

Table 11 shows the detailed d-BLEU scores of our main experiments. Table 12 shows the detailed s-COMET scores of our main experiments. Table 13 shows the detailed s-COMET scores of our experiments in refining GPT translations.

G Comparison of Our Approach with Reranking Variant

Since our approach uses two intermediate translations, we compare it to a reranking variant that selects the better sentence-level translation from our two baselines, ensuring a fair comparison. Specifically, we calculate the percentage of sentences, based on the reference-based COMET score, where our approach either outperforms, underperforms, or ties[13] with the reranking variant.

Figure 6 presents the comparison results for De ↔ En translation. It demonstrates that our approach outperforms the reranking variant by winning more sentences, even when the latter reranks several different two baselines.

[13]If the difference in their COMET scores is 0.1 or smaller, the two translations are considered a tie.

H Prompt for Analysing Translation Errors

We present the prompt used for analysing translation errors in Table 14. "Mistranslation", "Overtranslation", "Undertranslation", "Addition" and "Omission" are sentence-level translation error types, while "Cohesion", "Coherence", "Inconsistent style" and "Multiple terms in translation" are document-level translation error types.

Task	Prompt Template
$\text{SentRefine}_{\text{sent}}$	You are an expert in editing translations. Given a <*src_lang*> source sentence and a <*tgt_lang*> translated version, please produce an improved translated version. Don't give any explanations. <*src_lang*> Source:<*sent_src*> <*tgt_lang*> Translation:<*hyp*> <*tgt_lang*> Translation Refinement:
$\text{DocRefine}_{\text{sent}}$ $\text{DocRefine}_{\text{doc}}$	You are an expert in editing translations. Given a <*src_lang*> source document text and a <*tgt_lang*> translated version, please produce an improved translated version. Don't give any explanations. Each sentence is separated by #id. <*src_lang*> Source: <*doc_src*> <tgt_lang> Translation: <*hyp*> <*tgt_lang*> Translation Refinement:

Table 10: Prompts used in our baselines.

System	*X*→En					En→*X*					Avg.
	De→	Es→	Ru→	Fr→	Zh→	→ De	→ Es	→ Ru	→ Fr	→ Zh	
LLaMA-3-8B-Instruct											
Sent2Sent	34.73	40.81	31.16	33.30	22.35	25.07	39.33	22.25	31.85	29.12	30.99
$\text{Sent2Sent}_{\text{tuned}}$	48.26	53.44	41.58	45.09	34.02	31.93	43.92	27.24	34.29	36.07	39.58
Doc2Doc	37.02	43.01	32.92	34.52	26.33	25.68	40.04	23.09	30.32	33.41	32.63
$\text{Doc2Doc}_{\text{tuned}}$	47.04	53.50	42.80	43.35	35.95	30.11	44.59	27.37	34.96	38.65	39.83
$\text{SentRefine}_{\text{sent}}$	46.11	52.54	42.20	43.58	32.88	30.22	44.84	27.38	35.05	38.07	39.29
$\text{DocRefine}_{\text{sent}}$	45.16	53.77	44.33	45.44	35.92	30.02	43.93	26.68	34.90	37.79	39.79
$\text{DocRefine}_{\text{doc}}$	46.16	53.90	44.32	45.07	36.14	29.50	44.65	28.34	34.73	37.65	40.05
Ours	**48.51**	**54.70**	**45.59**	**45.57**	**37.66**	**32.23**	**45.78**	**28.74**	**35.26**	**38.96**	**41.30**
- QA Fine-tuning	47.86	54.07	44.81	45.02	37.07	31.47	44.87	28.43	34.42	38.77	40.68
Mistral-Nemo-Instruct											
Sent2Sent	38.18	43.20	34.45	35.87	27.51	29.02	41.88	25.44	33.17	34.37	34.31
$\text{Sent2Sent}_{\text{tuned}}$	40.62	45.67	39.29	38.93	31.90	30.00	42.77	27.15	33.73	35.07	36.51
Doc2Doc	40.92	45.20	37.51	37.98	29.74	29.70	42.10	27.88	34.10	37.09	36.22
$\text{Doc2Doc}_{\text{tuned}}$	49.17	55.10	43.35	46.01	38.25	31.65	45.75	22.15	37.10	42.24	41.08
$\text{SentRefine}_{\text{sent}}$	46.11	52.54	47.90	45.25	32.65	30.22	44.84	30.40	36.05	35.10	40.11
$\text{DocRefine}_{\text{sent}}$	48.75	55.56	46.45	46.49	36.76	34.13	46.12	31.13	37.45	41.44	42.43
$\text{DocRefine}_{\text{doc}}$	49.77	55.70	46.29	46.52	37.09	33.82	46.33	31.02	37.29	42.68	42.65
Ours	**51.17**	**56.20**	**48.58**	**47.97**	**41.00**	**35.44**	**47.01**	**32.79**	**38.43**	**43.13**	**44.17**
- QA Fine-tuning	50.43	55.37	47.97	45.92	37.89	35.28	46.64	31.62	37.87	42.41	43.14

Table 11: Performance in document-level (d-BLEU) score.

System	*X*→En					En→*X*					Avg.
	De→	Es→	Ru→	Fr→	Zh→	→ De	→ Es	→ Ru	→ Fr	→ Zh	
LLaMA-3-8B-Instruct											
Sent2Sent	87.71	88.32	83.74	86.63	84.60	84.47	86.82	83.23	84.55	79.76	84.98
Sent2Sent$_{\text{tuned}}$	88.93	88.91	86.38	88.33	86.27	86.28	87.12	86.25	86.43	86.49	87.14
Doc2Doc	88.62	88.76	84.47	87.36	85.84	83.87	87.07	82.61	84.79	83.85	85.72
Doc2Doc$_{\text{tuned}}$	89.35	89.91	80.51	88.29	86.38	87.20	88.20	83.76	86.26	85.51	86.54
SentRefine$_{\text{sent}}$	89.12	89.65	**85.29**	88.08	86.53	87.10	88.17	**87.16**	86.38	86.70	87.42
DocRefine$_{\text{sent}}$	88.96	89.08	83.09	88.45	87.19	87.18	88.17	83.21	86.08	86.42	86.78
DocRefine$_{\text{doc}}$	89.22	89.51	84.45	88.24	87.25	86.86	88.34	86.12	86.39	86.70	87.31
Ours	**89.63**	**89.95**	84.58	**88.58**	**87.26**	**87.76**	**88.61**	86.34	**86.50**	**86.88**	**87.61**
- QA Fine-tuning	89.41	89.88	84.44	88.43	87.19	87.43	88.37	85.63	86.14	86.69	87.36
Mistral-Nemo-Instruct											
Sent2Sent	88.52	88.40	84.24	87.00	86.18	86.64	87.32	86.25	85.52	85.41	86.54
Sent2Sent$_{\text{tuned}}$	88.49	88.55	85.03	87.78	86.42	87.24	87.22	87.17	86.52	85.85	87.03
Doc2Doc	89.15	89.29	85.16	87.90	86.81	86.56	87.30	86.66	85.65	85.74	87.02
Doc2Doc$_{\text{tuned}}$	89.70	90.20	85.01	88.61	87.70	85.91	88.66	85.19	86.99	87.56	87.53
SentRefine$_{\text{sent}}$	89.33	89.80	85.51	88.24	86.71	88.04	88.30	87.89	86.77	86.71	87.73
DocRefine$_{\text{sent}}$	89.63	90.03	84.21	88.02	87.64	88.24	88.68	86.93	86.90	87.55	87.78
DocRefine$_{\text{doc}}$	89.74	90.06	83.50	88.21	87.49	88.21	88.69	86.33	86.85	87.44	87.65
Ours	**89.94**	**90.45**	**86.10**	88.51	**87.96**	**88.53**	**89.02**	**88.31**	**87.16**	**88.04**	**88.40**
- QA Fine-tuning	89.90	90.12	85.82	**88.65**	87.87	88.49	88.87	87.81	87.07	87.71	88.23

Table 12: Performance in sentence-level COMET (s-COMET) score.

#	System	*X*→En					En→*X*					Avg.
		De→	Es→	Ru→	Fr→	Zh→	→ De	→ Es	→ Ru	→ Fr	→ Zh	
1	GPT Sent2Sent	88.39	88.51	83.76	87.05	86.34	87.43	87.63	87.55	86.35	87.09	87.01
2	GPT Doc2Doc	88.12	89.10	85.24	87.02	86.96	87.98	88.41	87.88	86.83	87.70	87.52
3	GPT SentRefine$_{\text{sent}}$	88.51	88.56	84.42	87.63	86.60	87.72	88.28	87.16	86.72	87.44	87.30
4	GPT DocRefine$_{\text{sent}}$	88.65	88.69	84.82	87.61	86.55	88.41	88.78	88.45	87.10	87.24	87.63
5	GPT DocRefine$_{\text{doc}}$	88.64	88.90	84.83	87.70	86.65	88.38	88.71	88.47	87.16	87.42	87.69
6	GPT DocRefine$_{\text{doc+sent}}$	88.99	**89.25**	85.09	87.79	86.98	88.28	88.59	88.41	87.03	87.79	87.82
7	L-DocRefine$_{\text{doc+sent}}$	88.78	88.98	84.28	87.81	86.73	88.16	**89.11**	86.45	86.95	87.32	87.46
8	M-DocRefine$_{\text{doc+sent}}$	**90.02**	89.05	**86.29**	**87.92**	**86.99**	**88.67**	89.08	**88.57**	**87.32**	**87.95**	**88.19**

Table 13: Performance in s-COMET when refining translations from GPT-4o-mini. For the GPT-based refinement systems, we use the same prompt templates as those used in our approach, but without fine-tuning. L-* and M-* denote our fine-tuned LLaMA-3-8B-Instrcut and Mistral-Nemo-Instruct, respectively.

[Source]:
<*src_doc*>
[Reference]:
<*ref_doc*>
[Hypothesis]:
<*hyp_doc*>

[Error Types]:
- Mistranslation: Error occurring when the target content does not accurately represent the source.
- Overtranslation: Error occurring in the target content that is inappropriately more specific than the source.
- Undertranslation: Error occurring in the target content that is inappropriately less specific than the source.
- Addition: Error occurring in the target content that includes content not present in the source.
- Omission: Error where content present in the source is missing in the target.
- Cohesion: Portions of the text needed to connect it into an understandable whole (e.g., reference, substitution, ellipsis, conjunction, and lexical cohesion) missing or incorrect.
- Coherence: Text lacking a clear semantic relationship between its parts, i.e., the different parts don't hang together, don't follow the discourse conventions of the target language, or don't "make sense."
- Inconsistent style: Style that varies inconsistently throughout the text, e.g., One part of a text is written in a clear, "terse" style, while other sections are written in a more wordy style.
- Multiple terms in translation: Error where source content terminology is correct, but target content terms are not used consistently.

Considering the provided context, please identify the errors of the translation from the source to the target in the current sentence based on a subset of Multidimensional Quality Metrics (MQM) error typology.
You should pay extra attention to the error types related to the relationship between the current sentence and its context, such as "Unclear reference", "Cohesion", "Coherence", "Inconsistent style", and "Multiple terms in translation".
For each sentence in machine translation, please give the error types and brief explanation for errors.The returned format is as follows:
Sentence #id :
Error types: ...
Explanation for errors: ...

Table 14: Prompt used for analyzing translation errors.