Contextual Refinement of Translations: Large Language Models for Sentence and Document-Level Post-Editing

Sai Koneru[1], **Miriam Exel**[2], **Matthias Huck**[2], and **Jan Niehues**[1]
[1] Karlsruhe Institute of Technology
[2] SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
{sai.koneru, jan.niehues}@kit.edu
{miriam.exel, matthias.huck}@sap.com

Abstract

Large language models (LLMs) have demonstrated considerable success in various natural language processing tasks, but open-source LLMs have yet to attain state-of-the-art performance in Neural Machine Translation (NMT). Nevertheless, their significant performance in tasks demanding a broad understanding and contextual processing shows their potential for translation. To exploit these abilities, we investigate using LLMs for MT and explore recent parameter-efficient fine-tuning techniques. Surprisingly, our initial experiments found that fine-tuning with Q-LoRA for translation purposes led to performance improvements in terms of BLEU but degradation in COMET compared to in-context learning. To overcome this, we propose an alternative approach: adapting LLMs as Automatic Post-Editors (APE) rather than direct translators. Building on the ability of the LLM to handle long sequences, we also propose extending our approach to document-level translation. We show that leveraging Low-Rank-Adapter fine-tuning for APE can yield significant improvements across both sentence and document-level metrics while generalizing to out-of-domain data. Most notably, we achieve a state-of-the-art accuracy rate of 88.7% on the ContraPro test set, which assesses the model's ability to resolve pronoun ambiguities when translating from English to German. Lastly, during manual post-editing for document-level translation, the source sentences are iteratively annotated, which can be used to refine further translations in the document. Here, we demonstrate that leveraging human corrections can significantly reduce the number of edits required for subsequent translations.

1 Introduction

Large Language Models (LLMs) are currently being explored for many Natural Language Processing tasks such as Question Answering and Dialogue Applications (Touvron et al., 2023; Anil et al., 2023; Thoppilan et al., 2022; Tan et al., 2023). Moreover, they are even shown to achieve or surpass state-of-the-art performance based on traditional methods. This achievement demonstrates their ability to possess general understanding and process long inputs. Given these strengths, LLMs might also be suitable for Machine Translation (MT) as many of them are also inherently multilingual from being trained on the web.

However, LLMs for MT still remain an underexplored area of research. While there are initial works on using LLMs for MT at both sentence and document-level (Vilar et al., 2022; Hendy et al., 2023; Wang et al., 2023; Zhang et al., 2023), the performance of open-source models in the range of 13B parameters still lags behind the current state-of-the-art Neural MT (NMT) methods (Xu et al., 2023). Nonetheless, closed models such as GPT-4 are on par or better depending on the language pair (Kocmi et al., 2023). Hence, it is necessary to investigate adapting small-scale open-source models for for translation.

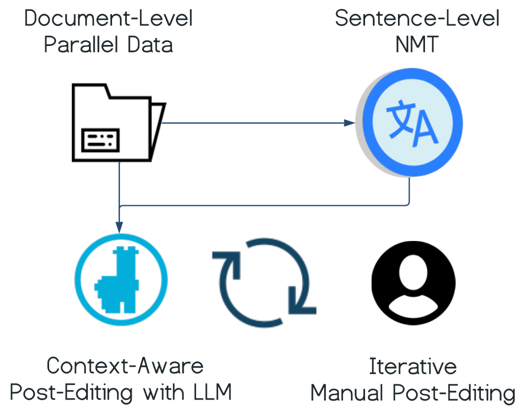

Figure 1: **Iterative Manual Post-Editing**: For manual PE, the annotator supplies the gold target context iteratively. The LLM then utilizes this gold target context for generating translations at document-level.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2711–2725
June 16-21, 2024 ©2024 Association for Computational Linguistics

Notably, these methods mainly employ In-Context-Learning (ICL) (Brown et al., 2020) as fine-tuning these models often requires a significant amount of computational resources. Hence, there might be a barrier to optimally adapting the LLMs for MT.

Parameter-efficient techniques for fine-tuning such as Low-Rank Adapters (LoRA) (Hu et al., 2021; Dettmers et al., 2023) were recently proposed to overcome large computational requirements. This enables a new adaptation process for LLMs. However, whether these techniques are sufficient for successful adaptation and better generalization is still unclear.

This work investigates exploiting LLMs for MT at both sentence and document levels. We initially experiment using ICL and parameter-efficient fine-tuning techniques such as Q-LoRA to use LLM for MT and find that adding adapters alone is insufficient and may even lead to degradation in COMET (Section 4). While full fine-tuning may resolve such discrepancies, it is important to efficiently adapt open-source LLMs with minimal computational resources. To mitigate this and still exploit the strengths of LLMs, we propose to adapt them as Automatic Post-Editors (APE) correcting NMT systems hypothesis rather than direct translators.

This approach offers several advantages. It introduces modularity, allowing state-of-the-art or customized NMT techniques to be applied independently, followed by LLM improvements. Additionally, LLMs can refine the sentence-level NMT systems output and generate consistent and coherent text using their ability to generate fluent and long documents.

The cascaded system of NMT and LLM offers modularity and can also enable the integration of human feedback. By feeding the LLM human-corrected translations from previous sentences in the document, we show that it can leverage this feedback to improve the current sentence's translation. This process can be applied iteratively in practice, sentence by sentence, as the annotator progressively corrects the translated document (see Figure 1 for Iterative Manual Post-Editing).

We summarize our main findings and contributions below:

- **Effective Combination of NMT and LLM**: Our sentence-level LLM APE demonstrates a successful fusion of knowledge from NMT systems and LLMs, leading to substantial enhancements in translation quality. Importantly, we observe that the LLM APE exhibits robustness and can adeptly correct NMT systems, even for test sets from different domains that it was not explicitly trained on.

- **Extension to Document-level Post-Editing**: We extend our approach to document-level APE and observe significant improvements in both sentence and document-level translation metrics. Notably, we achieve a state-of-the-art accuracy of 88.7% on the ContraPro English to German test set, underscoring the effectiveness of APE.

- **Iterative Manual Post-Editing**: We introduce a promising use-case scenario for iterative post-editing (as depicted in Figure 1). We show that providing gold target context significantly enhances the remaining translation quality, both at the sentence and document levels, as indicated by various metrics.

2 Approach: Adapting LLM for APE

Open-source LLMs may not be as proficient translators as state-of-the-art NMT systems due to no explicit training with large amounts of parallel data. However, LLMs being trained on the web containing data from several domains possess general knowledge that is lacking in an NMT model. Moreover, they are capable of processing significantly longer inputs compared to a standard sentence-level NMT. Given their strengths of knowledge and ability to process long sequences, we propose to use them for APE at both sentence and document levels. Hence, we first generate translations with NMT and then perform APE with LLMs. Our approach combines the translation capacity of NMT with fluency and understanding of LLMs.

We use a technique similar to Niehues et al. (2016), which combines Phrase-Based and NMT models. We extend this approach to incorporate LLMs for APE at both sentence and document levels. We first explain our system's complete pipeline at both text representation levels of sentence and document. Then, we describe how we train and create the data for each step in our setup. Finally, we explain how human feedback can be easily integrated into our cascaded approach during manual PE.

2.1 Pipeline

Given a source sentence s, we use an NMT model to generate an initial translation h_{NMT}. Then for our APE model, we do not provide the h_{NMT} alone as it cannot distinguish when the hypothesis from NMT is severely mistranslated but still fluent. Hence, we feed the source sentence and the initial translation to LLM and generate a refined hypothesis h_{LLM}.

$$h_{NMT} = \mathcal{G}(\theta_{NMT}, s) \tag{1}$$
$$h_{LLM} = \mathcal{G}(\theta_{LLM}, s, h_{NMT}) \tag{2}$$

where θ_{NMT} and θ_{LLM} are models trained for translation and APE. $\mathcal{G}(\theta, s, h)$ indicates generating a hypothesis using the model θ given s and h using beam search.

For APE at the document level, we extend the above formulation to process a sequence of sentences. Consider a document $\mathcal{D}$ with n source sentences s^i where i ranges from 1 to n. We first use the NMT model to generate each translation in isolation at the sentence level. Let them be denoted as h^i_{NMT}. Then, we perform APE using the sequence of source and hypothesis sentences, exploiting the LLM's ability to process and use contexts. We denote the generated document translation as h^D_{LLM}:

$$h^i_{NMT} = \mathcal{G}(\theta_{NMT}, s^i)\ \forall i \in 1..n \tag{3}$$
$$h^D_{LLM} = \mathcal{G}(\theta_{LLM}, s^D, h^D_{NMT}) \tag{4}$$

where s^D and h^D_{NMT} are the source and sentence-level hypothesis sentences joined by a separator token to form a document.

2.2 LLM Fine-tuning for APE

In our cascaded approach, we have an NMT and an LLM. For training our NMT model, we train it on available parallel data in a conventional fashion. We do not take any additional steps as our main motivation was to exploit LLMs for further enhancements. In the case of the LLM, we propose to go beyond ICL approaches and fine-tune them for maximum utility, as described in the following.

2.2.1 Training on MT Errors

To further optimize the LLM for the task of APE, we propose to fine-tune them using Q-LoRA (Hu et al., 2021; Dettmers et al., 2023). It is ideal to fine-tune the LLM by providing the source and hypothesis as input and predicting the corresponding post-edited reference. This needs data in the form of triples comprising the source, initial hypothesis, and reference. To simulate real test conditions, we need the initial hypothesis to consist of the errors generated from the NMT model we plan to use.

To achieve this, we follow these steps:

1. We partition the training data into two halves.

2. We train two separate NMT models, one for each half of the data.

3. We utilize the model trained on the first half to perform inference on the second half, and vice versa.

This process yields the same quantity of instances as the original training set, with initial hypotheses that exhibit errors typical of the NMT system we plan to improve on. Subsequently, we format this data into a prompt template, as described in Appendix A.4 or A.5, depending on the level of representation. Then, we employ Q-LoRA for fine-tuning[1] our LLM. For document-level APE (Doc APE), we simply split into non-overlapping chunks according to a chunk limit of source tokens and create our training data.

2.2.2 Inference

In our setup, the level of granularity can be a sentence or a document. The decoding process is straightforward for sentence-level APE (Sent APE). We generate an initial translation and feed it to the adapted LLM for our final refined hypothesis.

In the context of document translation, decoding poses a more intricate challenge compared to sentence-level. Decisions must be made regarding the direction of context for each source sentence, whether it should be drawn from the left, right, or both sides. In our work, we explore the following strategies:

Chunk-Based: This is a straightforward approach where we employ the same method used to create our training data. We create non-overlapping chunks and translate them individually. In this setup, it's possible that some sentences may lack left or right context. Note that if the number of sentences in the hypothesis doesn't match the source, we replace them with the sentence-level $\triangle$ LM outputs exploiting the modularity of the cascaded approach. We've observed that this situation occurs

[1]Training details can be found in Appendix C.1

infrequently, with at most 30 sentences, and thus for rare instances.

Batched Sliding Window: We translate the document using a sliding window approach with a payload, as described in Post and Junczys-Dowmunt (2023). Following our chunk limit, we append the sentence we intend to translate with as much preceding source context as possible. Then, we translate the entire chunk, including the context (*Payload*), and extract the last sentence using the separator token.

Continuous Sliding Window: Similar to the previous approach above, we append the left source context according to the chunk limit for translation. However, the key distinction here lies in not regenerating the target context at every step. When translating a sentence, we force-decode the previous sentence translation that is the target context in the next step. Hence, at each step, only one sentence is translated into the target language, which is then used for forced decoding in the subsequent step to provide the target context (Referred to as *Sequential Decoding* in Herold and Ney (2023)).

2.3 Integrating Manual Feedback

Consider the case of manual PE, where the annotator edits each sentence in the document. Here, we have access to the human-corrected target context, which can be used to refine future translations.

We propose to integrate this information into our APE system. We condition its subsequent translations on this expert knowledge by iteratively feeding the model human-corrected contextual information from preceding sentences and appending it to the prompt. This modular approach enables straightforward integration of human input without requiring additional training.

3 Experimental Setup

Models: In our proposed approach, we have a sentence-level NMT system that generates an initial hypothesis and an LLM which then improves it. Nonetheless, we want a strong NMT model to assess the benefits of using LLMs. Furthermore, the model should be efficient to add more latency to the two step approach. Therefore, we fine-tune the pre-trained DeltaLM[2] (Ma et al., 2021) ($\triangle$ LM) for initializing our NMT sentence-level model (Refer to Appendix C.2 for more details). We chose this model given its size and the performance in the constrained setup in the IWSLT evaluation for multilingual track (Agarwal et al., 2023). We also ablate with NLLB (Costa-jussà et al., 2022) to compare with state-of-the-art NMT models (Section 5.3). While For LLM, we use the recently open-sourced Llama-2-13b-chat-hf (Touvron et al., 2023) as it is instruction-finetuned and has reasonable compute and memory requirements when adapting with 4bit Q-LoRA (Dettmers et al., 2023).

During training (Refer to Appendix C.1 for more details), we mask the loss on the prompt, which means that the LLM is exclusively trained to predict the reference given the source and hypothesis.

Datasets & Metrics: We primarily focus on translating talks from **English to German** at a document-level. This choice is based on the large availability of document-level parallel data, the current state of sentence-level NMT quality, and the necessity for contextual information in this translation direction.

For training our sentence NMT and post-editor LLMs, we utilize the MuST-C V3 Corpus (Di Gangi et al., 2019). This corpus aligns well with our objectives, containing parallel data annotated with talk IDs for document-level translation.

For testing, we report results on three test sets. First, we select a subset of the training corpus with the most contextual phenomena in the speeches. This choice stems from the need for context not always prevalent, and hence, standard tests may not suffice for document-level evaluation. To identify such contextual phenomena, we employ the MuDA tagger (Fernandes et al., 2023), which automatically tags words requiring contextual information in the training corpus. We select talks with the highest number of tags for each phenomenon, resulting in 14 talks in our test sets, addressing contextual occurrences related to pronouns, formality, and lexical cohesion. Then, we remove the selected talks from our training data and use them for testing.

To evaluate the robustness of our approach with out-of-domain data, we use the WMT21 News test set (Akhbardeh et al., 2021) and the ACL dev set from IWSLT23 (Salesky et al., 2023). Although the ACL data consists of talks, its content contains terminology and domains that are unlikely to be found in the training data. Furthermore, both test sets are annotated at the document level, aligning with our experimental setup.

Table 1 presents an overview of the datasets. Notably, the number of detected tags is relatively small

[2] We use the $\triangle$LM base model with $360M$ parameters

Dataset	Sentences	Documents	MuDA Tags		
			Pronouns	Formality	Lexical Cohesion
MuST-C V3 Train	261.4K	2.5K	28K	82K	86K
MuST-C V3 Test	3637	14	332	1127	1268
WMT 21 News	1002	68	42	145	381
ACL Dev	468	5	12	38	478

Table 1: Statistics of our training and test data sets. We report the number of sentences and documents along with the total tag occurrences annotated by the MuDA tagger.

compared to the total sentences, even when accounting for false positives in WMT and ACL test sets. Therefore, creating a custom test set was essential to reliably evaluate context usage and sentence-level translation quality.

We also report scores on the ContraPro (EN $\rightarrow$ DE) (Müller et al., 2018) for a targeted evaluation of context usage in resolving pronoun ambiguities.

Regarding metrics, we employ various methods to assess the quality at both the sentence and document levels. Our report includes BLEU (Papineni et al., 2002), ChrF2 (Popović, 2016) using SacreBLEU (Post, 2018), and COMET[3] (Rei et al., 2022) scores for sentence-level evaluation. To gauge the quality of word prediction using contextual information at the document level, we report Precision, Recall, and F1 scores for words detected by the MuDA tagger.

4 Automatic Post-Editing is Necessary

Shot Size	BLEU	ChrF2	COMET
Sent-level $\triangle$ LM	**30.45**	**57.0**	**0.8179**
Llama2 13B ICL (Random 2)	21.53	50.0	0.7795
Llama2 13B MT	28.92	55.9	0.7664
Llama2 13B 0-Shot APE	27.26	53.9	0.8008

Table 2: ICL, LoRA fine-tuning for MT and 0-Shot APE performance of the Llama2 13B on Must-C test set. We report BLEU, ChrF2 and COMET scores for each configuration (EN $\rightarrow$ DE) and highlight the best scores in **bold** for each metric. We report only best performing ICL configuration but provide results of 1-5 shots in the Appendix B.

While APE with LLMs seems intuitively advantageous, our first step is to empirically evaluate it against several baselines, including alternatives like In-Context-Learning (ICL) (Brown et al., 2020) and fine-tuning with LoRA. To justify the development of a cascaded system with added computational complexity, we assess the following configurations[4]:

Sentence-Level NMT: We fine-tune $\triangle$LM on the training data at sentence-level in conventional fashion.

In-Context-Learning with Llama2 13B: We prompt the model according to Vilar et al. (2023), selecting examples randomly or similar to the current prompt. To find sentences more closely related to our source, we extract sentence embeddings from our training data using Sent-BERT (Reimers and Gurevych, 2019) and retrieve the nearest neighbors for efficiency using FAISS (Johnson et al., 2019) (*Llama2 13B ICL*).

Llama2 13B + Adapters: Leveraging the recent advancements in efficient fine-tuning of LLMs, we fine-tune with adapters using LoRA (Hu et al., 2021) (Training and Hyper-Parameter Details can be found in Appendix C.1). Like the sentence-level NMT, we fine-tune it on all of our training data at the sentence level (*Llama2 13B MT*).

Zero-Shot Post-Editing with Llama2 13B: Finally, we consider the case of simple zero-shot PE to evaluate the in-built ability of the model to use the knowledge from another system and compare it to ICL where it acts as a direct translator.

Results for the above setups are reported in Table 2. First and foremost, we observe that the sentence-level $\triangle$ LM achieves the highest scores across all metrics. This underscores the highly competitive performance of a dedicated NMT model with 360M parameters compared to a 13B LLM.

Furthermore, we find that in the case of ICL, the selection strategy is relatively unimportant, with both random and FAISS performing similarly as indicated by the scores in Appendix B. Addition-

[3] Unbabel/wmt22-comet-da

[4] Prompt templates for all the configurations described in Appendix A

ally, increasing the number of exemplars in the prompt had a detrimental effect on our setup. Moreover, adapting with LoRA yields the highest BLEU and ChrF2 scores of 28.92 and 55.9 when compared to setups that rely solely on the LLM. However, it's worth noting that COMET scores decrease compared to ICL. These findings suggest that fine-tuning with Q-LoRA on extensive parallel data may lead to higher scores in lexical metrics but degradation in COMET. Whether such a trend is also observed with full fine-tuning demands further exploration, and we do not perform it due to a lack of computational resources.

Zero-Shot APE beats ICL across metrics (COMET included), unlike LoRA, showing LLMs' innate post-editing ability. However, it falls short of sentence-level △ LM. Therefore, we propose to train the adapter for APE rather than direct translators to exploit LLMs.

5 Llama2 13B as Sentence-Level Post Editors

Model	BLEU	ChrF2	COMET
MuST-C V3			
△ LM	30.45	57	0.8179
Llama2 13B MT	28.92	55.9	0.7663
Llama2 13B 0-Shot APE	27.26	53.9	0.8009
△ LM + Llama2 13B Sent APE	**31.71**	**58.3**	**0.8330**
WMT 21 News			
△ LM	21.53	52.6	0.7911
Llama2 13B MT	23.61	54.3	0.7931
Llama2 13B 0-Shot APE	21.44	52.0	0.7982
△ LM + Llama2 13B Sent APE	**25.16**	**56**	**0.8411**
ACL Dev			
△ LM	31.36	60.5	0.7945
Llama2 13B MT	31.47	60.5	0.772
Llama2 13B 0-Shot APE	30.83	60.3	0.8028
△ LM + Llama2 13B Sent APE	**36**	**63.9**	**0.8321**

Table 3: Performance of Sent-Level Llama2 13B APE on test sets in and out of the domain. △ *LM + Llama2 13B Sent APE* denotes using the hypothesis of △ LM as input for our adapted Sentence-Level Llama2 13B post editor. We report BLEU, ChrF2, and COMET scores for each approach (EN → DE) and highlight the best scores in **bold** for each metric.

In this section, we evaluate the performance of improving sentence translations with APE. First, we discuss the results of improving translations generated only by △ LM on in-domain test data. Then, we analyze the influence of moving away from our training conditions to assess the robustness of the model. We achieve this by first evaluating its performance on out-of-domain test sets and combining it with hypotheses generated by models other than △LM.

5.1 Improved Translation Quality with Sentence-Level APE

We evaluate our Sentence-Level Llama2 APE and present the results in Table 3. To assess the utility of APE, we also report scores for the individual models, namely, the sentence-level △ LM and the Llama2 fine-tuned with LoRA on the parallel data.

We see that post-editing the output of △ LM with Llama2 outperforms other models across all metrics while fine-tuning for MT alone shows degradation. We hypothesize that this is primarily due to LLMs' internal knowledge and intrinsic ability to generate fluent sentences while lacking in translation capability. However, by providing initial translations to make the task easier, LLM improves the quality by a high margin.

5.2 Generalizability to Out-Of-Domain Data

From Table 3, we observe that the performance gains are more pronounced on the WMT21 News and ACL test sets compared to our MuST-C test set. The primary difference is that WMT and ACL fall outside the domain of the training data. Hence, we observe more significant improvements compared to our MuST-C test set. We gain by 1.26 BLEU on MuST-C but up to 5.64 and 3.63 BLEU on the out-of-domain ACL and WMT test sets respectively.

This scenario mirrors practical situations where a system encounters out-of-domain sentences and performs sub-optimally. By utilizing Llama2, containing a broader spectrum of "knowledge" (Illustrated in Table 10), we demonstrate that it can significantly enhance translation quality.

5.3 Generalizability to NMT Models

Apart from improving △ LM hypothesis, it is ideal if the APE with Llama2 can enhance translations of various NMT models. Therefore, to critically assess the generalization ability of the Sentence-Level Llama2 APE, we evaluate it on correcting hypotheses that were not generated from △ LM on MustC and out-of-domain ACL dev set. For this purpose, we utilize the NLLB models[5] (Costa-jussà et al., 2022) and present the results in Table 4.

[5]We perform inference with 8-bit quantization and achieve slightly lower scores than reported in the literature

Model	BLEU	ChrF2	COMET
MuST-C V3			
Llama2 13B MT	28.92	55.9	0.7663
NLLB 3.3B	31.6	58.8	0.8265
NLLB 3.3B + Llama2 13B Sent APE	**32.44***	**58.9***	**0.8320***
ACL Dev			
Llama2 13B MT	31.47	60.5	0.772
NLLB 3.3B	43.01	69.7	0.8321
NLLB 3.3B + Llama2 13B Sent APE	40.09	67.2	0.8372*
NLLB 54B	**45.82**	**71.56**	**0.844**
NLLB 54B + Llama2 13B Sent APE	40.91	67.8	0.8407

Table 4: Analyzing the robustness of the Llama2 13B Sentence-Level APE. *NLLB X LLM + LLM Sent APE* denotes using the hypothesis of NLLB X as input for our adapted Sentence-Level LLM APE. Best scores for a test set are in **bold** for each approach. If the post-editor improves the hypothesis according to a metric, we denote it with *

We show that for the in-domain test set MuST-C, the APE sucessfully improves the scores in all metrics with notably an increase in COMET score of 0.65. In the case of out-of-domain ACL, APE improves the COMET score for the 3.3B NLLB model (0.5 gain) but hurts lexical metrics, suggesting it rephrases outputs while maintaining quality. However, APE harms 54B NLLB translations, likely due to difficulty finding errors in such strong models and adapter training focused on lower-quality hypotheses.

6 Llama2 as Document-Level Post Editors

Another motivation for our approach is to exploit LLMs ability to process long sequences for Doc APE. In this section, we evaluate and analyze the performance of our Doc APE model in detail.

To gain insights on whether the Doc APE with Llama2 is beneficial, we compare it against several models. Apart from the previously mentioned sentence-level models such as $\triangle$ *LM* and *Llama2 + LoRA*, we also extend them to the document-level by concatenating sentences (Tiedemann and Scherrer, 2017) (*Doc2Doc*). Furthermore, we evaluate different decoding strategies and report both sentence and document-level metrics in Table 5.

After tuning on the dev data, we set the LLama2 maximum chunk token sizes as 1024 for training and 256 for inference (See Figure 2 for more information). This ensured at least 5 preceding sentences for most data, which we found to be reasonable given the computation requirements with large inputs. For $\triangle$ LM Doc2Doc, we use a smaller chunk size (128 tokens) due to its limited capacity.

Concatenating Sentences for Doc2Doc Proves Insufficient: Our analysis reveals that models fine-tuned with $\triangle$ LM and Llama2 separately at the document level exhibit subpar performance when compared to the sentence-level $\triangle$ LM across all considered metrics. This limitation likely stems from the scarcity of document-level parallel data, a common occurrence, particularly in the context of low-resource languages. This highlights the inadequacy of concatenation as a standalone approach in practical use cases.

Navigating the Trade-off between Sentence and Document APE: Doc APE models outperform sentence-level on BLEU/ChrF2 (despite slight COMET dip of 0.003 between Sent and Doc APE), showing promise for document translation. The Doc APE models leverage context effectively for pronouns and formality by achieving the best F1 scores of 0.75 and 0.74. However, it is still unclear why the COMET score of Doc APE model is worse while we observe improvements in all other metrics.

Impact of Decoding Strategy: Doc APE's different decoding strategies (chunking, windowing) show no clear winner in the sentence or document-level metrics. Batched sliding window, though computationally expensive, offer no significant advantage. Thus, a continuous sliding window or chunking may be preferred for efficiency. However, further research across domains and languages is crucial to comprehensively understand Doc2Doc decoding strategies.

6.1 Incorporating Target Context during Manual Post Editing

Until now, we have focused on APE and assumed there is no human feedback. However, in the case of manual PE, we force decode the previous target sentences as the manually corrected target context and condition the model to generate the translation of the current source sentence. We denote this as $\triangle$ *LM + Llama2 Doc APE Gold Target Context* in Table 5.

By feeding gold target sentences as context to Doc APE, we achieve substantial gains across metrics: +4.14 BLEU, +2.6 ChrF2, +0.268 COMET, compared to sentence-level $\triangle$ LM. This validates Doc APE's ability to leverage context and suggests the potential for reducing manual edits in PE, leading to cost savings.

Approach	BLEU	ChrF2	COMET	Pronouns			Formality			Lexical Cohesion		
				Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
△ LM	30.45	57	0.8179	0.65	0.76	0.70	0.68	0.70	0.69	0.60	0.74	0.67
△ LM Doc2Doc	30.66	57.7	0.7481	0.66	0.78	0.71	0.68	0.72	0.69	0.6	0.74	0.66
Llama2 13B MT	28.92	55.9	0.7663	0.66	0.77	0.71	0.67	0.71	0.69	0.61	0.76	0.68
Llama2 13B MT Doc2Doc	28.98	56.1	0.8221	0.67	0.75	0.71	0.68	0.74	0.71	0.61	0.70	0.65
△ LM + Llama2 13B SENT APE	31.71	58.3	0.8330*	0.66	0.77	0.71	0.67	0.71	0.69	0.61	0.76	0.68*
△ LM + Llama2 13B Doc APE Chunk	31.47	58.4	0.8306	0.68	0.82	0.74	0.66	0.76	0.71	0.60	0.76	0.67
△ LM + Llama2 13B Doc APE Batched SW	31.77	58.9*	0.8300	0.68	0.83*	0.75*	0.67	0.77	0.72	0.61	0.77*	0.68*
△ LM + Llama2 13B Doc APE Continuous SW	31.85*	58.9*	0.8298	0.69*	0.72	0.71	0.68*	0.81*	0.74*	0.62*	0.64	0.63
△ LM + Llama2 13B Doc APE Gold Target Context	**34.59**	**59.6**	**0.8347**	**0.73**	**0.8**	**0.76**	**0.77**	**0.78**	**0.77**	**0.69**	**0.77**	**0.73**

Table 5: Comparing our Document Level APE with Llama2 13B with sentence level APE and conventional approaches. We use chunk-based decoding unless it is explicitly mentioned for Doc2Doc models. We report BLEU, ChrF2 and COMET scores for sentence level evaluation and MuDA tagger scores for document level. The best score in each metric is highlighted in **bold**. We also compare APE models without gold target context in isolation and append * for the best score in each metric.

6.2 Disambiguating Pronouns with Doc APE

We also report scores on the ContraPro test set (Müller et al., 2018). This is a benchmark designed to assess the disambiguation of pronouns, specifically "Er" (masculine)," "Sie," (feminine) and "Es" (neutral) when translating "It" from English to German. We evaluate on two setups following Post and Junczys-Dowmunt (2023). For contrastive, we force-decode the target context and determine which pronoun is most likely based on the log-likelihood. In the case of generative, we directly translate the full source paragraph and extract the last sentence to check if it contains the correct pronoun.

	Cxt Size	Contra/Gen (%)
Post and Junczys-Dowmunt (2023)	10	77.9/**70.5**
Lupo et al. (2023)	4	82.54/_
△ LM + Llama2 Doc APE	2	87.7/68.0
△ LM + Llama2 Doc APE	4	**88.7**/69.7

Table 6: Contrastive and Generative accuracy on the ContraPro English → German Test Set. Results for Sent APE and additional configurations in Table 9.

We find that our document-level APE model achieves state-of-the-art accuracy 88.7% in choosing the right pronoun. This can be attributed to LLMs pre-training in long texts. We are very comparable to Post and Junczys-Dowmunt (2023) for generative accuracy with fine-tuning only on TED talks. Moreover, this shows that when target context is made available, LLMs seem to better exploit them and are ideally suited for document-level.

7 Related Work

Document NMT: Conventional approaches in DocNMT rely on a straightforward concatenation technique (Tiedemann and Scherrer, 2017; Agrawal et al., 2018; Post and Junczys-Dowmunt, 2023). Several works also explored complicated adaptations to transformer architectures, such as the inclusion of additional context encoders (Jean et al., 2017; Voita et al., 2018), adjustments to positional information (Bao et al., 2021; Li et al., 2023), and the application of data augmentation strategies (Sun et al., 2022), among others. Similar to our work is Voita et al. (2019), where sentence-level translations are refined to create a coherent document but without considering the source context.

LLM for MT: LLMs are currently being explored for MT given their success in several tasks. These techniques were mainly facilitated by ICL (Brown et al., 2020) at sentence-level (Zhang et al., 2023; Vilar et al., 2022) or document-level (Hendy et al., 2023; Wang et al., 2023). Similar to our work, the other line of direction is integrating translation memories (Mu et al., 2023; Moslem et al., 2023) or correcting NMT system outputs in the prompt (Raunak et al., 2023; Chen et al., 2023). It is worth noting that our work sets itself apart from these approaches by leveraging efficient LoRA and enabling the effective fusion of NMT with LLMs at both the sentence and document-levels.

Online Learning for NMT: Integrating human feedback for MT was explored in both statistical and neural MT (Formiga et al., 2015; Logacheva, 2017). Many methods perform additional training steps using the feedback and alter the MT model at run-time (Turchi et al., 2017; Kothur et al., 2018). Few works explored integrating retrieval and cache mechanisms to avoid further fine-tuning (Gu et al., 2018; Shang et al., 2021; Wang et al., 2022). Our approach incorporates human feedback as context and does not need any changes.

8 Conclusion

Our work highlights LLMs' potential for APE, significantly boosting NMT at both sentence and document levels. We showed that it enables modularity, deeper text understanding, and document-level quality boosted by LLMs' massive pretraining.

For future work, we consider several research avenues. These include training the adapters on substantially larger volumes of document-level parallel data, assessing various open-source LLMs, and conducting similar experiments with low-resource languages and domain.

9 Limitations

The main disadvantage of the proposed cascaded system is the latency of generating a translation. From Table 5, we find the $\triangle$ LM performance is worse but comparable to the APE approaches with LLM. However, $\triangle$ LM can produce translations with significantly shorter latency than LLMs. Therefore, integrating techniques from quality estimation to decide when to perform APE may be helpful to overcome this limitation.

The other drawback of the cascaded approach is that it does not simulate a deep fusion. The LLM can make mistakes even when the NMT is highly confident and correct for a given translation. However, fusing them is not trivial due to the models having different vocabularies.

Finally, we would also like to mention that we performed experiments for only English to German direction, which was highly present during LLMs pretraining. The benefits of APE should also be validated for low-resource languages for generalizability, where the monolingual data of such languages may be significantly less in the LLM pretraining.

References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, William Chen, Khalid Choukri, Alexandra Chronopoulou, Anna Currey, Thierry Declerck, Qianqian Dong, Kevin Duh, Yannick Estève, Marcello Federico, Souhir Gahbiche, Barry Haddow, Benjamin Hsu, Phu Mon Htut, Hirofumi Inaguma, Dávid Javorský, John Judge, Yasumasa Kano, Tom Ko, Rishu Kumar, Pengwei Li, Xutai Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Ha Nguyen, Jan Niehues, Xing Niu, Atul Kr. Ojha, John E. Ortega, Proyag Pal, Juan Pino, Lonneke van der Plas, Peter Polák, Elijah Rippeth, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Yun Tang, Brian Thompson, Kevin Tran, Marco Turchi, Alex Waibel, Mingxuan Wang, Shinji Watanabe, and Rodolfo Zevallos. 2023. FINDINGS OF THE IWSLT 2023 EVALUATION CAMPAIGN. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 1–61, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Ruchit Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 31–40, Alicante, Spain.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R Costa-jussà, Cristina España-Bonet, Angela Fan, Christian Federmann, et al. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3442–3455, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2023. Iterative translation refinement with large language models. *arXiv preprint arXiv:2306.03856*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Patrick Fernandes, Kayo Yin, Emmy Liu, André Martins, and Graham Neubig. 2023. When does translation require context? a data-driven, multilingual exploration. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 606–626, Toronto, Canada. Association for Computational Linguistics.

Lluís Formiga, Alberto Barrón-Cedeno, Lluis Marquez, Carlos A Henriquez, and José B Mariño. 2015. Leveraging online user feedback to improve statistical machine translation. *Journal of Artificial Intelligence Research*, 54:159–192.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Christian Herold and Hermann Ney. 2023. Improving long context document-level machine translation. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 112–125, Toronto, Canada. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2106.09685*.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. 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.

Sachith Sri Ram Kothur, Rebecca Knowles, and Philipp Koehn. 2018. Document-level adaptation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 64–73, Melbourne, Australia. Association for Computational Linguistics.

Yachao Li, Junhui Li, Jing Jiang, Shimin Tao, Hao Yang, and Min Zhang. 2023. P-transformer: Towards better document-to-document neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Varvara Logacheva. 2017. *Human Feedback in Statistical Machine Translation*. Ph.D. thesis, University of Sheffield.

Lorenzo Lupo, Marco Dinarelli, and Laurent Besacier. 2023. Encoding sentence position in context-aware neural machine translation with concatenation. In *The Fourth Workshop on Insights from Negative Results in NLP*, pages 33–44, Dubrovnik, Croatia. Association for Computational Linguistics.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *arXiv preprint arXiv:2106.13736*.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland. European Association for Machine Translation.

Yongyu Mu, Abudurexiti Reheman, Zhiquan Cao, Yuchun Fan, Bei Li, Yinqiao Li, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2023. Augmenting large language model translators via translation memories. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10287–10299, Toronto, Canada. Association for Computational Linguistics.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third*

Conference on Machine Translation: Research Papers, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2016. chrf deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504.

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, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and Marcin Junczys-Dowmunt. 2023. Escaping the sentence-level paradigm in machine translation. *arXiv preprint arXiv:2304.12959*.

Vikas Raunak, Amr Sharaf, Hany Hassan Awadallah, and Arul Menezes. 2023. Leveraging gpt-4 for automatic translation post-editing. *arXiv preprint arXiv:2305.14878*.

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 (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 62–78, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Wei Shang, Chong Feng, Tianfu Zhang, and Da Xu. 2021. Guiding neural machine translation with retrieved translation template. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3537–3548, Dublin, Ireland. Association for Computational Linguistics.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

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

Marco Turchi, Matteo Negri, M Farajian, and Marcello Federico. 2017. Continuous learning from human post-edits for neural machine translation. *The Prague*

Bulletin of Mathematical Linguistics, 108(1):233–244.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Dongqi Wang, Haoran Wei, Zhirui Zhang, Shujian Huang, Jun Xie, and Jiajun Chen. 2022. Non-parametric online learning from human feedback for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11431–11439.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. *arXiv preprint arXiv:2304.02210*.

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

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. *arXiv preprint arXiv:2301.07069*.

A Prompt Templates

In this Section, we provide our prompt templates for our experiments

A.1 Prompt: LLM In-Context-Learning

Below is the prompt template for our few-shot ICL experiments. In this example, we perform 2-shot. Given, the two previous examples are either selected randomly or nearest in the embedding space for the source sentences.

```
### INSTRUCTION:
Translate the input from
English to German.

###Input: [SRC1]
####Response: [TGT1]

###Input: [SRC2]
####Response: [TGT2]

###Input: [SRC3]
####Response:
```

A.2 Prompt: LLM Adapter

We use the following template when adapting the LLM for sentence-level translation. Note that it is different from the ICL in Section A.1. However, we experimented with the below prompt for ICL and found similar results. We do not again perform the experiments for all configurations due to the computational load.

```
[INST] <<SYS>>\nYou are a
professional translator
from English to German.

The output should only be the
translation in one line.<</SYS>>

 English: [SRC]
 [/INST]
 German:
```

A.3 Prompt: LLM ZeroShot PE

Since this scenario is zero-shot, we provide more instructions so that is much easier for the model to understand the task. Since we found it was generating explanations and notes even when explicitly mentioned not to, we ask to always end the answer with "###". Later, we use it as a separator for parsing the output. We provide the prompt template below

```
[INST] <<SYS>>You are a post-editor.
You improve translations from English
to German using the English source and
German translation. Do not provide any
explanation or correction.
The translation should end with
### in new line
<</SYS>>
English: [SRC]
German Translation: [HYP]
[/INST]
Post-Edited Translation:
```

A.4 Prompt: LLM LoRA Sentence APE

In this case, we decrease the prompt size as now we perform fine-tuning and instructions are not necessary. Furthermore, it will lead to less memory consumption as the sequences are much shorter.

```
English: [SRC]
German Translation: [HYP]
Post-Edited Translation: [REF]
```

A.5 Prompt: LLM LoRA Document APE

We format the prompt similarly to the sentence level. The only difference is that now we have sentences separated by "<SS>" token in the document.

```
English: [SRC1] <SS> [SRC2] <SS> [SRC3]
German Translation: [HYP1] <SS> [HYP2]
                    <SS> [HYP3]
Post-Edited Translation: [REF1] <SS>
                    [REF2] <SS> [REF3]
```

B LLM In-Context-Learning: Zero to 5-shot

Please refer to Table 7 for results on ICL with random selection and Table 8. The results reported are on the Must-C v3 test set using LLama2.

C Training Details

C.1 Llama2 Experiments

We use the transformers library (Wolf et al., 2020) for training and inference with Llama2. While training the adapters, we set the hyper-parameters to rank 8, alpha 32, dropout 0.1, and bias as *'LoRA_only'*. Following Dettmers et al. (2023) to make the model robust to LoRA hyper-parameters, we adapt on all layers. The modules we add to the adapter include *q_proj,k_proj,v_proj,gate_proj,up_proj and down_proj*. We set a batch size for each device to 32 initially and enable *auto_find_batch_size* to *True* on 4 NVIDIA RTX A6000 GPU's. To simulate a larger batch size, we set *gradient_accumulation_steps* to 20. We use a *learning_rate* of $2e-5$. The other parameters are set to default. We train for 3 epochs and select the model with the best validation loss. During inference, we use beam search with a *num_beams* set as 3 as we find it to be reasonable given the computation and performance.

Shot Size	BLEU	ChrF	COMET
Sent-level △ LM	30.45	57.0	0.8179
0	20.47	48.3	0.7592
1	20.73	48.8	0.7697
2	21.53	50.0	0.7795
3	20.34	50.1	0.7685
4	19.87	50.0	0.7609
5	20.33	50.5	0.7658

Table 7: In-Context-Learning with Llama2 13B using Random Selection Strategy

Shot Size	BLEU	ChrF	COMET
Sent-level △ LM	30.45	57.0	0.8179
0	20.47	48.3	0.7592
1	21.16	49.8	0.7755
2	21.13	50.2	0.7724
3	19.61	49.8	0.7593
4	18.82	49.9	0.7531
5	18.51	49.9	0.7402

Table 8: In-Context-Learning with Llama2 13B using FAISS Selection Strategy

C.2 DeltaLM Experiments

We use the fairseq library (Ott et al., 2019) for our experiments with △ LM. During training, we use cross-entropy loss with label smoothing set to 0.1. We set a learning rate of 0.0001 with Adam optimizer, betas $(0.9, 0.98)$ and the initial learning rate to $1e-7$. We set both dropout and attention dropout to 0.1. We use a batch size of 2000 max tokens and perform gradient accumulation for 3 steps. We train until the validation loss increases after 5 consecutive interval steps that are set to 4500 steps (Roughly $1/3$ of epoch). During inference, we do beam size with the number of beams set to 5. The other parameters not mentioned are set to default.

D ContraPro Scores for Sentence and Document APE

	Cxt Size	Contra/Gen (%)
Post and Junczys-Dowmunt (2023)	10	77.9/70.5
Lupo et al. (2023)	4	82.54/_
△ LM + Llama2 13B Sent APE	0	60.0/_
△ LM + Llama2 13B Sent APE	2	85.8/_
△ LM + Llama2 13B Doc APE	0	50.9/_
△ LM + Llama2 13B Doc APE	2	87.7/68.0
△ LM + Llama2 13B Doc APE	4	**88.7**/69.7

Table 9: Comparing Sentence and Document APE Accuracy on the ContraPro English $\rightarrow$ German Test Set. For generative results, we only report on sentences from 1 to 10 using th evaluation script from Post and Junczys-Dowmunt (2023).

Source	This is a sentence in Spanish: Las prendas bestsellers se estampan con motivos fLoRAles, animal print o retales tipo patchwork.
Reference	Dies ist ein Satz auf Spanisch: Las prendas bestsellers se estampan con motivos fLoRAles, animal print o retales tipo patchwork.
△ LM Hypothesis	Das ist ein Satz auf Spanisch: Die Bestsellers se multidisciplinan conão fLoRAles, Tierdruck oder Reliefs ol Flitterwerk.
Post-Edited with Llama2 13B	Das ist ein Satz auf Spanisch: Las prendas bestsellers se estampan con motivos fLoRAles, animal print o retales tipo patchwork.

Table 10: Example from the ACL dev set taken from Talk id: 268 and Sentence 26. The △ LM translates everything into German including the Spanish phrase that needs to be retained in the original language. However, after APE, Llama2 13B does not translate the Spanish Phrase as it was also not translated in the source sentence.

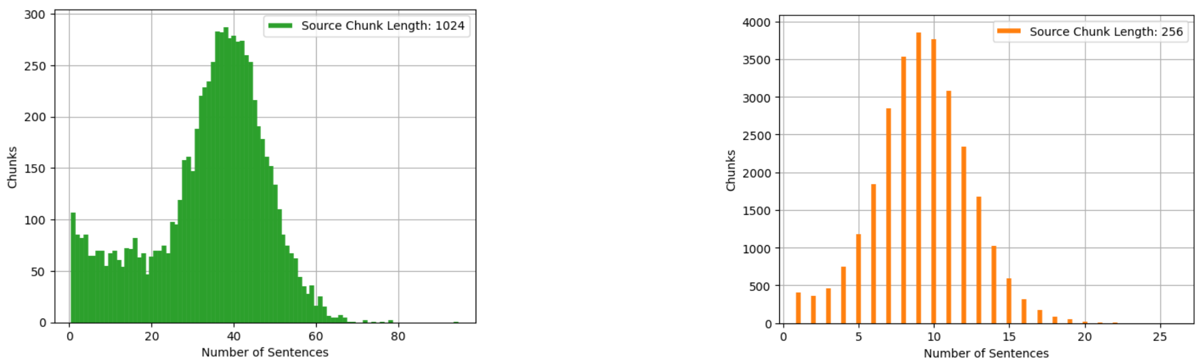

Figure 2: Number of sentences in a document with chunk sizes 1024 and 256.