LLMs in Post-Translation Workflows: Comparing Performance in Post-Editing and Error Analysis

Celia Soler Uguet Fred Bane Mahmoud Aymo João Torres
Anna Zaretskaya Tània Blanch Miró
TransPerfect
Passeig de Gràcia 11, 5B
08007 Barcelona, Spain
{csuguet,fbane,mahmoud.aymo,joao.torres,azaretskaya,tblanch}
@transperfect.com

Abstract

This study conducts a comprehensive comparison of three leading LLMs—GPT-4, Claude 3, and Gemini—in two translation-related tasks: automatic post-editing and MQM error annotation, across four languages. Utilizing the pharmaceutical EMEA corpus to maintain domain specificity and minimize data contamination, the research examines the models' performance in these two tasks. Our findings reveal the nuanced capabilities of LLMs in handling MTPE and MQM tasks, hinting at the potential of these models in streamlining and optimizing translation workflows. Future directions include fine-tuning LLMs for task-specific improvements and exploring the integration of style guides for enhanced translation quality.

1 Introduction

Large language models (LLMs) have been at the forefront of many recent advancements in natural language processing. These models show impressive capabilities in a range of tasks, including tasks that were unseen at training time. As modern LLMs are typically multilingual, machine translation is a natural application of these models. Despite initial optimism, so far research has found that well-tuned encoder-decoder models trained specifically for the task tend to outperform LLMs in most content types in the task of machine translation (Kocmi et al., 2023). However, promising results have been obtained in the peripheral tasks of machine translation post-editing (Raunak et al., 2023) and machine translation quality evaluation (Kocmi and Federmann, 2023).

Post-editing of machine translation is a common step in modern localization workflows, and can be a significant expense for global organizations producing content in multiple languages. Quality evaluation can be used to obtain actionable insights into the sources of machine translation errors, and automated quality evaluation performed at translation time in the production workflow can inform decisions about whether a translation needs additional attention or can be used directly. With the great advancements in generative language models over the last year, the possibility of automating these tasks using large language models (LLMs) has received growing attention.

Thus, in this work, we set out to compare the performance of three state-of-the-art LLMs on these two tasks in four target languages: Portuguese for Brazil (PTBR), Italian (IT), German (DE), and Japanese (JA).

2 Related Research

With the advent of LLMs, several attempts have been made to apply them to different translation tasks. Many works explore prompting LLMs to perform translation and compare their performance with the encoder-decoder based systems (Kocmi et al., 2023; Hendy et al., 2023; Gao et al., 2023; Lu et al., 2024; Vilar et al., 2023; Garcia et al., 2023). Moslem (2023) proposed an adaptive translation workflow using LLMs. Other scenarios include a human-in-the-loop pipeline to guide an LLM to produce customized output (Yang et al., 2023) or an AI-mediated post-editing process (Cady et al., 2023).

There is evidence that LLMs can be successfully applied for MT quality evaluation. In the first attempt to use a GPT model for this purpose, Kocmi and Federmann (2023) demonstrated their potential as zero-shot evaluators. Fernandes et al. (2023) then took it one step further and experimented with the AutoMQM methodology: prompting an LLM to produce MQM-style annotations of MT errors. This study is motivated by the recent finding that the evaluation methodologies that are based on MQM annotations (Lommel et al., 2014b) demonstrate higher correlation with human judgments (Freitag et al., 2021a).

Automatic post-editing (APE) of MT is another area where the utilization of LLMs has been considered. APE consists in using automated techniques to improve the quality of black-box machine translation systems. It has been a popular research topic in the MT community since the times of statistical MT systems (Simard et al., 2007; Bechara et al., 2011). With the evolution of deep learning, neural models have been increasingly applied to APE tasks (Junczys-Dowmunt and Grundkiewicz, 2016; Pal et al., 2017; Tebbifakhr et al., 2018; Correia and Martins, 2019). A detailed overview on the history of APE can be found in the excellent review article by do Carmo et al. (2021).

To our knowledge, two experiments were published so far on utilizing LLMs for this task. Vidal et al. (2022) used GPT-3 for the task and reported promising results while concluded that there was room for improvement in several areas. In a later publication, Raunak et al. (2023) reported significant improvements over the initial MT output as well as over the WMT baseline. However, the authors did find it challenging to control the model hallucinations.

Despite the promising results reported in these studies, the latest WMT task on APE paints a different picture (Bhattacharyya et al., 2023). Out of the three participating systems, one used GPT-3.5-turbo (the other two used non-LLM methods), however, it did not show any improvement over the baseline. This was the first time LLMs were used in this shared task.

From a wider perspective, one can notice that the WMT competitions, which serve as an industry and research reference, present a consistent picture across different translation tasks. In the translation task, LLMs are not "quite there" compared to the encoder-decoder systems (Kocmi et al., 2023). The same is demonstrated for APE: the transformer systems trained specifically for this task performed better than the more generic LLMs.

In our experiment, we carry out a varied study on LLMs for AutoMQM evaluation and for automatic post-editing by utilizing several state-of-the-art LLMs. We suggest that our methodology will help the community get a deeper understanding of how efficient LLMs are for these tasks. Our contribution, compared to previous experiments, consists in comparing a variety of LLMs and their performance on two different translation-related tasks. This way, we can get insights on LLMs' behavior: Do the models rank equally on both tasks? Are there any specific models that significantly outperform the rest?

3 Materials and Methods

Below we describe our process of data selection, the models used at different stages, and the LLM prompt development process for each task.

3.1 Data

For our work we chose to use data from a particular domain, specifically the pharmaceutical regulatory domain. We used the EMEA (European Medicines Agency) corpus available on the OPUS website (Tiedemann, 2009) so that our experiments could be more easily reproduced. All data were drawn from the English language corpus data. While the creators of GPT-4 and others have not made full details of their training data available, we must assume that well-known public data sets have a high likelihood of being included. However, this dataset does not include Japanese, so any model succeeding by brute-force memorization of the training set alone would be expected to perform more poorly in this language.

As this dataset contains a large number of duplicates and near duplicates, we first filtered the raw data to remove these redundant data in a case-, number-, punctuation-, and white-space-insensitive manner. We then selected 100 sentences at random as our test sentences. For each test sentence, we selected 3 similar sentences that would be used as the examples for in-context learning in the quality evaluation and post-edition tasks. Finding that the examples retrieved were still extremely similar to the test examples in many cases, we chose to impose a maximum similarity threshold of 90% (as determined by pair-wise cosine similarity of MiniLM embeddings).

The 100 test sentences and 300 example sentences were translated into each target language using our baseline machine translation models. Evaluators who are translators specialized on the Life Sciences domain were then asked to review the translations of the 400 sentences and provide a post-edited version resolving all errors and an error analysis in MQM format using a closed set of categories and severities (Lommel et al., 2014b). The evaluators had substantial experience in MTPE and a varied level of experience in error annotation. We worked with only one evaluator per language due limitations on the number of linguists available for the task. They worked in a proprietary data annotation tool and the sentences were presented to them in isolation (i.e. without any context) due to the the nature of the corpus. The annotations of the 300 example sentences would be provided to the models as examples (post-editions used for the MTPE task, error analyses used for the MQM task), while those pertaining to the test sentences would be used to evaluate model performance.

3.2 Models

Translations were obtained from models using the transformer base architecture. These models were trained using the Marian framework (Junczys-Dowmunt et al., 2018) and using the transformer-base architecture with guided alignment using alignment from fast align (Dyer et al., 2013). These models were trained with between ten and thirty million sentence pairs, for fifty epochs or until the early stopping criterion was met (no improvement in validation set perplexity for 6 successive validation checkpoints). The training data for each model was a large and diverse bilingual data set drawn from many domains, including the biomedical, clinical, and regulatory domains, but not including the EMEA dataset.

For the quality evaluation and post-editing tasks, we collected responses from three state-of-the-art LLMs: GPT-4 (*gpt-4-0125-preview*), Claude 3 (*anthropic.claude-3-sonnet-20240229-v1:0*), and Gemini (*gemini-pro*).

3.3 Prompt Development

Prompt templates were generated for each model and task (post-editing, MQM quality evaluation), providing an explanation of the task and 3 examples, along with a new sentence. The template prompted the model to perform either post-editing or MQM quality evaluation on the new sentence. To justify the complexity of the prompt, we also collected responses from the models without providing examples, but we do not report these results as they are strictly inferior to those obtained with examples. For the MQM task, we also prompt the model to produce a corrected translation.
An template of each prompt can be found in Appendix A. On top of that, an example of a complete prompt for all models analyzed in the paper along with all the responses for each model can be found on our GitHub repository.

4 Evaluation

4.1 Baseline Machine Translation

While machine translation is not within the scope of this research, the performance of the MT systems sets the context for the LLM tasks. We provide baseline quality metrics for the translation, a breakdown of error type distribution, and other details in Section 5.

4.2 Post-editing

The task of post-editing involves not just correcting errors, but also accepting correct translations. Most segments in our test set did not require post-edition, so we evaluate models both on their ability to recognize and correct errors, as well as recognize and maintain correct translations.

The quality of post-edition was judged using Word Error Rate (WER), sacreBLEU (Post, 2018), and COMET (Rei et al., 2022), each with respect to the post-edited sentence provided by the linguist. WER calculates the percentage of insertions, deletions, and substitutions needed to transform one sequence into another, while BLEU is a string-based metric commonly used to evaluate the quality of machine-generated translations by comparing them to human reference translations. A lower WER and higher BLEU score indicate a higher degree of textual similarity. Fugashi (McCann, 2020) is used for word segmentation of Japanese text for computing these metrics. With regards to COMET, it is a neural-based metric trained with the objective of predicting human judgments of MT quality. Unlike the text-based similarity metrics, COMET measures the syntactic similarity, or similarity in an abstract meaning space. A higher COMET score indicates a higher degree of semantic similarity. For our results, wmt22-comet-da was used for reference evaluation (COMET-REF) and wmt20-comet-qe-da for reference-free evaluation (COMET-QE)

To quantify the tendency of LLM's to over-edit, or unnecessarily modify correct translations, we use Mean Absolute Difference (MAD). MAD quantifies the average disparity between the Levenshtein distance of the Human Post-Edit and the Levenshtein distance of the LLM's output, from the original MT output. A lower MAD suggests that the LLM's post-edits are closer to the human-edited versions in terms of Levenshtein distance.

While the authors are aware of the limitations of the automatic metrics, human evaluation was not in scope for this experiment. As part of future research, we do see benefit in performing human MQM-based quality evaluation as described in Freitag et al. (2021b).

4.3 MQM

MQM error analysis involves identifying errors in a translation, localizing the error in the translation by providing the indices where the error begins and ends, classifying the error into a hierarchical error ontology, and judging the severity of the error. For our tasks, linguists were asked to find all errors in the sentences. To evaluate model performance on this complex task, we rely on sentence-level error-detection (whether a sentence contained an error or not), below referred to as ErrorAcc, a sentence-level fine-grained error-type detection accuracy referred to as ErrorTypeAcc, and a ErrorSpanPrecision, a token-level fine-grained error-type detection. The error ontology is provided in Table 2 below.

- ErrorAcc: with this group of metrics we aim at testing the ability of the model at detecting an error in a sentence without paying attention to the actual type of error. Accuracy measures the number of hits in comparison with the total number of segments (Total number of hits/Total number of sentences). Recall measures the percentage of segments detected as containing an error from the actual segments that contained an error (True positives / (True positives + False Negatives)), while Precision measures the percentage of actual segments with errors among the ones that were predicted as containing an error (True positives / (True positives + False Positives)). A low recall would mean that the LLM is not able to find all the segments with an error, while a low precision would mean that the LLM detects many segments as having an error, when they in fact do not contain one.

- ErrorTypeAcc: Accuracy of the model when detecting the fine-grained error category of a segment (16 classes in total). This accuracy type is calculated at sentence level in only those segments that were marked as containing an error by the linguists and trying to find any of the errors detected by the linguist in the predictions of the LLM. If none of them are found in the sentence, that prediction is counted as a failure, while if the same error is found, it is counted as hit.

- ErrorSpanPrecision: Precision of the model at detecting error spans. This metric is calculated at token-level only on the segments where errors were detected by the linguist.

We also calculate Cohen's Kappa coefficient between the human annotations and the labels from the LLMs at sentence-level in order to calculate agreement between the former and each of the latter. This coefficient has a range between -1 (perfect disagreement) and 1 (perfect agreement), with 0 representing no agreement. We present these results scaled to the range of -100 to 100 for readability. In future research, we believe it will be also necessary to analyse the specific types of errors where LLMs might demonstrate false positives and false negatives, as it can give us more valuable insights.

5 Results

5.1 Machine Translation Baseline

We first explore the error distribution from the test sentences in the 4 languages as annotated by our reviewers. Errors of type *Accuracy - Mistranslation* were the most common, amounting to a total of 92 errors of this type across the whole set and followed *Style* errors (a total of 68). See Table 2 for details about the error-type distribution. In general, the baseline machine translation presents an elevated number of critical and major errors, in large part due to the specific technical nature of the domain.

When comparing the machine translation with the human post-edit, BLEU scores for each language are between 84 and 90, and reference-based COMET scores for each language were between 92 and 94 (See Table 1 below). Of the 400 test translations reviewed and corrected by our reviewers, only 197 translations were modified (approximately 49%). These results present a pretty good baseline to start from, and will allow us, not only to test the ability of LLMs at post-editing segments but also at leaving untouched those that do not need a correction.

Post-edition accuracy metrics - Baseline scores	
metric	**raw_MT**
DE	
BLEU	86.6
WER	11.87
COMET-REF	92.71
IT	
BLEU	84.6
WER	11.23
COMET-REF	93.59
PT	
BLEU	89.6
WER	9.16
COMET-REF	93.40
JA	
BLEU	86.1
WER	12.14
COMET-REF	93.51

Table 1: PE metrics of raw MT and human post-edited version

5.2 Results of the post-edition task

Table 3 presents the results of the post-edition task, categorized by metric and language.

Both Gemini and GPT demonstrated strong performance across all metrics, with Gemini outperforming in string-based automated evaluations (i.e., BLEU, MAD and WER). However, it is noteworthy that despite these promising performances, raw_MT consistently attained high scores across our chosen metrics, achieving the highest BLEU and WER scores across all languages and second best COMET-scores for German and Italian. This suggests that the introduced LLM post-editing did not significantly enhance the translation quality. A key factor contributing to this phenomenon might be the inclination of the models to excessively edit the machine-translated segment, resulting in deviations from the human post-editing.

Our experiments were conducted with the human post-edited version as the reference in metrics such as BLEU, WER, and MAD. Hence, it is plausible that while the LLM post-edit may not be inherently deficient, it may have overly altered the translation compared to the reference.

When considering the COMET QE score, which is calculated in a quality-estimation setting (i.e., without using a reference translation), we notice that COMET assigns higher scores to LLM post-edits than the raw_MT and even the human post-edit. Further research should be carried out in order to understand whether these results are proof of an actual better quality from LLM outputs than the original raw MT, or whether COMET-QE might be biased towards machine-generated content.

Moreover, across languages, the raw machine translation also achieved the highest scores with string-based metrics, and outperformed most LLM responses when using reference-based COMET with the human post-edit as the reference.This result is consistent with our findings that LLMs might tend to make more extensive edits, leading to an increased divergence from the human post-edit.

While the models demonstrate capabilities in certain aspects, such as string-based assessments, their overall impact on translation quality requires further investigation. These findings emphasise the importance of refining post-editing strategies to align more effectively with human preferences and expectations.

5.3 Results of the MQM-analysis task

We now move to analyze the results obtained in the MQM task. In Table 4, we present all the metrics that were described in Section 4.3 including error detection accuracy, precision and recall (*ErrorAcc*, *ErrorPrecision* and *ErrorRecall*), error-type categorization accuracy (*ErrorTypeAcc*) and error-span detection precision (*ErrorSpanPrecision*).

In this regard, GPT seems to be the winner outperforming Claude and Gemini in all metrics provided, although not by a large margin. However, it is fair noticing that the ErrorTypeAcc (36.68%) and ErrorSpanPrecision are still quite low even for this model meaning that GPT shows promise at detecting sentences with errors but is still lagging behind at categorizing them according to the MQM types and detecting the actual error span.

MQM metrics			
metric	Claude	GPT_mqm	Gemini_mqm
ErrorAcc	65.75	**66.75**	64.5
ErrorPrecision	60.03	**62.31**	60.80
ErrorRecall	82.09	**85.58**	82.58
ErrorTypeAcc	35.17	**36.68**	**36.68**
ErrorSpanPrecision	**22.07**	18.23	9.79

Table 4: MQM accuracy metrics.

Error distribution of test data-set				
	Critical	Major	Minor	Total
Acc-Mistranslation	32%	44%	22%	92
Style	0%	7%	92%	68
Grammar	0%	24%	75%	45
Acc-Untranslated	10%	13%	75%	29
Domain	7%	46%	46%	28
Acc-Omission	0%	46%	53%	13
Typography	0%	9%	90%	11
Source	0%	14%	85%	7
Register	0%	20%	80%	5
Inconsistency	0%	25%	75%	4
Locale Convention	0%	0%	100%	4
Termbase	0%	0%	100%	4
Spelling	0%	33%	66%	3
Acc-Addition	0%	100%	0%	2
Unintelligible	0%	50%	50%	2

Table 2: Error distribution on the test data-set (sentences for post-edition + sentences for examples)

Post-edition accuracy metrics					
metric	human	raw_MT	claude_pe	GPT_pe	gemini_pe
DE					
BLEU	N/A	**86.6***	74.42	70.65	77.55
MAD	N/A	N/A	11.30	10.90	**8.99**
WER	N/A	**11.87***	22.73	24.27	18.16
COMET-REF	N/A	92.71	90.82	91.71	91.21
COMET-QE	42.38	43.09	**43.42**	42.8	42.92
IT					
BLEU	N/A	**84.6***	72.79	79.31	65.92
MAD	N/A	N/A	7.72	7.97	**6.58**
WER	N/A	**11.23***	22.81	21.93	18.21
COMET-REF	N/A	**93.59**	92.01	93.00	92.93
COMET-QE	36.43	37.52	**38.49**	35.59	37.33
PT					
BLEU	N/A	**89.6***	79.86	83.20	75.60
MAD	N/A	N/A	7.57	9.01	**5.79**
WER	N/A	**9.16**	17.75	19.48	13.32
COMET-REF	N/A	**93.40**	92.56	92.72	93.18
COMET-QE	35.39	37.14	38.36	**38.98**	36.82
JA					
BLEU	N/A	**86.1***	70.00	71.30	76.13
MAD	N/A	N/A	11.67	14.56	**9.60**
WER	N/A	**12.14**	19.46	22.02	14.91
COMET-REF	N/A	**93.51**	92.63	92.69	93.45
COMET-QE	31.65	31.02	31.6	**33.52***	31.27

Table 3: Metrics for the PE methods and raw MT, with reference to the human post-edit. * indicates scores with a statistically significant difference from the second best score (p <0.05).

For further exploring agreement between human annotations and models predictions, we calculate Cohen's Kappa. Results in Table 5 show a similar behaviour to the aforementioned metrics: GPT_mqm shows the highest agreement with human annotators for MQM with a coefficient of 37.74. This is a compelling finding, since as, Popovic et. al (2014a) claimed in their research about inter-annotators' agreement (IAA) on error-analysis, human annotators' meta-understanding of language is variable, even when working with professional translators. In this paper the authors calculated IAA using Cohen's Kappa in several languages. Their resulting coefficients were around 30 points for all the languages they studied.

Cohen's Kappa Coefficient		
Claude_mqm	GPT_mqm	Gemini_mqm
35.76	**37.74**	30.17

Table 5: Cohen's Kappa between human error annotations and predictions from models with the MQM prompt.

Finally, when exploring the resulting post-edited segments in this MQM setting, we found out that these tend to outperform those achieved by the PE prompt in many occasions (refer to Table 4 in Appendix B for a complete description of the metrics). However, while COMET-REF scores are higher for the MQM methods, string-based metrics are still higher for raw MT. In a similar manner as in the PE task, this suggests that LLMs are over-editing correct segments.

5.4 Results on the accuracy of models at selecting segments for post-edition

In Table 10 in Appendix C we offer a description of the number of segments which, according to linguists, needed a correction and those that were indeed corrected by the models (True Positives). This only includes the segments from the test-set (400 segments, 100 per language), since as it was mentioned in section 3.1, the extra 1,200 were passed as examples to the prompts. The most striking result here is that MQM methods correct 40% less than the PE methods, thus leading to a higher recall of the latter models. Depending on the production setting, this might be a desirable outcome where human review can be limited to the segments that were modified by the model. In a setting where a balance between precision and recall is desired, GPT_pe was the best performing model with a f1-score of 70.66 points.

Table 6 highlights each LLM's effectiveness in modifying segments containing errors as well as their ability to accurately modify the identified errors within those segments. To calculate the former, we just search for how many segments have been post-edited by the LLM. To calculate the latter, which is possibly more interesting for our research, we get the error span marked by the linguist and search for an exact match in the post-edited version. If the sub-string is not found in the post-edited sentence, we assume the error was modified. As shown in Table 6, we observe that Claude tends to modify more segments than the other two LLMs, and that the percentage of errors that were modified is below the percentage of modified segments. This points out once again that, while models are rewriting many segments, they are not always correcting the actual error that was marked by the linguist.

5.5 Qualitative analysis of generated MQM and post-editions

We further carry out a small manual analysis of the outputs of the LLMs in the quest for getting a better understanding of their behaviour. We decide to select Portuguese segments for its simplicity in analysis. Examining some of the responses from the PE and the MQM prompts, we observe the following:

- Sometimes the LLM detects the error and even gets the right type. However, while the PE prompt gets the post-edit right, the resulting fixed translation from the LLM using the MQM prompt is different from the one provided by the linguist (see Table 7). In the context of Life Sciences, there is a myriad of regulatory instructions as to how certain phrases should be translated, and while the LLM produces a correct translation it does not comply with the guidelines for this kind of documents. Adding a style guide in the prompt could the LLM produce a corrected version that follows the style and wordings from the guide.

Percentage of segments and errors modified by the LLMs per severity						
	Segments modified			Errors modified		
	Critical	Major	Minor	Critical	Major	Minor
Claude_mqm	80%	66%	84%	77%	45%	63%
Claude_pe	100%	100%	88%	91%	80%	56%
GPT_mqm	75%	61%	78%	66%	35%	64%
GPT_pe	94%	93%	95%	89%	85%	67%
Gemini_mqm	57%	43%	52%	54%	25%	38%
Gemini_pe	77%	81%	85%	69%	53%	49%

Table 6: Comparison of the LLMs' performance in modifying segments and errors per severity

Source: How does Nonafact work?

MT: Como funciona o não-afeto?

Human PE: Como o Nonafact funciona?

MQM:

Category: Accuracy - Mistranslation

Description: No relation to the source

Severity: Critical

Error span: não-afeto

GPT PE (PE prompt): Como o Nonafact funciona?

GPT PE (MQM prompt): Como funciona o Nonafact?

GPT MQM:

Category: Accuracy - Mistanslation

Severity: Critical

Description: -

Error span: não-afeto

Table 7: Example 1: Comparing human PE and MQM analysis with GPT's generated output

- In other occasions, we also observe that the LLM does not detect any error but still corrects the sentence, although in different ways depending on the prompt (see Table 8). Nevertheless, once again the post-editions do not align with the one provided by the linguist and seem to be changing parts of the sentence that were correct.

All these examples prove that there is still room for improvement on the use of LLMs for the tasks of post-edition and MQM analysis, specifically on the domain that has been researched throughout this paper. Firstly, by including a style guide of the preferred output and secondly, by encouraging a better alignment of the MQM analysis and the fixed translation generated.

Source: If the control of epilepsy is not clinically significantly improved after an adequate trial, vigabatrin treatment should not be continued.

MT: Se o controle da epilepsia não melhorar significativamente após um estudo clínico adequado, o tratamento com vigabatrina não deve ser continuado.

Human PE: Se o controle da epilepsia não melhorar significativamente após um estudo clínico adequado, o tratamento com vigabatrina não deverá ser continuado.

MQM:

Category: Fluency - Grammar

Description: Parallelism

Severity: Major

Error span: deve

GPT PE (PE prompt): Se o controle da epilepsia não apresentar uma melhoria clínica significativa após uma tentativa adequada, o tratamento com vigabatrina não deve ser continuado.

GPT PE (MQM prompt): Se o controle da epilepsia não melhorar significativamente após um período de teste adequado, o tratamento com vigabatrina não deve ser continuado.

GPT MQM: No errors found

Table 8: Example 2: Comparing human PE and MQM analysis with GPT's generated output

6 Discussion

After having carried out the automatic evaluation of the results obtained on the two proposed tasks (namely, APE and MQM analysis) we can conclude the following:

- With regards to APE, while there is still promise in using LLMs for improving MT outputs, as the COMET-QE scores from Table 3 suggest, when taking into account the compliance with a given reference segment, LLMs do not seem to be “quite there” yet as other authors have previously pointed out (Kocmi et al., 2023). In order to find out whether LLMs did indeed improve the MT translations and that results from COMET-QE are not biased towards machine-generated outputs, further research should be carried out, for instance by obtaining pair-wise human preferences between translations.

- With regard to automatic MQM detection, while error-detection metrics present some promise, error-type categorization results are still only at 36% accuracy. While this metric is quite low, it is on par or slightly above reported inter-rater agreement in human evaluations such as that carried out by Popovic et. al (2014a).
- In addition to error analysis, our MQM prompts also asked the models to produce a fixed translation. Comparing the fixed translations obtained in this way with those from the MTPE prompt, we observed that results generally improved for all languages and metrics, suggesting that the model benefits from the additional information and chain-of-thought style prompting. Further research could be carried out as to how removing the corrected translation from the examples given to the MQM prompt would affect these results.
- When comparing the accuracy of models at selecting segments for post-edition, we saw a large difference in the number of post-edited segments using the PE prompt vs. using the MQM prompt. The former tended to correct almost twice as much as the latter.
- Although, as we have mentioned, these models do not seem to be ready for production just yet, if there is an interest in using these models in completely independent workflow to carry out MQM analysis and PE, the choice as to which model to use should be made taking into account not only the accuracy of the edited content but the precision and recall metrics at selecting which segments indeed need to be post-edited as well, in order to reduce efforts while ensuring good results.
- Finally, when considering the results broken down by language, in general, we do not see great variance across languages for any of the tasks. While the reference-based metrics for Japanese are often lower than for other languages, this is a common occurrence for this language. The commensurate performance across languages suggests that data contamination has not overly biased the results, and that the LLMs have strong priors for each of the languages we studied.

7 Future Work

Among our future work plans we intend to explore the fine-tuning of LLMs for the task of post-edition and MQM and compare the performance and costs with the approach proposed on this paper. Fine-tuning an LLM for a certain task has been proven to be a successful technique for achieving better results in certain tasks while reducing costs due to the shorter prompts that need to be sent to the model.

Another item of interest would be studying the integration of a style guide either by introducing it into the prompt or during fine-tuning. This could be useful for correction of stylistic errors for client customization.

Moreover, taking into account that our baseline models already achieve state-of-the-art performance, it would be interesting to carry out the same experiments on MT output which is objectively of poorer quality and analyze whether LLM post-edition and MQM analysis could significantly improve the translation in those cases.

Finally, human evaluation of the quality of LLM-post-edited content could be performed in order to get a better understanding of the results that were achieved with the automatic metrics presented on this paper.

References

[Bechara et al.2011] Bechara, Hanna, Yanjun Ma, and Josef van Genabith. 2011. Statistical post-editing for a statistical MT system. In *Proceedings of Machine Translation Summit XIII: Papers*, Xiamen, China, September 19-23.

[Bhattacharyya et al.2023] Bhattacharyya, Pushpak, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri, and Marco Turchi. 2023. Findings of the WMT 2023 shared task on automatic post-editing. In Koehn, Philipp, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 672–681, Singapore, December. Association for Computational Linguistics.

[Cady et al.2023] Cady, Larry, Benjamin Tsou, and John Lee. 2023. Comparing Chinese-English MT performance involving ChatGPT and MT providers and the efficacy of AI mediated post-editing. In Yamada, Masaru and Felix do Carmo, editors, *Proceedings of Machine Translation Summit XIX, Vol. 2: Users Track*, pages 205–216, Macau SAR, China, September. Asia-Pacific Association for Machine Translation.

[Correia and Martins2019] Correia, Gonçalo M. and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In Korhonen, Anna, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy, July. Association for Computational Linguistics.

[do Carmo1 et al.2021] do Carmo1, Félix, Dimitar Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2021. A review of the state-of-the-art in automatic post-editing.

[Dyer et al.2013] Dyer, Chris, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *In Proc. NAACL*.

[Fernandes et al.2023] Fernandes, Patrick, Daniel Deutsch, Mara Finkelstein, Parker Riley, André F. T. Martins, Graham Neubig, Ankush Garg, Jonathan H. Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation.

[Freitag et al.2021a] Freitag, Markus, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474, 12.

[Freitag et al.2021b] Freitag, Markus, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021b. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

[Gao et al.2023] Gao, Yuan, Ruili Wang, and Feng Hou. 2023. How to design translation prompts for chatgpt: An empirical study.

[Garcia et al.2023] Garcia, Xavier, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation.

[Hendy et al.2023] Hendy, Amr, 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.

[Junczys-Dowmunt and Grundkiewicz2016] Junczys-Dowmunt, Marcin and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In Bojar, Ondřej, Christian Buck, Rajen Chatterjee, Christian Federmann, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Pavel Pecina, Martin Popel, Philipp Koehn, Christof Monz, Matteo Negri, Matt Post, Lucia Specia, Karin Verspoor, Jörg Tiedemann, and Marco Turchi, editors, *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany, August. Association for Computational Linguistics.

[Junczys-Dowmunt et al.2018] Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July. Association for Computational Linguistics.

[Kocmi and Federmann2023] Kocmi, Tom and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In Nurminen, Mary, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz, editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland, June. European Association for Machine Translation.

[Kocmi et al.2023] Kocmi, Tom, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In Koehn, Philipp, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore, December. Association for Computational Linguistics.

[Lommel et al.2014a] Lommel, Arle, Maja Popovic, and Aljoscha Burchardt. 2014a. Assessing inter-annotator agreement for translation error annotation. In *MTE: Workshop on Automatic and Manual Metrics for Operational Translation Evaluation*, pages 31–37. Language Resources and Evaluation Conference Reykjavik.

[Lommel et al.2014b] Lommel, Arle, Hans Uszkoreit, and Aljoscha Burchard. 2014b. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. In *Tradumatica*, pages 455–463.

[Lu et al.2024] Lu, Qingyu, Baopu Qiu, Liang Ding, Kanjian Zhang, Tom Kocmi, and Dacheng Tao. 2024. Error analysis prompting enables human-like translation evaluation in large language models.

[McCann2020] McCann, Paul. 2020. fugashi, a tool for tokenizing japanese in python. *arXiv preprint arXiv:2010.06858*.

[Moslem et al.2023] Moslem, Yasmin, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models.

[Pal et al.2017] Pal, Santanu, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural automatic post-editing using prior alignment and reranking. In Lapata, Mirella, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain, April. Association for Computational Linguistics.

[Post2018] Post, Matt. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

[Raunak et al.2023] Raunak, Vikas, Amr Sharaf, Yiren Wang, Hany Hassan Awadallah, and Arul Menezes. 2023. Leveraging gpt-4 for automatic translation post-editing.

[Rei et al.2022] Rei, Ricardo, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT 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.

[Simard et al.2007] Simard, Michel, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In Sidner, Candace, Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York, April. Association for Computational Linguistics.

[Tebbifakhr et al.2018] Tebbifakhr, Amirhossein, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source transformer with combined losses for automatic post editing. In Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 846–852, Belgium, Brussels, October. Association for Computational Linguistics.

[Tiedemann2009] Tiedemann, Jörg, 2009. *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, volume V, pages 237–248.

[Vidal et al.2022] Vidal, Blanca, Albert Llorens, and Juan Alonso. 2022. In Campbell, Janice, Stephen Larocca, Jay Marciano, Konstantin Savenkov, and Alex Yanishevsky, editors, *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 84–106, Orlando, USA, September. Association for Machine Translation in the Americas.

[Vilar et al.2023] Vilar, David, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting palm for translation: Assessing strategies and performance.

[Yang et al.2023] Yang, Xinyi, Runzhe Zhan, Derek F. Wong, Junchao Wu, and Lidia S. Chao. 2023. Human-in-the-loop machine translation with large language model.

8 Appendix A - Prompt Examples

8.1 GPT-MQM

”””Hello! Could you please tell me whether there is an error in the translation below?. Please, think it carefully before giving an answer:

If there is indeed an error or more than one error, could you please categorize them according to the error categories and subcategories below?

In order to carry out a proper analysis, first think of the error category and once you have that clear, subcategorize the error using the subcategories from that error category.

<error_category>Fluency: errors related to the linguistic well-formedness of the text, including problems with grammaticality, spelling, punctuation, and mechanical correctness.<error_category>

<sub_category>Domain Terminology: although the translation might be correct, it is not suited for the type of text.</sub_category>
<sub_category>Grammar: errors related to the grammar of a language</sub_category>
<sub_category>Inconsistency: translation is not consistent with previous words</sub_category>
<sub_category>Register: not using the right register (formal, informal, neutral)</sub_category>
<sub_category>Spelling: a term was misspelt, e.g., it contained to ss instead of one or a different letter was used.¡/sub_category¿
<sub_category>Typography: typographic errors, related to punctuation or tags</sub_category>
<sub_category>Unintelligible: it is a made-up word or difficult to understand in normal language</sub_category>

<error_category>Terminology: errors arising when a term does not conform to normative domain or organizational terminology standards or when a term in the target text is not the correct, normative equivalent of the corresponding term in the source text.<error_category>
<sub_category>Domain Terminology: the error is a terminology issue deriving from the domain, the type of text: medical, tourism, daily life, law...</sub_category>

<error_category>Accuracy: errors occurring when the target text does not accurately correspond to the propositional content of the source text, introduced by distorting, omitting, or adding to the message.</error_category>
<sub_category>Addition: Addition of content.</sub_category>
<sub_category>Mistranslation: when a word has been translated differently that it should</sub_category>
<sub_category>Omission: omission of content<sub_category>
<sub_category>Untranslated: term was not translated<sub_category>

<error_category>Style: errors occurring in a text that are grammatically acceptable but are inappropriate because they deviate from organizational style guides or exhibit inappropriate language style.</error_category>

<error_category>Locale convention: errors occurring when the translation product violates locale-specific content or formatting requirements for data elements.</error_category>

<error_category>Design: Errors regarding handling xml tags.</error_category>

<error_category>Source: There is an error on the SOURCE segment</error_category>.

Here are some examples that you can use as a reference:

Translation pair: {example}

Translation pair: {example}

Translation pair: {example}

Translation pair:
{translation pair to analyze and post-edit}
Analysis:”””

8.2 Gemini-PE

””” As an expert linguist, your task is to perform post-editing (Light post-edit or Full post-edit) on machine-translated segments.
You will be working with {source language} as the source language and {target language} as the target language.
Below are three examples with human post-edits on the translations:

{example with source segment, translation, and post-edit}

{example with source segment, translation, and post-edit}

{example with source segment, translation, and post-edit}

Your task is to complete the following example by post-editing the translation, applying gender bias reduction if necessary. If no post-edit is needed, the post-edited translation should remain the same as the translation.

Example:
{example with source segment and translation}
”””

9 Appendix B - Post-edition results using the MQM prompt

Post-edition accuracy metrics (MQM prompt)					
metric	human	raw_MT	claude_mqm	GPT_mqm	gemini_mqm
DE					
BLEU	N/A	**86.6**	85.0	85.1	82.8
MAD	N/A	N/A	**6.05**	6.79	10.21
WER	N/A	**11.87**	13.30	13.56	37.40
COMET-REF	N/A	92.71	**93.24**	93.10	91.21
COMET-QE	42.38	43.09	**43.61**	40.33	42.92
IT					
BLEU	N/A	**84.6***	79.90	82.90	83.10
MAD	N/A	N/A	**4.99**	5.62	6.20
WER	N/A	**11.23**	14.20	12.91	12.91
COMET-REF	N/A	93.59	79.55	**93.95**	79.71
COMET-QE	36.43	37.52	38.14	37.58	38.15
PT					
BLEU	N/A	**89.6**	88.80	89.50	88.50
MAD	N/A	N/A	**4.46**	5.65	5.64
WER	N/A	**9.16**	10.51	9.95	11.13
COMET-REF	N/A	93.40	93.86	**93.95***	93.42
COMET-QE	35.39	37.14	37.78	36.91	**40.14***
JA					
BLEU	N/A	**86.1**	82.90	85.80	81.70
MAD	N/A	N/A	10.22	**8.62**	11.82
WER	N/A	**12.14**	14.88	12.56	26.80
COMET-REF	N/A	93.51	93.20	**93.55**	91.83
COMET-QE	31.65	31.02	33.2	29.12	31.49

Table 9: Metrics for each of the methods and raw MT, with reference to the human post-edit. * indicates scores with a statistically significant difference from the second best score (p <0.05)

10 Appendix C - Accuracy of models at choosing segments for post-edition

	Claude_mqm	Claude_pe	GPT_mqm	GPT_pe	Gemini_mqm	Gemini_pe
Needed Correction	197/400					
TP+FP	193	323	155	304	159	267
TP	131	183	111	177	86	161
percentage TP	66.50	**92.89**	56.35	89.85	43.65	81.73
percentage TN	69.46	31.03	**78.33**	37.44	64.04	47.78
percentage FN	33.50	7.11	43.65	**10.15**	56.35	18.27
precision	67.88	56.66	**71.61**	58.22	54.09	60.30
recall	66.50	**92.89**	56.35	89.85	43.65	81.73
f1-score	67.18	70.38	63.07	**70.66**	48.31	69.40

Table 10: Accuracy of models at choosing which segments to post-edit. If a segment needed a correction and was post-edited it is counted as a True Positive, while if a segment did not need a correction and was left untouched, it is counted as a True Negative