Machine Translation Agent with Dimension-Specific Feedback

Anonymous ACL submission

Abstract

Recent machine translation (MT) agents rely on large language models (LLMs) as judges, typically using coarse prompts that jointly assess all types of error. We propose a framework that decomposes the evaluation into multiple expert evaluators, each specializing in a specific error dimension with detailed criteria, examples, and external knowledge. Based on a set of dimension-specific feedbacks, we tested and analyzed translation refinement using sequential, parallel, and comprehensive strategies. Experiments with both small and large models show that combining specialized evaluators outperforms a single holistic judge by more effectively capturing fine-grained errors. Our findings highlight the benefits of decomposing complex evaluation for more effective self-refinement of LLM. Furthermore, by using smaller, open-source LLMs, our approach achieves strong performance with significantly reduced computational cost, making robust translation evaluation more accessible. This work opens new avenues for scalable, modular quality control in automated translation systems.

1 Introduction

Machine translation (MT) output often exhibits ambiguities and diverse variations, making post-editing a long-standing research interest—from traditional manual approaches (Specia et al., 2017) to recent agent-based methods (Yao et al., 2023). Most of these approaches rely on holistic evaluation guided by single-score metrics such as BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020), which offer limited interpretability and diagnostic value.

To address this, the Multidimensional Quality Metrics (MQM) framework (Arle Lommel and Uszkoreit, 2014) categorizes translation errors into fine-grained dimensions such as addition, omission, mistranslation, grammar, and spelling.

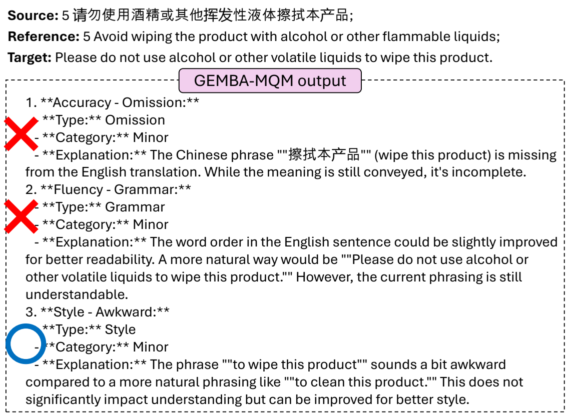


Figure 1: Example of output of Gemba MQM. This example shows that single-prompt evaluations can misclassify correct translations as errors, revealing their inherent limitations. This highlights the need for our proposed multi-evaluator approach.

While MQM provides a more transparent evaluation scheme, its reliance on expert human annotation makes it costly, subjective, and hard to scale.

Large language models (LLMs), with their impressive linguistic and reasoning capabilities, present a promising foundation for addressing these limitations. Their ability to understand nuanced language patterns makes them suitable candidates for identifying translation errors across a variety of dimensions. However, prior efforts to leverage LLMs for MQM-style evaluation have predominantly relied on single-shot, holistic prompts that ask the model to assess multiple error types simultaneously (Kocmi and Federmann, 2023). While such approaches can detect major errors, they often lack the granularity, structure, and interpretability needed for targeted and actionable feedback—particularly in cases requiring subtle revisions. As shown in Figure 1, these single-prompt evaluations sometimes misclassify correct translations as erroneous, likely due to the prompt's conflation of distinct criteria and the model's inability to focus on specific dimensions in isolation. This leads to noisy feedback and undermines trust in automated evaluation, further motivating the need for decomposed, dimension-specific assessment.

In this work, we propose a structured, multi-agent pipeline that moves beyond passive, one-off evaluation and introduces active, criterion-aware correction. Our framework is grounded in the MQM taxonomy but reimagines its application through automation. It consists of three specialized LLM-driven agents—Translate, Evaluate, and Refine—that together simulate the iterative review process typically followed by human translators and editors. Unlike prior work, we explicitly decompose the Evaluate stage into multiple sub-agents, each responsible for a specific MQM error type (e.g., Accuracy, Fluency, Terminology). This allows for dimension-specific evaluation, in which each agent produces detailed feedback including error spans, severity ratings, and explanatory rationales aligned with the corresponding MQM category.

This structured feedback is then passed to the Refine agent, which performs targeted post-editing on the initial translation. Rather than regenerating the full sentence, the Refine agent applies localized changes based on evaluator feedback, preserving valid content while improving problematic segments. This modularity enables a high degree of control and interpretability, making it possible to track which error types were addressed, how, and to what extent.

To our knowledge, this is the first work to decompose MQM evaluation into dimension-specific LLM agents and to integrate their structured outputs into an end-to-end post-editing pipeline. We evaluate our approach across multiple datasets and LLM backbones, demonstrating that our evaluator–refiner configuration consistently improves translation quality. Specifically, we observe up to a 12.6% reduction in MQM errors compared to a translate-only baseline. These results underscore the value of modular, interpretable, and scalable architectures in next-generation MT evaluation and refinement systems, paving the way for more adaptive and intelligent translation workflows.

Our contributions are summarized as follows:

- We introduce a multi-agent framework that decomposes MQM evaluation into dimension-specific LLM-based evaluators and integrates their feedback into a refinement agent, enabling fine-grained and interpretable translation correction.
- We demonstrate that dimension-specific evaluation outperforms holistic prompting in both MQM error reduction and feedback quality.
- We investigate multiple refinement strategies—sequential, parallel, and comprehensive—and provide insights into effective agent orchestration.
- Our approach improves MT performance across diverse LLMs, highlighting its robustness and scalability.

2 Related Work

Machine Translation Post-editing: Post-editing in machine translation has long been explored as a means of improving the output of automatic systems through manual or automatic revisions. Early studies emphasized human post-editing for quality assurance (Specia et al., 2017), while more recent work has attempted to automate this process using neural models trained in post-editing datasets (Junczys-Dowmunt and Grundkiewicz, 2018). However, such approaches typically require aligned triplets of source, hypothesis, and post-edited reference, which limits scalability. Our work departs from this paradigm by employing LLM-based agents that refine translations through evaluative feedback, enabling a more modular and interpretable approach to post-editing without requiring large-scale post-edited corpora.

MT Agents and Refinement Strategies: The concept of agent-based or multistep translation has gained traction with the advent of prompt-based LLM systems. Approaches such as chain-of-thought prompting (Wei et al., 2022) and agentic reasoning (Yao et al., 2023) have inspired frameworks where translation, evaluation, and refinement are treated as separate but interacting roles. Some works (Gu et al., 2024) propose multi-agent systems for natural language tasks, yet most apply generic self-review loops or feedback without domain-specific structure. In contrast, our approach defines clear agent roles, particularly focusing on MQM-guided evaluation and corresponding refinement, providing domain-aware feedback loops specifically for MT quality enhancement.

LLMs as Judges and Decomposition of Evaluation Criteria: LLMs have recently been proposed as "judges" capable of evaluating outputs in various

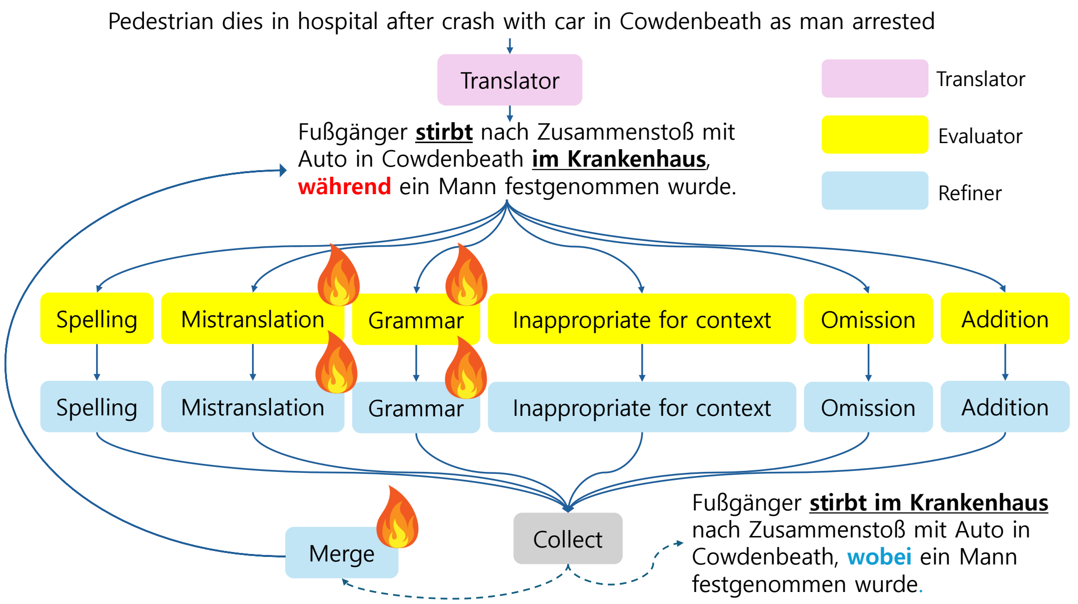


Figure 2: Overview of Multi Eval + Multi Refine (Parallel + Merge). Each evaluator-refiner pair edits the translation based on a specific MQM criterion, and their outputs are merged into a final version.

NLP tasks, including MT, with competitive performance compared to human annotators (Fu et al., 2024). Decomposing the evaluation into specific criteria such as adequacy, fluency, and coherence has been shown to improve reliability and interpretability (Zheng et al., 2024). Our work builds upon these ideas by integrating MQM-style decomposition directly into the evaluation prompts, with each prompt targeting a single dimension of translation quality. This enables both precise diagnostic insight and structured feedback suitable for downstream refinement agents.

Machine Translation Evaluation Metrics: Standard automatic metrics like BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) offer surface-level evaluations of MT output but are often criticized for poor alignment with human judgment. More recent learned metrics such as COMET (Rei et al., 2020) and BLEURT (Sellam et al., 2020) provide stronger correlations with human evaluation, yet still fall short in offering actionable feedback. MQM (Arle Lommel and Uszkoreit, 2014) was introduced as a fine-grained human evaluation framework and has seen limited success in automatic evaluation. We extend MQM by operationalizing its taxonomy through LLM prompts, enabling multi-dimensional scoring with accompanying rationales that can directly inform post-editing steps.

3 Agent Configuration

In this work, we propose a multi-agent pipeline for automating the evaluation and refinement of machine translation outputs. Our framework comprises three specialized LLM-driven agents — Translate, Evaluate, and Refine — which collectively simulate the iterative workflow typically performed by human translators and reviewers.

While prior studies have attempted to incorporate MQM into automatic evaluation pipelines, these approaches have predominantly relied on a single prompt to jointly assess multiple MQM dimensions and extract error spans (Kocmi and Federmann, 2023). Although effective for coarse-grained error detection, such methods often fall short in providing fine-grained, criterion-specific feedback necessary for targeted post-editing.

To address this limitation, we introduce a prompt decomposition strategy that formulates separate evaluation prompts for each MQM dimension (e.g., accuracy, fluency, terminology). The Evaluate agent employs these prompts to perform granular quality assessment, error severity ratings and textual rationales for each criterion. This structured feedback is then passed to the Refine agent, which applies targeted edits to the translation, thereby improving quality in a principled and interpretable manner.

The pipeline operates through a structured Translate–Evaluate–Refine cycle, enabling iterative enhancement of translation quality. We begin with a minimal agent configuration to validate the core methodology and assess its efficacy through empirical experiments. Depending on the observed performance, the framework can be further extended with more sophisticated agent interactions or adaptive prompting strategies.

In the following sections, we detail the design and operational scope of each agent, describe the construction of criterion-specific MQM prompts, and present our overall architecture for automated translation evaluation and refinement.

Our pipeline is instantiated as a fixed automaton without a planning agent. It comprises three core agent types: Translator, Evaluator, and Refiner. All agents are implemented using a frozen Qwen 2.5 7B instruct model, ensuring consistency across configurations and isolating the effect of agent interaction patterns from model scale.

3.1 Translator Agent

The Translator agent serves as the initial component in the pipeline, responsible for generating a base translation given a source input. It operates in a zero-shot setting without access to gold references, and its output serves as the input to downstream agents. While the translator is not explicitly optimized during the pipeline execution, its deterministic behavior ensures a consistent translation hypothesis, serving as a stable foundation for evaluation and refinement. This setup enables controlled analysis of quality improvements introduced solely by the downstream agents.

3.2 Evaluator Agent

The Evaluator agent is designed to perform fine-grained, criterion-specific assessments of the translation output. Rather than relying on a single holistic prompt, the evaluator utilizes prompts tailored to individual MQM categories such as Accuracy, Fluency, Style, and Terminology. For each dimension, the agent produces an error severity along with a textual explanation that highlights observed issues. These outputs are used not only for diagnostic analysis but also as direct feedback for refinement. The modularity of the evaluator enables flexible configurations (e.g., single vs. multiple evaluators) and supports both sequential and parallel evaluation workflows.

We created individual evaluators for each of the major error types based on their prevalence in the MQM error annotations from the WMT 2020 En-De human evaluation dataset (google/wmt-mqm-human-evaluation). Specifically, we selected the most frequent and impactful error categories—such as Accuracy, Fluency, and Mistranslation—as these account for the majority of errors in the dataset, ensuring our evaluators are aligned with realistic and significant translation quality issues.

3.3 Refiner Agent

The Refiner agent takes the evaluator's feedback and performs post-editing on the initial translation. Its goal is to resolve the issues identified by the evaluator while preserving the strengths of the original output. The refinement process is conditioned on both the translation and the structured feedback, allowing the agent to apply targeted edits rather than regenerate the sentence from scratch. In configurations with multiple refiners, each one may address a specific MQM dimension independently, while in single-refiner setups, the agent integrates multi-faceted feedback into a cohesive revision. This flexible structure supports a range of refinement strategies aimed at improving translation quality in an interpretable and controllable manner.

To systematically investigate the impact of agent composition and coordination strategies, we experiment with the following configurations:

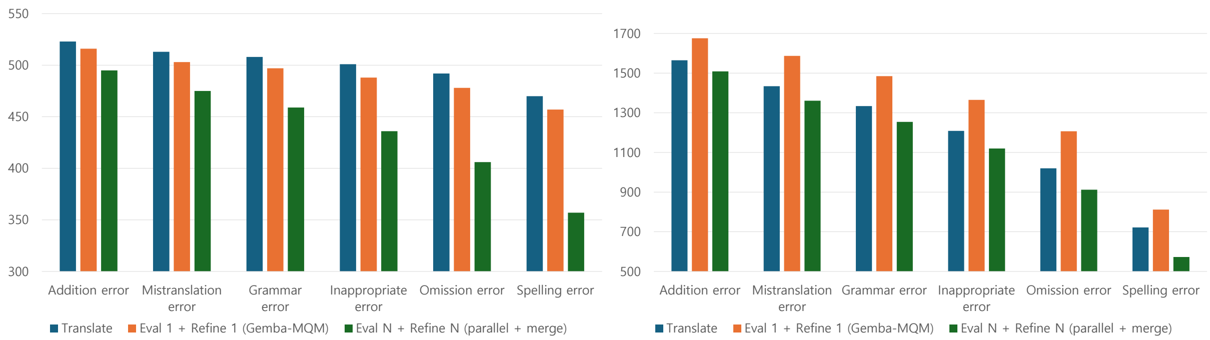
- Single Eval + Single Refine: A single evaluator assesses the initial translation using the Gemba-MQM (Kocmi and Federmann, 2023) prompt and generates feedback, which is then used by a single refiner for post-editing. This baseline reflects the limitations of prior monolithic prompting approaches, highlighting the need for more granular and modular evaluation-refinement strategies.

- Multi Eval + Single Refine: Multiple evaluators operate in parallel, each providing focused feedback on a specific MQM dimension (e.g., accuracy, fluency, terminology). These independent evaluations capture complementary aspects of translation quality. A single refiner consolidates the feedback and performs comprehensive post-editing, playing a crucial role in resolving potential conflicts and ensuring coherent integration of improvements.

- Multi Eval + Multi Refine (Sequential): Each evaluator focuses on a distinct MQM error category and operates in parallel. When multiple issues are detected, the translation is passed

Table 1: MQM-Errors Comparison on all data sets

	En-De		Zh-En	
Type	**wmttest2023**	**wmttest2024**	**wmttest2022**	**wmttest2023**
Translate	3007	5052	7557	7284
Single Eval + Single Refine(Gemba-MQM)	2984 (-0.8%)	5007 (-0.9%)	7997 (+5.8%)	8132 (+11.6%)
Multi Eval + Single Refine	**2628 (-12.6%)**	**4778 (-5.4%)**	7043 (-6.8%)	6936 (-4.8%)
Multi Eval + Multi Refine (Sequential)	2898 (-3.6%)	4801 (-5.0%)	7151 (-3.6%)	6887 (-5.5%)
Multi Eval + Multi Refine (Parallel + Merge)	**2628 (-12.6%)**	4837 (-4.3%)	**7031 (-7.0%)**	**6729 (-7.6%)**



(a) MQM-Errors on WMT2023 En–De

(b) MQM-Errors Rate on WMT2023 Zh–En

Figure 3: Our method consistently yields the lowest MQM-Errors compared to both simple translation and Gemba-MQM. Especially for Zh–En, Gemba-MQM increased errors, while our approach clearly improved translation quality.

through a sequence of refiners, each specialized in one error type. This setting allows us to analyze how individual corrections interact when applied sequentially.

- Multi Eval + Multi Refine (Parallel + Merge): Evaluators and refiners operate in parallel, each targeting a specific error type. Refined outputs are then merged to form a unified translation. This setup examines whether combining independent refinements can yield higher overall quality while preserving contributions from each dimension. (see Figure 2)

We evaluate these configurations across multiple datasets to assess the robustness and generalizability of our pipeline. The fixed-agent architecture supports controlled experimentation and enables fine-grained analysis of how evaluation specificity and refinement strategies affect translation quality.

4 Experiments

To investigate whether MQM-style translation quality assessment and refinement can be effectively automated without human annotation, we conducted experiments using large language model (LLM)-based agents. Unlike prior approaches such as Gemba-MQM, which attempt to detect multiple error types simultaneously through a single evaluation prompt, our method decomposes the evaluation task into six of the most frequent MQM error categories observed in human annotations. For each selected error type—such as mistranslation, omission, and grammar—we design a dedicated evaluator that provides fine-grained feedback, which is then used for targeted post-editing. This modular design aims to reduce the complexity of multi-error evaluation and leads to more accurate and interpretable refinements.

We validated our approach on three high-quality benchmark datasets: WMT test sets from 2022, 2023, and 2024, covering two language pairs: English–German (En–De) and Chinese–English (Zh–En). These datasets offer a diverse and challenging testbed for evaluating the effectiveness of our agent-based MQM pipeline, and allow for direct comparison against existing prompting baselines such as Gemba-MQM.

For all experiments, we adopt MQM-Errors, an LLM-based evaluation protocol that categorizes translation errors into six major types: addition, mistranslation, grammar, inappropriate, omission, and spelling. The total number of errors across these categories constitutes the final MQM-Errors score, which we use as our primary evaluation metric.

(System) You are a helpful translater. translate english to german.
(User) Pedestrian dies in hospital after crash with car in Cowdenbeath as man arrested ➔ *source_seg*
(Output) Fußgänger **stirbt** **nach Zusammenstoß mit Auto in Cowdenbeath** **im Krankenhaus**, während ein Mann festgenommen wurde. ➔ *target_seg*

(Evaluator_prompt)
English source: *source_seg*
German translation: *target_seg*
Identify errors in the translation and explain which part is incorrect and why.
(Addition_Evaluator_output)
Accuracy/Addition_Severity: minor
Explanation: The minor addition error in the German translation comes from the phrase "während ein Mann festgenommen wurde" (while a man was arrested), which implies a simultaneous occurrence that is not explicitly stated in the English source text. The English source mentions that a man was arrested but does not specify the timing relative to the pedestrian's death.
(Mistranslation_ Evaluator_output)
Accuracy/Mistranslation_Severity: minor ₩nExplanation: ...
(Grammar Evaluator_output)
Fluency/Grammar_Severity: minor ₩nExplanation: ...

(Refiner_prompt)
english source: *source_seg*
german translation: *target_seg*
Severity & Explanation: **Evaluators_output**
Make corrections to the translation to ensure minimal errors in the next evaluation.
Focus on improving accuracy, clarity, and context to reduce mistakes in future reviews.
(Refiner_output)
Fußgänger **stirbt im Krankenhaus** **nach Zusammenstoß mit Auto in Cowdenbeath**, **wobei** ein Mann festgenommen wurde.

Figure 4: Overview of Eval N + Refine N (Parallel + Merge). Each evaluator-refiner pair edits the translation based on a specific MQM criterion, and their outputs are merged into a final version.

To apply MQM-Errors consistently and at scale, we employ a metric agent based on the Qwen 2.5 32B model. Manual MQM evaluation by humans is highly labor-intensive, costly, and often inconsistent, making it impractical for large-scale benchmarking. An automated metric agent enables fast, reproducible, and fine-grained error detection aligned with MQM criteria. We fix the metric agent across all experiments to ensure evaluation consistency, avoiding the variability that could arise from using different LLMs or prompt settings. Among the candidates considered—Qwen 2.5 7B, Qwen 2.5 32B, and GPT-4o—the 32B model offered the best trade-off between cost-efficiency and evaluation accuracy, making it the most suitable and stable choice for our evaluation framework.

The evaluation was performed on a large number of sentences, allowing for a detailed analysis of both aggregate MQM-Errors scores and fine-grained error distributions. In each case, we report not only the raw error counts but also the relative percentage improvement over the translate-only baseline, providing a clear picture of each strategy's effectiveness.

A key goal of this work is to reduce the high cost and labor demands of human MQM evaluation by leveraging LLMs to automatically detect and correct common error types. By demonstrating that our framework significantly reduces MQM-Errors using a relatively small, open-source LLM as the base model, we show that high-quality post-editing is possible without relying on prohibitively expensive proprietary models. This highlights the practicality and accessibility of our method, enabling broader adoption and reproducibility even in resource-constrained settings.

5 Analysis

Our experimental results are reported in Table 1. Compared to both simple translation without evaluation and the single-prompt evaluation approach used in Gemba-MQM, our proposed method consistently yields the lowest MQM-Errors across all settings. Notably, for the Zh–En language pair, the Gemba-MQM approach not only failed to reduce errors but actually increased the total MQM-Errors, resulting in degraded translation quality. In contrast, our method significantly reduced the number of errors, producing more accurate and reliable post-edited translations.

Across all evaluated datasets and language pairs, we observe that prompting strategies incorporating explicit evaluation and refinement stages consistently outperform the translate-only baseline in terms of MQM-Error reduction. These results underscore the effectiveness of multi-pass, feedback-informed generation methods in enhancing translation quality.

For the En–De language pair on the WMT2023 dataset, the Gemba-MQM approach yielded only a marginal 0.8% reduction in MQM-Errors compared to the translate-only baseline, indicating limited effectiveness in correcting translation errors. In contrast, our proposed method, Multi Eval + Multi Refine (parallel + merge), achieved a substantial 12.6% reduction, clearly demonstrating its superiority. Even the simpler Multi Eval + Multi Refine (sequential) variant delivered a notable 3.6% improvement, underscoring that structured, multi-step prompting strategies offer significantly greater error correction capabilities than single-shot evaluations such as Gemba-MQM. These findings, as shown in Table 1, highlight the robustness of our approach.

On the more challenging WMT2024 En–De test set, a similar pattern emerges. The Multi Eval + Single Refine configuration again produced the lowest error count, with a 5.4% reduction, followed closely by Multi Eval + Multi Refine (parallel + merge) at 4.3%. These results reinforce the generalizability of the approach across data distributions and task difficulties.

For the Zh–En setting, which typically exhibits higher baseline error rates due to greater linguistic divergence and tokenization variability, the benefits of targeted refinement are even more pronounced. On WMT2023, the translate-only system yielded 7,284 errors, whereas our best-performing configuration, Multi Eval + Multi Refine (parallel + merge), reduced this number to 6,729—achieving a 7.6% improvement. Notably, the Single Eval + Single Refine setup—closely aligned with the Gemba-MQM approach—performed worse than the baseline, resulting in an increased number of MQM-Errors. This outcome clearly highlights the shortcomings of single-shot evaluation strategies in linguistically complex translation scenarios. In contrast, our method, which leverages lightweight open-source models and modular prompting with minimal computational overhead, proves both more effective and more accessible. These results provide strong empirical support for the effectiveness and practicality of our proposed methodology. (see Figure 3)

A similar trend was observed on the WMT2022 Zh–En test set, where the translate-only baseline resulted in 7,557 total errors. Once again, Multi Eval + Multi Refine (parallel + merge) delivered the best performance, reducing the error count to 7,031—a 7.0% relative improvement. Notably, all

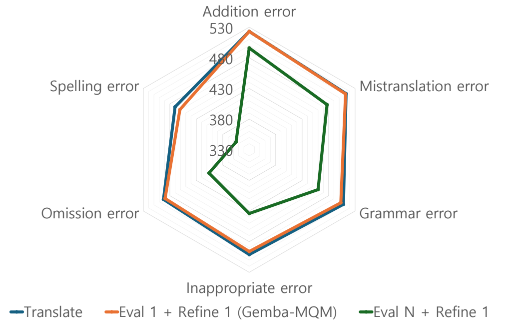


Figure 5: This figure shows the correction rate for each MQM error type. While most errors are effectively reduced, spelling errors are particularly well corrected.

refinement strategies except Single Eval + Single Refine yielded consistent gains, reinforcing the importance of scaling up the evaluation phase—either through increased quantity or greater diversity (e.g., multi-model setups)—as a key factor in enhancing translation quality. (see Figure 3)

In terms of error type breakdown (see Figure 5), refinement-based prompting was particularly effective at reducing spelling, omission, and inappropriate errors. These categories are especially challenging for single-pass translation models, which often lack sufficient context awareness or post-hoc reasoning capabilities—areas where refinement-based approaches excel. In contrast, addition and mistranslation errors showed more modest improvements, likely reflecting the current limitations of LLMs in detecting subtle semantic deviations without task-specific fine-tuning or supervision.

Overall, our findings demonstrate that utilizing LLMs to automatically identify and reduce MQM errors can substantially reduce the cost and effort traditionally associated with human evaluation. By adopting a small, open-source LLM as the base model, we show that it is possible to achieve significant improvements in translation quality through effective MQM-Error reduction. This indicates that reliance on large, expensive LLMs is not essential to obtain high-quality, reliable translation outputs. Our approach underscores the practical advantage of lightweight and accessible models, which enable robust and scalable translation quality control while minimizing computational and financial resources. This makes advanced error detection and correction methods more feasible and accessible to a broader range of users and applications.

In addition to experiments with Qwen models, we further validated our proposed method using GPT-4o and GPT-4o Mini. For both models, our multi-evaluator framework consistently outperformed baseline methods, including the single-prompt Gemba-MQM and translate-only approaches. Remarkably, even the lightweight GPT-4o Mini, despite its smaller size, achieved significant reductions in MQM-Errors through our specialized evaluators and refinement strategy. These results demonstrate the robustness and versatility of our approach across different LLM architectures and scales, highlighting its practical effectiveness in both resource-limited and resource-rich environments.

Error Type-Specific Evaluator Effectiveness We conducted ablation studies by selectively removing or adding evaluators corresponding to specific error types, such as spelling and omission, to assess their individual contributions to overall error reduction. Notably, removing evaluators for more subjective or ambiguous error categories—such as style/awkwardness and fluency/punctuation—resulted in improved overall MQM-Errors reduction. This suggests that excluding these less well-defined error types allows the refinement process to focus more effectively on clear-cut errors, thereby enhancing the quality of post-edited translations. Our findings highlight the importance of carefully selecting error dimensions for evaluation to maximize the impact of specialized evaluators in translation refinement.

Sequential vs. Parallel + Merge Refinement: Quantitative Comparison We quantitatively compare two refinement strategies for post-editing: sequential refinement and parallel refinement with a merging step. Sequential refinement applies specialized refiners one after another, progressively correcting errors. In contrast, parallel refinement runs all refiners simultaneously and merges their outputs to resolve conflicts and integrate complementary corrections.

Our results show that while both approaches improve translation quality over the baseline, the parallel + merge strategy consistently achieves greater reductions in MQM-Errors. The merge step effectively combines corrections from multiple evaluators, leading to more comprehensive error correction. Additionally, parallel refinement offers better scalability and flexibility compared to the sequential approach, which can suffer from error propagation and limited adaptability.

These findings suggest that parallel evaluation and merging provide a more effective and modular framework for leveraging specialized error evaluators in translation refinement.

Impact of Merging Step The merging process following parallel refinement plays a critical role in resolving conflicts among multiple evaluator outputs and integrating diverse corrections into a coherent final translation. While our current merging strategy demonstrates improvements in overall translation quality, further enhancements in this step could yield even more effective error resolution. Future work will focus on developing more sophisticated merge algorithms to better reconcile conflicting edits and optimize the final output, thereby maximizing the benefits of multi-evaluator refinement frameworks. This analysis underscores the importance of the merging step as a key component in achieving high-quality post-edited translations.

6 Conclusion

In this work, we explored a range of prompting strategies for improving machine translation quality via large language models, with a particular focus on structured evaluation and refinement workflows. By integrating systematic multi-pass evaluations (Multi eval) and targeted refinements (Multi refine or Single refine), we demonstrated consistent reductions in MQM-Errors error rates across multiple datasets and language pairs, including both high-resource (En–De) and more challenging, structurally divergent (Zh–En) scenarios.

Our findings indicate that prompting strategies which explicitly separate the evaluation and generation phases—particularly those using parallel evaluation followed by aggregated refinement—outperform conventional translate-only baselines by substantial margins. These gains are most pronounced in error types that benefit from iterative reasoning and context verification, such as omission and inappropriate content, while improvements for more subtle errors like mistranslations remain comparatively modest.

This suggests that LLMs, when properly prompted, can emulate some aspects of human post-editing workflows, and that prompting design plays a critical role in maximizing their effectiveness. In future work, we plan to investigate the role of LLM diversity in evaluation stages, integrate external error annotation resources to guide refinement decisions, and extend this framework to other generation tasks such as summarization and data-to-text generation.

7 Limitations

Despite the effectiveness of our proposed method, several limitations remain. First, while assigning dedicated evaluators to each MQM dimension enables more targeted error detection, the interactions and overlaps between dimensions (e.g., omission vs. fluency) are not explicitly modeled. Second, the current merging strategy in the refinement stage is heuristic and may not optimally reconcile conflicting or complementary suggestions from different evaluators, leaving room for more advanced aggregation techniques in future work.

References

Aljoscha Burchardt Arle Lommel and Hans Uszkoreit. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. In *LREC*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *NAACL*.

Tom Kocmi and Christian Federmann. 2023. Gemba-mqm: Detecting translation quality error spans with gpt-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775.

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.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Lucia Specia et al. 2017. Translation quality and post-editing. In *Handbook of Translation Studies*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yiwei Zheng, Yixin Nie, and Mohit Bansal. 2024. Decomposed prompting for large language model evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.