PromptOptMe: Error-Aware Prompt Compression for LLM-based MT Evaluation Metrics

Daniil Larionov[1,2] **Steffen Eger**[1,3]
[1] NLLG, [2] University of Mannheim, [3] University of Technology Nuremberg
daniil.larionov@uni-mannheim.de

Abstract

Evaluating the quality of machine-generated natural language content is a challenging task in Natural Language Processing (NLP). Recently, large language models (LLMs) like GPT-4 have been employed for this purpose, but they are computationally expensive due to the extensive token usage required by complex evaluation prompts. In this paper, we propose a prompt optimization approach that uses a smaller, fine-tuned language model to compress input data for evaluation prompt, thus reducing token usage and computational cost when using larger LLMs for downstream evaluation. Our method involves a two-stage fine-tuning process: supervised fine-tuning followed by preference optimization to refine the model's outputs based on human preferences. We focus on Machine Translation (MT) evaluation and utilize the GEMBA-MQM metric as a starting point. Our results show a 2.37× reduction in token usage without any loss in evaluation quality. This work makes state-of-the-art LLM-based metrics like GEMBA-MQM more cost-effective and efficient, enhancing their accessibility for broader use.

1 Introduction

The rapid advancement of Natural Language Generation (NLG) technologies has led to an increasing reliance on automated systems for producing human-like text across various domains, including machine translation (MT). As these systems become more prevalent, the need for effective and efficient evaluation metrics to assess generated content quality has also grown. Evaluating NLG systems is a fundamental challenge in the field of Natural Language Processing (NLP).

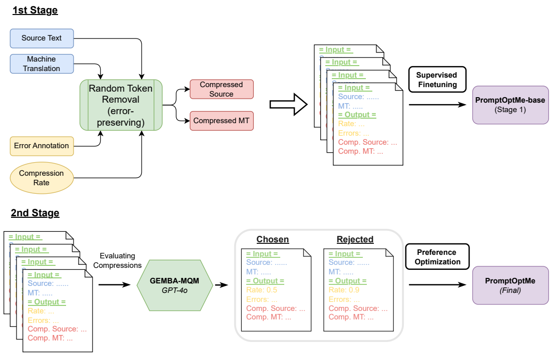

Figure 1: The two-stage model training approach used in PromptOptMe. At the first stage, the model is fine-tuned in a supervised way to adapt it for the compression task and prompt format. At the second stage, we utilize preference data obtained through evaluation of compressed prompts to train the model to select the best compression for each example.

Traditionally, automatic evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019), and MoverScore (Zhao et al., 2019) have been widely used due to their simplicity and ease of implementation. BLEU and ROUGE measure n-gram overlap between the generated text and reference texts, but they often fail to capture the semantic meaning and penalize legitimate lexical variations (Callison-Burch et al., 2006). BERTScore and MoverScore leverage contextual embeddings from pre-trained language models to compute similarity at a deeper semantic level; however, they can still struggle with capturing nuanced errors in meaning and may not effectively evaluate aspects like factual correctness or language fluency (Freitag et al., 2021a; Kocmi et al., 2021; Chen and Eger, 2023).

Subsequently, trained evaluation metrics such as COMET (Rei et al., 2022a), BLEURT (Sellam et al., 2020), and xCOMET (Guerreiro et al., 2023) emerged. These models are trained on human-annotated datasets to predict quality scores, allowing them to better align with human judgments. Despite their improved quality, these metrics require substantial amounts of labeled data

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 11807–11820
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

for training and may not generalize well across different tasks or domains.

With the advent of large language models (LLMs) like GPT-4 (Achiam et al., 2023), researchers have started to explore their potential for NLG evaluation through prompting instead of traditional training. Metrics such as GEMBA-DA (Kocmi and Federmann, 2023b), GEMBA-MQM (Kocmi and Federmann, 2023a), AutoMQM (Fernandes et al., 2023), and G-Eval (Liu et al., 2023) utilize LLMs by providing instructions and few-shot examples within prompts. This approach offers several key benefits: **a)** elimination of task-specific training, as LLMs can perform evaluations based on prompts without the need for labeled datasets; **b)** high-quality results, with LLM-based metrics demonstrating superior alignment with human judgments and more effectively capturing linguistic nuances and contextual information; **c)** flexibility and generalization, since these models can be easily adapted to different tasks and domains by modifying prompts, offering greater flexibility compared to traditional trained metrics.

Despite these advantages, using LLMs like GPT-4 for evaluation is computationally expensive due to the extensive token usage required by elaborate prompts. For instance, the original GEMBA-MQM prompt typically requires between 1100 and 1200 tokens per example, accounting for exhaustive instructions and three few-shot examples. If we aim to evaluate this metric on 60k examples from the WMT22 Metrics Challenge Test Set (Freitag et al., 2022), the total token usage would be on the order of:

$$\text{Total Tokens} \approx 60\text{k} \times 1200 \approx 72\text{M tokens.}$$

Given that GPT-4 pricing is $10 per million tokens used, the estimated cost for evaluating the entire dataset would be:

$$\text{Cost per full run} \approx \frac{72\text{M}}{1\text{M}} \times 10 \approx \$720.$$

These substantial costs present a barrier to practical deployment, especially in budget-constrained settings. It also makes LLM-based metrics like GEMBA-MQM egregiously expensive for large-scale evaluation scenarios, like online re-ranking of MT-systems outputs or web-scale dataset processing (Peter et al., 2023).

To address these challenges, we introduce **PROMPTOPTME**,[1] a novel approach to prompt optimization that reduces token usage and computational costs in LLM-based evaluation by utilizing a smaller, fine-tuned language model to compress input data without sacrificing necessary information. We summarize our contributions as follows: **a)** We propose a two-stage fine-tuning process for **PROMPTOPTME**, involving supervised fine-tuning and preference optimization (Hong et al., 2024), to refine the model's outputs based on actual metric behavior with compressed inputs. **b)** We apply **PROMPTOPTME** to MT, achieving a 2.32× reduction in token usage for MT evaluation, without any loss in evaluation quality.

Our work enhances the accessibility of advanced LLM-based evaluation metrics by making them more cost-effective and efficient for broader use in real-world NLG applications, thereby promoting diversity and inclusion in NLP research and application by enabling participation from under-resourced communities (Belouadi and Eger, 2023).

2 Related Work

The field of prompt optimization has seen substantial activity in recent years. This section covers works related to prompt optimization, efficient evaluation, and explainable metrics, highlighting how our approach builds upon and differs from existing research.

We begin by examining the landscape of **explainable evaluation metrics** (Kaster et al., 2021; Opitz and Frank, 2021; Leiter et al., 2023). Leiter et al. (2024) propose a taxonomy for such metrics, within which the GEMBA-MQM metric falls into the category of fine-grained error metrics. Naturally, these metrics critically depends on preservation of error-spans within source text and translation. Unlike xCOMET (Guerreiro et al., 2023), GEMBA-MQM's sentence-level scores are derived solely from extracted errors and their severities, adhering to the MQM guidelines (Freitag et al., 2021a). Due to this fact, we designed our method to fine-tune the model to preserve the error-spans in the source and translation text by utilizing the WMT MQM-annotated dataset. Further, PROMPTOPTME can

[1] PROMPTOPTME stands for "**Prompt Opt**imization for **Me**trics" and is inspired by PrExMe (Leiter and Eger, 2024), a recent method to investigate prompt exploration for MT evaluation metrics.

also be seen as explaining prompt-based evaluation metrics because it yields additional insights into which prompt parts are relevant for a metric to perform with high-quality.

In the realm of **prompt compression and optimization**, several notable approaches have been developed. LLMLingua (Jiang et al., 2023) introduces a coarse-to-fine prompt compression method, employing a budget controller and an iterative token-level compression algorithm. This approach utilizes a smaller language model to compute token importance, substantially reducing token usage and inference latency across various tasks. Building upon this foundation, LLMLingua-2 (Pan et al., 2024) proposes a data distillation procedure to derive knowledge from LLMs for efficient and faithful task-agnostic prompt compression. By formulating prompt compression as a token classification problem and leveraging a Transformer encoder, LLMLingua-2 further improves compression efficiency and generalizability. Unlike both of those approaches, we specifically target the task of LLM-based evaluation metrics. We incorporate preference optimization based on actual metric behavior, tailoring the prompt compression to preserve essential evaluation information.

Black-Box Prompt Optimization (BPO) (Cheng et al., 2024) applies prompt optimization for black-box LLM alignment, using a prompt preference optimizer trained on human preference data. This model-agnostic approach enhances the alignment of LLM outputs with human intents without requiring access to model parameters or additional training. Our method shares similarities in leveraging preference learning, but we extract these preferences by exploiting measurable quality differences during prompt execution with optimized outputs.

PRewrite (Kong et al., 2024) proposes an automated prompt engineering approach using reinforcement learning to rewrite prompts for improved prediction quality. While both PRewrite and our method optimize prompts without modifying the underlying language model, our approach distinguishes itself by simultaneously striving to reduce prompt size and improve quality.

In the domain of **efficient MT evaluation metrics**, several approaches have been proposed to address the computational challenges posed by large trained metrics. xCOMET-lite (Larionov et al., 2024), COMETinho (Rei et al., 2022b), and FrugalScore (Kamal Eddine et al., 2022) all employ techniques such as pruning, quantization, and distillation to create smaller, more efficient models. While these methods achieve impressive compression ratios, resulting in models with under 500M parameters, they often experience a drop in correlation with human judgment. Our approach aims to mitigate this quality loss by continuing to rely on large backbone LLMs while improving efficiency through targeted prompt optimization.

Larionov et al. (2023) explore an alternative approach to creating an efficient version of BERTScore and MoverScore by replacing the underlying encoder model with smaller alternatives, such as pruned or distilled models. We touch this subject too, by evaluating our prompt compression model not only on large backbone LLM GPT-4o, but also on smaller models: GPT-4o mini and LLaMa 3.2. As we demonstrate in Section 5, our approach proves effective across different model sizes, improving evaluation efficiency even on smaller LLMs.

3 Method

In this section, we detail our two-stage approach for prompt optimization aimed at reducing token usage and computational costs in LLM-based evaluation metrics. We first look into the optimization of prompt inputs (source texts and machine translations) and later detail our approach to compressing the rest of the prompt. The first stage involves supervised fine-tuning of a language model to learn the specifics of the prompt compression task. The second stage employs preference optimization using the Odds-Ratio Preference Optimization (ORPO) algorithm (Hong et al., 2024) to refine the model's outputs based on the observed behavior of the metric. We chose this algorithm due to its specific focus on reducing the likelihood of generating the rejected responses, which are, in our case, low-quality compressed source and translation texts. The overall approach is visualized in Figure 1.

3.1 Stage One: Supervised Fine-Tuning

In the first stage, we perform supervised fine-tuning of a language model to enable it to effectively compress input texts while preserving essential information required for accurate evaluation. In the context of the MT evaluation task, the model is trained to accept original uncompressed source texts and their respective machine translations and to generate three outputs:

1. A compression rate r, selected as a floating-point number from the set $\mathcal{R}_{comp} = \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, where 1.0 indicates no compression.
2. A list of potential substrings from the source and MT texts that contain translation errors.
3. The compressed versions of the source and MT texts.

We achieve two primary objectives by prompting the model to extract potential error spans from the source and MT texts. First, this process teaches the model to actively search for translation errors, thereby enhancing its understanding of MT evaluation. Second, by conditioning the compression of the texts on these identified error spans, the model ensures that critical information related to translation errors is preserved in the compressed outputs. This approach allows the compressed texts to retain essential details necessary for accurate evaluation, even with reduced token usage.

To construct the training dataset for supervised fine-tuning, we utilize Multidimensional Quality Metrics (MQM)-annotated data from the WMT Metrics shared tasks (Freitag et al., 2021b). For each example in the dataset, we extract error-annotated spans based on the MQM annotations. We then perform random token removal to generate compressed texts, ensuring that the identified error spans remain intact in the compressed versions. The compression rate r is randomly selected from the predefined set for each example.

Formally, let S denote the original source text, T the original MT text, and E the set of error spans extracted from MQM annotations. The compressed source text S' and compressed MT text T' are generated by removing tokens from S and T, respectively, while preserving all tokens within E.

This supervised fine-tuning process trains the model to perform the prompt compression task effectively, balancing the reduction in token count with the retention of essential information required for accurate evaluation.

3.2 Stage Two: Preference Optimization

In the second stage, we apply ORPO to further tune the model based on preferences between various compressions, enabling it to select the optimal compression for each example. The goal is to optimize the trade-off between token reduction and the preservation of evaluation quality.

To create the preference dataset needed for ORPO, we generate compressed versions of each example using the previously defined set of compression rates. For each compression rate r, we obtain compressed texts S'_r and T'_r using the fine-tuned model from Stage One. We then incorporate these compressed texts into the original GEMBA-MQM prompt and submit them to GPT-4o to obtain evaluation scores s_r.

The evaluation scores are compared to the score $s_{1.0}$ obtained from the uncompressed texts (with $r = 1.0$). For each example, we identify:

- The *chosen* compression with rate r_{chosen} whose evaluation score $s_{r_{chosen}}$ has the minimal absolute difference from $s_{1.0}$. If multiple compression rates have equally minimal differences, we select the one with the lowest r to prioritize higher compression.
- The *rejected* compression with rate $r_{rejected}$, whose evaluation score $s_{r_{rejected}}$ has the maximal absolute difference from $s_{1.0}$. If multiple compression rates have equally maximal differences, we select the one with highest r.

Formally, we define the absolute score difference $\Delta_r = |s_r - s_{1.0}|$. The chosen and rejected compression rates satisfy:

$$r_{chosen} = \arg\min_{r \in \mathcal{R}_{comp}} \Delta_r,$$

$$r_{rejected} = \arg\max_{r \in \mathcal{R}_{comp}} \Delta_r.$$

Using these preferences, we train the model from Stage One using the ORPO algorithm, which adjusts the model's parameters to increase the likelihood of generating outputs corresponding to the chosen compression rates over the rejected ones. The ORPO loss function is defined based on the odds ratio of the probabilities assigned to the chosen and rejected outputs:

$$\mathcal{L}_{\text{ORPO}} = \mathbb{E}_{(x,y_c,y_r)}\left(\mathcal{L}_{\text{SFT}} + \lambda \cdot \mathcal{L}_{\text{OR}}\right)$$

$$\mathcal{L}_{\text{OR}} = -\log \sigma\left(\log \frac{odds_\theta(y_c|x)}{odds_\theta(y_r|x)}\right)$$

$$odds_\theta(y|x) = \frac{P_\theta(y|x)}{1 - P_\theta(y|x)}$$

where x, y are a prompt and a completion respectively, $P_\theta(y_c|x)$ and $P_\theta(y_r|x)$ are the probabilities assigned by the model with parameters θ to the chosen and rejected outputs, respectively. $\mathcal{L}_{SFT}$ refers to standard cross-entropy loss used for language modeling.

This preference optimization process teaches the model to select compressions that yield evaluation scores closest to those obtained with uncompressed texts, effectively maintaining evaluation quality while reducing token usage. By prioritizing compression rates that minimize score discrepancies, the model learns to balance the trade-off between efficiency and quality.

3.3 Simplified Prompts

In the context of MT evaluation, the source texts and their translations are typically short, often comprising up to two or three sentences. However, the GEMBA-MQM prompt includes long instructions, outlining the entire MQM error typology. Those instructions are included in both few-shot examples and in the target example. We hypothesize that this surplus of tokens raises computational costs without necessarily enhancing evaluation quality.

Initially, we also trained a prompt optimization model using preference data to compress the instruction component of the prompt. However, upon evaluation, we found that the compressed instructions generated were effectively the same across different inputs. This uniformity suggested that a fixed simplified instruction could suffice without dynamic compression per example.

Moreover, as demonstrated in Section 4, the use of these simplified instructions did not adversely affect the metric quality. The evaluation quality remained comparable to that achieved with the original, more verbose GEMBA-MQM prompt.

Consequently, we adopt a fixed simplified instruction template for our evaluations and discontinued further instruction compression. For reference, both the original GEMBA-MQM prompt and our simplified version are provided in the Appendix A.

4 Experiments

In this section, we evaluate the effectiveness of our proposed prompt optimization approach in reducing token usage while maintaining evaluation quality in LLM-based evaluation metrics. We conduct experiments on MT evaluation and assess the compression efficiency and the impact on evaluation quality.

As a base model for fine-tuning, we utilize the LLaMA-3.2 (Dubey et al., 2024) model in two variants — 1B and 3B parameters — chosen for its extensive vocabulary and multilingual pre-training beneficial for handling diverse language pairs.

Stage One: Supervised Fine-Tuning For the first stage, we use the WMT Metrics shared task datasets with MQM annotations from the years 2020 to 2022 (Freitag et al., 2021b). We exclude the 'news' portion of WMT22 dataset for validation purposes. The remaining training data consists of approximately 145k examples across three language pairs: English-Russian, English-German, and Chinese-English.

Following the approach described in Section 3, we construct prompts with system instructions for the model. The prompt follows this template:

```
System: You are a helpful AI assistant that intelligently compresses and summarizes the given Machine Translation outputs for further evaluation.

User: Compress the following MT input and output: Source:“‘{src-text}“‘ MT:“‘{mt-text}“‘
```

The expected model completion is:

`Assistant:Rate =` r

`Quality-relevant parts of Source:` [*error-span-1*]; ... ;[*error-span-N*] `or None`

`Quality-relevant parts of Translation:` [*error-span-1*]; ...; [*error-span-N*] `or None`

`Compressed Source:`“‘{*compressed-src*}“‘

`Compressed MT:`“‘{*compressed-mt*}“‘

Here, {*src-text*} and {*mt-text*} denote the original source and machine-translated texts. We provide an example of the text compression with trained model in Figure 2, where we demonstrate both model inputs — source text and machine translation — as well as the outputs — compression rate, error spans and compressed texts.

<model-prompt>
Example Source: So they’re crossing thousands of kilometers in a fraction of a second, and as they do so, they not only curve space, but they leave behind in their wake a ringing of space, an actual wave on space-time.
Example MT: Sie überqueren also in Sekundenbruchteilen Tausende von Kilometern und krümmen dabei nicht nur den Raum, sondern hinterlassen auch ein Klingeln des Raums, eine tatsächliche Welle der Raumzeit.
</model-prompt>
Example Reference Translation: Sie können Tausende von Kilometern im Bruchteil einer Sekunde zurücklegen. Und während sie das tun, beugen sie nicht nur den Raum, sondern sie hinterlassen in ihrem Kielwasser ein Klingen des Raumes, eine echte Welle aus Raumzeit.
Example Gold MQM Annotations: [’start’: 4, ’end’: 14, ’severity’: ’minor’, ’text’: ’überqueren’, ’start’: 178, ’end’: 179, ’severity’: ’minor’, ’text’: ’ ’]
<model-generation>
Rate: 0.5
Quality-relevant parts of Source: None
Quality-relevant parts of MT: [Sie überqueren also in Sekundenbruchteilen Tausende von Kilometern];
Compressed Source: “‘thousands kilometers fraction of a second curve spaceringing of spacean actual wave“‘
Compressed MT: “‘Sie überqueren Sekundenbruchteilen Tausende von Kilometern krümmen nicht eine tatsächliche Welle“‘
</model-generation>

Figure 2: Example of Model-generated compression. Reference translation and Gold MQM annotations are provided for readers convenience and are not part of the model prompt.

We train the model for one epoch using a batch size of 64 and a learning rate of 2×10^{-5} with cosine warmup over 6% of the training steps. We apply a weight decay of 0.01. For efficient training, we use Low-Rank Adaptation (LoRA) adapters (Hu et al., 2022), with the following parameters: rank $r = 32$, scaling factor $\alpha = 16$, and dropout rate of 0.5.

Stage Two: Preference Optimization For the second stage, we construct a preference dataset by selecting a subset of 20k examples from the training data. For each example, we generate compressed versions at each compression rate from our predefined set, resulting in eight different compressions per example.

We incorporate these compressed examples into the original GEMBA-MQM prompt and evaluate them using the GPT-4o model to obtain evaluation scores. Based on these scores, we select the *chosen* and *rejected* compressions following the procedure described in Section 3.

Using this preference dataset, we fine-tune the model from Stage One employing ORPO. We train for three epochs with a batch size of 64, a learning rate of 1×10^{-5}, and cosine warmup over 6% of training steps. We set the ORPO λ parameter to 0.1 following the original paper authors.

Evaluation Procedure To evaluate the effectiveness of our model, we apply it to compress the examples in the test set. This compression is performed on both the few-shot examples and the target example within the GEMBA-MQM prompts. We then conduct MT evaluation using these compressed prompts with proprietary LLMs GPT-4o and GPT-4o-mini and the openly-available LLaMA-3.2-Vision model with 90B parameters. In our experiments, we utilize both the original and simplified prompts (as described in Section 3.3) and generate outputs in both plain text format and in JSON.

We assess the quality of the evaluations by computing pairwise accuracy at the system level, as suggested by Deutsch et al. (2023), comparing our results with human judgments. Additionally, we calculate segment-level Pearson r and Kendall-τ correlation for each language pair to measure the agreement between our evaluations and human assessments. In addition to the validation set, we use WMT23 Metrics Challenge Freitag et al. (2023) dataset to perform meta-evaluation of our approaches and compare them with existing metrics.

To quantify the efficiency gains, we measure the cumulative token usage during the evaluations. This provides insights into the computational cost savings achieved through our prompt optimization approach. To compare the effectiveness of our approach, we also include two baseline results obtained using the LLMLingua-2 (Pan et al., 2024) prompt compression method. In those cases, we have applied *microsoft/llmlingua-2-xlm-roberta-large-meetingbank* to compress source texts and machine translations similarly as we do with PromptOptMe. We fix and test two compression rates: 30% and 50%.

5 Results

In this section, we present the results of our experiments evaluating the effectiveness of our proposed prompt optimization approach.

As shown in Table ??, the baseline model, **GPT-4o ref**, uses GPT-4o with the full (uncompressed) GEMBA-MQM prompt, ends up with a total amount of 19M input tokens used for entire test set of 16k examples and serving as a reference

Model + Prompt	Token Usage	Reduction Rate	Pairwise Accuracy	En-Ru τ	En-De τ	Zh-En τ
GPT-4o ref	19M	1.00	0.7789	0.4365	0.3950	0.3692
GPT-4o mini ref	19M	1.00	0.7631	0.3723	0.3165	0.3472
LLaMa3.2-90B ref	20M	1.00	0.7526	0.3416	0.2920	0.3576
GPT-4o lite	10.4M	1.84	0.7736	0.3838	0.3207	0.2890
GPT-4o lite						
PROMPTOPTME-3B	8.07M	2.37	0.7736	**0.4455**	**0.4065**	**0.3738**
PROMPTOPTME-1B	8.8M	2.15	0.7644	0.4122	0.3900	0.3743
GPT-4o mini lite						
PROMPTOPTME-3B	8.07M	2.37	0.7842	0.3177	0.3238	0.3596
PROMPTOPTME-1B	8.8M	2.15	0.7531	0.3089	0.3125	0.3468
LLaMa3.2-90B lite						
PROMPTOPTME-3B	8.8M	2.27	0.7526	0.3505	0.3191	0.3123
PROMPTOPTME-1B	9.0M	2.22	0.7345	0.3203	0.2891	0.3000
GPT-4o lite						
LLMLingua2 @ 50%	7.6M	2.5	0.4736	0.0055	0.0492	0.1247
LLMLingua2 @ 30%	7.0M	2.7	0.5421	0.00	0.03	0.1949

Table 1: Evaluation results for machine translation with different prompting strategies and models. **GPT-4o ref** refers to the original GPT-4 with the full (uncompressed) GEMBA-MQM prompt. **lite** denotes the simplified prompting approach with JSON-formatted output. PROMPTOPTME-3B and PROMPTOPTME-1B represent our prompt optimization models based on LLaMA 3.2 with 3B and 1B parameters, respectively, used for input compression. **Token Usage** shows the total token usage. **Reduction Rate** is the token reduction rate compared to the baseline (**-ref** for each model). **Pairwise Accuracy** stands for pairwise system-level accuracy. **En-Ru** τ, **En-De** τ, and **Zh-En** τ refer to the segment-level Kendall τ correlations for English-Russian, English-German, and Chinese-English language pairs, respectively. Boldface indicates the best quality in each column.

point. The simplified prompting approach from Section 3.3, denoted as **GPT-4o lite**, reduces the token usage to 10.4M tokens, achieving a reduction rate of 2.04× while maintaining a similar pairwise accuracy of 0.7736, but lower segment-level Kendall τ correlations across the language pairs (0.3838 for En-Ru, 0.3207 for En-De, and 0.2890 for Zh-En).

By applying our prompt optimization model, PROMPTOPTME-3B, with **GPT-4o lite** prompt template, we achieve the highest reduction rate of 2.37×, reducing token usage from 19M to 8.3M tokens. Surprisingly, PROMPTOPTME-3B attains the best segment-level Kendall τ correlations across all language pairs, with 0.4455 for En-Ru, 0.4065 for En-De, and 0.3738 for Zh-En, fully recovering reduced scores in the baseline, while also retaining the same level of system-level pairwise accuracy.

This trend continues with other backbone LLMs. For instance, using PROMPTOPTME-3B with **GPT-4o mini lite** prompt, we, again, achieve the same reduction rate of 2.37× and observe an improvement in pairwise accuracy to **0.7842**, which is a 2.77% increase over the **GPT-4o mini ref** baseline. However, the segment-level correlations exhibit mixed results: while there is a substantial decrease for En-Ru (14.67% lower than the baseline), there is a positive improvement for En-De (an increase of 2.30%) and Zh-En (a 3.57% increase).

Similarly, for the **LLaMa3.2-90B** model, applying PROMPTOPTME-3B results in a reduction rate of 2.27×. The pairwise accuracy remains the same as the baseline (0.7526), but the segment-level Kendall τ correlations show modest improvements for En-Ru (a 2.61% increase) and En-De (9.28% higher), while Zh-En experiences a decrease of 12.68%.

Thus, overall, we see substantial efficiency gains and simultaneously often an increase in quality of the resulting metrics; however, there are cases when metric quality reduces, sometimes considerably.

In comparison with the baseline approaches with the LLMLingua-2, we see dramatically decreased quality in both compression settings, as compared to our approach and baseline metrics. For the system-level pairwise accuracy, we observe a 31%-40% decrease from the original uncompressed metric. On a segment level, the quality decrease is even more catastrophic. For two out of three language pairs, the correlation plunges to near zero. For Zh-En, it drops to 0.19-0.12. This indicates that

LLMLingua-2 substantially damages the prompt to the point of complete non-usability.

In Table 2, we extend our evaluation to the full WMT23 Metrics Challenge dataset, to further assess the generalization of our prompt optimization model. The reduction rate on the WMT23 Metrics Challenge is slightly smaller (1.7× compared to 2.37× on the WMT22 'news' subset). This likely reflects a more cautious compression for the Hebrew–English pair, which was unseen during training. Despite this, the model preserves evaluation quality, compared to uncompressed metrics, as indicated by the Pairwise Accuracy.

6 Discussion

Our proposed prompt optimization model, PROMPTOPTME, demonstrates substantial improvements in computational efficiency for machine translation evaluation. By integrating PROMPTOPTME with a simplified prompting approach, we achieve substantial reductions in token usage—up to 2.37× less than the baseline—while maintaining or even improving evaluation quality across various metrics. These findings suggest that it is possible to compress inputs substantially while maintaining evaluation quality. We speculate that outputs of few-shot examples, which are not compressed in any way in our experiments, play a crucial role in establishing model quality in MT evaluation, as they themselves cover all error severity levels as well as error categories from MQM typology. Preserving entire source and translation texts, in turn, appears to be redundant. We are able to retain quality while preserving only part of the text, which includes error-spans.

The superior quality of PROMPTOPTME-3B, compared to the smaller PROMPTOPTME-1B model, indicates that a larger prompt optimization model is more effective at compressing input data efficiently. Specifically, PROMPTOPTME-3B refers to our prompt optimization model based on LLaMA 3.2 with 3 billion parameters, achieves 10% (2.37 vs. 2.15) higher compression rate on average, along with 2%-3% higher pairwise accuracy, compared to PROMPTOPTME-1B. It demonstrates the importance of model capacity in the compression process. Further experiments with larger LLMs could potentially help get a better understanding of scaling laws for that particular task.

When applying PROMPTOPTME-3B to different backbone LLMs, such as **GPT-4o mini** and **LLaMa3.2-90B**, we observe varying results. For **GPT-4o mini**, we achieve an improved pairwise accuracy compared to the baseline, with a reduction rate of 2.37×. In contrast, when applying PROMPTOPTME-3B to **LLaMa3.2-90B**, the pairwise accuracy remains consistent with the baseline, and we observe modest improvements in segment-level Kendall τ correlations for some language pairs. In both cases, we notice a slight degradation in segment-level correlations in one of the language pairs, however. Nonetheless, we can conclude that our prompt optimization model achieves generalization across different backbone LLMs, despite being trained only on preferences obtained through GPT-4o.

Additionally, compared to baseline approaches using LLMLingua-2, our method clearly outperforms in both system-level and segment-level evaluations. The substantial quality degradation observed with LLMLingua-2, particularly at the segment level, indicates that it may not be suitable for effective prompt compression in MT evaluation tasks. This result is in agreement with our hypothesis that preserving quality-relevant spans in text is essential for maintaining evaluation quality. LLMLingua-2, as a non-task-specific prompt compression method, is not trained to take that into account.

Evaluation on the WMT23 Metrics Challenge further supports the robustness of our prompt optimization model. The preserved pairwise accuracy and high pearson correlation indicates that our approach generalizes across different datasets and language pairs. Future work may investigate adaptive compression strategies to better handle low-resource or previously unseen language pairs.

7 Conclusion

In this paper, we introduced **PROMPTOPTME**, a prompt optimization approach designed to reduce token usage and computational costs in large language model-based evaluation metrics. By leveraging a smaller, fine-tuned language model to compress input data, we achieved substantial reductions in token usage without compromising evaluation quality. Specifically, **PROMPTOPTME** reduced token usage by up to 2.37× while

Model + Prompt	Token Usage	Reduction Rate	Pairwise Accuracy	En–De r	He–En r	Zh–En r
GPT-4o ref	60.32M	1.00	0.896	0.479	0.422	0.523
GPT-4o mini ref	60.32M	1.00	0.928	0.410	0.433	0.458
GPT-4o lite	40.63M	1.48	0.936	0.610	0.424	0.547
GPT-4o-mini lite	40.63M	1.48	0.916	0.490	0.419	0.467
GEMBA-MQM[noref]	–	–	0.944	0.502	0.401	0.449
GPT-4o						
PromptOptMe-3B	34.81M	1.73	0.944	0.587	0.453	0.574
GPT-4o-mini						
PromptOptMe-3B	34.81M	1.73	0.932	0.508	0.459	0.516

Table 2: Evaluation results on WMT23, computed using ‘mt-metrics-eval‘ evaluation framework. En–De r, He–En r and Zh–En r stand for segment-level Pearson-r correlation with human judgment for respective language pairs. ‘GEMBA-MQM[noref]‘ represent results of the original submission of GEMBA-MQM to the WMT23 Metrics Challenge.

maintaining or improving segment-level Kendall τ correlations across multiple language pairs as well as system-level pairwise accuracy.

Our approach enhances the practicality of LLM-based evaluation metrics, making them more cost-effective and accessible for large-scale natural language generation (NLG) applications. By addressing the computational expense associated with extensive prompt token usage, **PromptOptMe** enables more efficient evaluations. This reduction in computational cost makes state-of-the-art MT evaluation more accessible to under-resourced researchers and students, who may have limited access to computational resources or funding for extensive LLM usage. By lowering the barriers to high-quality evaluation, **PromptOptMe** democratizes the ability to conduct advanced research and development in machine translation. Our approach is especially useful in this case, as it does not require a compromise in metric quality, offering no quality drop while decreasing the token usage. Moreover, we are able to effectively compress the inputs for smaller LLMs as well, such as GPT-4o mini and LLaMa 3.2 90B, which makes SOTA evaluation metrics even more accessible.

While our experiments focused on machine translation, where error highlights and annotations are common, we acknowledge that extending this approach to other NLG tasks may present challenges, particularly in scenarios where such detailed error annotations are not available. Future work could explore adapting **PromptOptMe** to other NLG evaluation tasks, investigating how input compression affects evaluation quality in contexts without explicit error spans, and determining the generalizability of our method across various domains.

It is also important to note, that the optimized prompts generated by **PromptOptMe** are not guaranteed to strictly adhere to the MQM error typology. This divergence from the standard evaluation framework may impact the consistency and interpretability of the evaluation results, especially when comparing outputs across different systems or studies. Future work should focus on enhancing the alignment of compressed prompts with established evaluation standards to improve the reliability and comparability of the assessments.

We plan to release the code and models to the public at `https://github.com/NL2G/promptoptme` to facilitate further research and application in this area. We believe that **PromptOptMe** offers a promising direction for efficient high-quality LLM-based evaluations.

Acknowledgements

The NLLG group gratefully acknowledges support from the Federal Ministry of Education and Research (BMBF) via the research grant "Metrics4NLG" and the German Research Foundation (DFG) via the Heisenberg Grant EG 375/5-1.

Limitations

While our proposed approach demonstrates substantial reductions in token usage without sacrificing evaluation quality, there are several limitations to this study that we acknowledge.

First, our experiments are primarily focused on machine translation MT evaluation. Although we achieved notable efficiency gains in this domain, we have not extensively tested the applicability of **PromptOptMe** on other NLG tasks. Further research is necessary to validate the effectiveness of **PROMPTOPTME** across a broader spectrum of NLG applications.

Second, we evaluated our method using only one openly available language model in addition to **GPT-4o**, specifically the **LLaMA-3.2** model. While the results are promising, testing our approach on a wider variety of models, including more open-source and proprietary LLMs with diverse architectures and sizes, would help establish the generalizability and robustness of **PROMPTOPTME**. Future work should include experiments with additional models to better understand the applicability of our prompt optimization technique.

Additionally, our approach relies on a two-stage fine-tuning process involving supervised fine-tuning and preference optimization. This process requires access to sufficient training data and computational resources for fine-tuning the smaller language model used for prompt compression. In scenarios where such resources are limited or unavailable, the practicality of our method may be constrained. Exploring alternative approaches that require less extensive fine-tuning or that leverage zero-shot or few-shot learning could mitigate this limitation. In addition to one-time training cost, practitioners using PROMPTOPTME should also consider inference costs for the compression model itself. Further work should take these costs into account for more fair comparison with the baseline.

Furthermore, while our prompt optimization aims to preserve essential evaluation information, there is a possibility that some fine-grained details, such as specific MQM error categories, may not be fully captured in the compressed prompts. This could potentially affect the precision in identifying certain types of translation errors, leading to less detailed evaluations. Ensuring that critical information is retained during compression without increasing token usage remains a challenge that warrants further investigation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jonas Belouadi and Steffen Eger. 2023. UScore: An effective approach to fully unsupervised evaluation metrics for machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–374, Dubrovnik, Croatia. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

Yanran Chen and Steffen Eger. 2023. MENLI: Robust evaluation metrics from natural language inference. *Transactions of the Association for Computational Linguistics*, 11:804–825.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3201–3219, Bangkok, Thailand. Association for Computational Linguistics.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12914–12929, Singapore. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083, Singapore. Association for Computational Linguistics.

Markus Freitag, 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.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the*

Eighth Conference on Machine Translation, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *arXiv preprint arXiv:2310.10482*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Moussa Kamal Eddine, Guokan Shang, Antoine Tixier, and Michalis Vazirgiannis. 2022. FrugalScore: Learning cheaper, lighter and faster evaluation metrics for automatic text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1305–1318, Dublin, Ireland. Association for Computational Linguistics.

Marvin Kaster, Wei Zhao, and Steffen Eger. 2021. Global explainability of BERT-based evaluation metrics by disentangling along linguistic factors. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8912–8925, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Weize Kong, Spurthi Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. PRewrite: Prompt rewriting with reinforcement learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–601, Bangkok, Thailand. Association for Computational Linguistics.

Daniil Larionov, Jens Grünwald, Christoph Leiter, and Steffen Eger. 2023. EffEval: A comprehensive evaluation of efficiency for MT evaluation metrics. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 78–96, Singapore. Association for Computational Linguistics.

Daniil Larionov, Mikhail Seleznyov, Vasiliy Viskov, Alexander Panchenko, and Steffen Eger. 2024. xcomet-lite: Bridging the gap between efficiency and quality in learned mt evaluation metrics. *arXiv preprint arXiv:2406.14553*.

Christoph Leiter and Steffen Eger. 2024. Prexme! large scale prompt exploration of open source llms for machine translation and summarization evaluation. *ArXiv*, abs/2406.18528.

Christoph Leiter, Piyawat Lertvittayakumjorn, Marina Fomicheva, Wei Zhao, Yang Gao, and Steffen Eger. 2024. Towards explainable evaluation metrics for machine translation. *Journal of Machine Learning Research*, 25(75):1–49.

Christoph Leiter, Juri Opitz, Daniel Deutsch, Yang Gao, Rotem Dror, and Steffen Eger. 2023. The Eval4NLP 2023 shared task on prompting large language models as explainable metrics. In *Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems*, pages 117–138, Bali, Indonesia. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 963–981, Bangkok, Thailand and virtual meeting. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jan-Thorsten Peter, David Vilar, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, and Markus Freitag. 2023. There’s no data like better data: Using QE metrics for MT data filtering. In *Proceedings of the Eighth Conference on Machine Translation*, pages 561–577, Singapore. Association for Computational Linguistics.

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. 2022a. 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.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022b. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium. European Association for Machine Translation.

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, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

A Original and Simplified GEMBA-MQM prompt

B Additional Details

B.1 Package versions

We use the following versions of key software packages in our environment:

- Python OpenAI SDK: 1.42.0
- mt-metrics-eval: 0.0.3
- SciPy: 1.14.1
- LLMLingua: 0.2.2

B.2 Artifact Intended Use and License

To the best of our knoweledge, our use of scientific artifacts, namely Llama 3.2 pretrained model(s), is consistent with intended use policy and with the license. Those are outlined in the respective model card [2].

B.3 Total Computational Budget

Our total computational budget for supervised finetuning and preference optimization of PromptOptMe models is 186 GPU hours.

[2] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

```
================================ System Message ================================
You are an annotator for the quality of machine translation. Your task is to identify errors and assess the quality of the translation.
================================ Human Message ================================
English source:
“‘I do apologise about this, we must gain permission from the account holder to discuss an order with another person, I apologise if this was done previously, however, I would not be able to discuss this with yourself without the account holders permission.“‘
German translation:
“‘Ich entschuldige mich dafür, wir müssen die Erlaubnis einholen, um eine Bestellung mit einer anderen Person zu besprechen. Ich entschuldige mich, falls dies zuvor geschehen wäre, aber ohne die Erlaubnis des Kontoinhabers wäre ich nicht in der Lage, dies mit dir involvement.“‘

Based on the source segment and machine translation surrounded with triple backticks, identify error types in the translation and classify them. The categories of errors are: accuracy (addition, mistranslation, omission, untranslated text), fluency (character encoding, grammar, inconsistency, punctuation, register, spelling), style (awkward), terminology (inappropriate for context, inconsistent use), non-translation, other, or no-error. Each error is classified as one of three categories: critical, major, and minor. Critical errors inhibit comprehension of the text. Major errors disrupt the flow, but what the text is trying to say is still understandable. Minor errors are technically errors, but do not disrupt the flow or hinder comprehension.
================================== AI Message ==================================
Critical:
no-error
Major:
accuracy/mistranslation - "involvement"
accuracy/omission - "the account holder"
Minor:
fluency/grammar - "wäre"
fluency/register - "dir"
================================ Human Message ================================
English source:
“‘Talks have resumed in Vienna to try to revive the nuclear pact, with both sides trying to gauge the prospects of success after the latest exchanges in the stop-start negotiations.“‘
Czech translation:
“‘Ve Vídni se ve Vídni obnovily rozhovory o oživení jaderného paktu, přičemž obě partaje se snaží posoudit vyhlídky na úspěch po posledních výměnách v jednáních.“‘

Based on the source segment and machine translation surrounded with triple backticks, identify error types in the translation and classify them. The categories of errors are: accuracy (addition, mistranslation, omission, untranslated text), fluency (character encoding, grammar, inconsistency, punctuation, register, spelling), style (awkward), terminology (inappropriate for context, inconsistent use), non-translation, other, or no-error. Each error is classified as one of three categories: critical, major, and minor. Critical errors inhibit comprehension of the text. Major errors disrupt the flow, but what the text is trying to say is still understandable. Minor errors are technically errors, but do not disrupt the flow or hinder comprehension.
================================== AI Message ==================================
Critical:
no-error
Major:
accuracy/addition - "ve Vídni"
accuracy/omission - "the stop-start"
Minor:
terminology/inappropriate for context - "partaje"
================================ Human Message ================================
Chinese source:
“‘大众点评乌鲁木齐家居卖场频道为您提供高铁居然之家地址，电话，营业时间等最新商户信息，找装修公司，就上大众点评“‘
English translation:
“‘Urumqi Home Furnishing Store Channel provides you with the latest business information such as the address, telephone number, business hours, etc., of high-speed rail, and find a decoration company, and go to the reviews.“‘

Based on the source segment and machine translation surrounded with triple backticks, identify error types in the translation and classify them. The categories of errors are: accuracy (addition, mistranslation, omission, untranslated text), fluency (character encoding, grammar, inconsistency, punctuation, register, spelling), style (awkward), terminology (inappropriate for context, inconsistent use), non-translation, other, or no-error. Each error is classified as one of three categories: critical, major, and minor. Critical errors inhibit comprehension of the text. Major errors disrupt the flow, but what the text is trying to say is still understandable. Minor errors are technically errors, but do not disrupt the flow or hinder comprehension.
================================== AI Message ==================================
Critical:
accuracy/addition - "of high-speed rail"
Major:
accuracy/mistranslation - "go to the reviews"
Minor:
style/awkward - "etc.,"
================================ Human Message ================================
{source_lang} source:
“‘{source_seg}“‘
{target_lang} translation:
“‘{target_seg}“‘

Based on the source segment and machine translation surrounded with triple backticks, identify error types in the translation and classify them. The categories of errors are: accuracy (addition, mistranslation, omission, untranslated text), fluency (character encoding, grammar, inconsistency, punctuation, register, spelling), style (awkward), terminology (inappropriate for context, inconsistent use), non-translation, other, or no-error. Each error is classified as one of three categories: critical, major, and minor. Critical errors inhibit comprehension of the text. Major errors disrupt the flow, but what the text is trying to say is still understandable. Minor errors are technically errors, but do not disrupt the flow or hinder comprehension.
```

Figure 3: Original Prompt for GEMBA-MQM metric.

```
================================ System Message ================================
Identify and categorize translation errors. Respond in JSON.
================================ Human Message ================================
English: “‘I do apologise about this, we must gain permission from the account holder to discuss an order with another person, I apologise if this was done previously, however, I would not be able to discuss this with yourself without the account holders permission.“‘
German: “‘Ich entschuldige mich dafür, wir müssen die Erlaubnis einholen, um eine Bestellung mit einer anderen Person zu besprechen. Ich entschuldige mich, falls dies zuvor geschehen wäre, aber ohne die Erlaubnis des Kontoinhabers wäre ich nicht in der Lage, dies mit dir involvement.“‘;
Errors?
================================= AI Message =================================
"critical": ["no-error"], "major": ["accuracy/mistranslation": "involvement", "accuracy/omission": "the account holder"], "minor": ["fluency/grammar": "wäre", "fluency/register": "dir"]
================================ Human Message ================================
English: “‘Talks have resumed in Vienna to try to revive the nuclear pact, with both sides trying to gauge the prospects of success after the latest exchanges in the stop-start negotiations.“‘
Czech: “‘Ve Vídni se ve Vídni obnovily rozhovory o oživení jaderného paktu, přičemž obě partaje se snaží posoudit vyhlídky na úspěch po posledních výměnách v jednáních.“‘;
Errors?
================================= AI Message =================================
"critical": ["no-error"], "major": ["accuracy/addition": "ve Vídni", "accuracy/omission": "the stop-start"], "minor": ["terminology/inappropriate for context": "partaje"]
================================ Human Message ================================
Chinese: “‘大众点评乌鲁木齐家居卖场频道为您提供高铁居然之家地址，电话，营业时间等最新商户信息，找装修公司，就上大众点评“‘
English: “‘Urumqi Home Furnishing Store Channel provides you with the latest business information such as the address, telephone number, business hours, etc., of high-speed rail, and find a decoration company, and go to the reviews.“‘;
Errors?
================================= AI Message =================================
"critical": ["accuracy/addition": "of high-speed rail"], "major": ["accuracy/mistranslation": "go to the reviews"], "minor": ["style/awkward": "etc.,"]
================================ Human Message ================================
{source_lang}: “‘{source_seg}“‘
{target_lang}: “‘{target_seg}“‘
Errors?
```

Figure 4: Simplifield Prompt for GEMBA-MQM metric.