The Devil is in the Errors: Leveraging Large Language Models for Fine-grained Machine Translation Evaluation

Patrick Fernandes[*,2,3,4] **Daniel Deutsch**[1] **Mara Finkelstein**[1] **Parker Riley**[1]
André F. T. Martins[3,4,5] **Graham Neubig**[2,6]
Ankush Garg[1] **Jonathan H. Clark**[1] **Markus Freitag**[1] **Orhan Firat**[1]
[1]Google [2]Carnegie Mellon University [3]Instituto Superior Técnico
[4]Instituto de Telecomunicações [5]Unbabel [6]Inspired Cognition
pfernand@cs.cmu.edu

Abstract

Automatic evaluation of machine translation (MT) is a critical tool driving the rapid iterative development of MT systems. While considerable progress has been made on estimating a single scalar quality score, current metrics lack the informativeness of more detailed schemes that annotate individual errors, such as Multidimensional Quality Metrics (MQM). In this paper, we help fill this gap by proposing **AUTOMQM**, a prompting technique which leverages the *reasoning* and *in-context learning* capabilities of large language models (LLMs) and asks them to identify and categorize errors in translations. We start by evaluating recent LLMs, such as PaLM and PaLM-2, through simple *score prediction* prompting, and we study the impact of labeled data through in-context learning and finetuning. We then evaluate AUTOMQM with PaLM-2 models, and we find that it improves performance compared to just prompting for scores (with particularly large gains for larger models) while providing interpretability through error spans that align with human annotations.

1 Introduction

Evaluating natural language generation systems has always been challenging, and as the output quality of these systems has improved, evaluation has become even more challenging and critical. For example, in Machine Translation (MT), a field where evaluation has garnered considerable attention, previous standard automatic surface-level metrics such as BLEU (Papineni et al., 2002) are becoming less reliable as the quality of generation systems improves, with little remaining correlation with human judgments (Freitag et al., 2022).

To keep pace with the constantly improving quality of MT output, the next generation of automatic metrics is rapidly evolving. *Learned* automatic metrics that leverage human-judgments to finetune

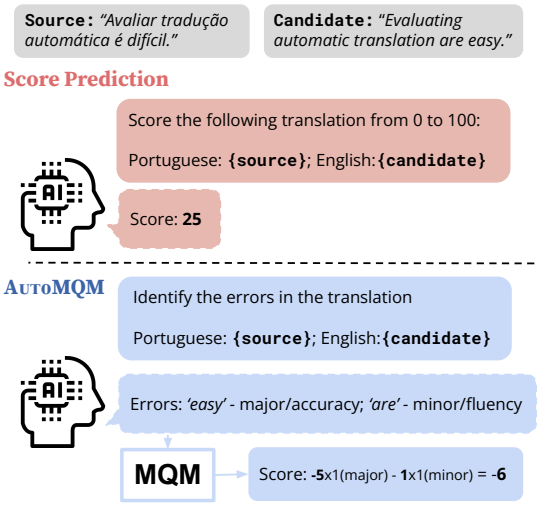

Figure 1: Illustration of how AUTOMQM uses LLMs to assess the quality of a translation. Rather than asking for a single quality score, AUTOMQM prompts models to identify and classify errors, and uses the MQM framework to produce a score.

language models (Sellam et al., 2020; Rei et al., 2022a) currently represent the state-of-the-art in automatic evaluation benchmarks like the WMT Metrics task (Freitag et al., 2022), and show high correlation with human judgments. However, these metrics typically output a single, *uninterpretable* quality score, making it difficult to understand the type and extent of errors identified by them. The lack of insights makes it difficult for model developers to leverage these metrics to improve their systems.

Unlike automatic metrics that only provide a single scalar value as quality score, state-of-the-art human evaluation methodologies like Multidimensional Quality Metrics (MQM; Lommel et al., 2014; Freitag et al., 2021a) ask professional annotators to identify and label error spans with a category and severity. This much richer feedback can be used to gain a better understanding of the current limitations of the model under evaluation and improve it.

In this paper, we ask whether large language

[*] Work done while working part-time at Google.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 1066–1083
December 6–7, 2023.

models (LLMs) in combination with a few human annotations can be used to design an automatic metric that generates rich feedback similar to that generated by human experts in MQM. This work is motivated by recent papers that demonstrated that LLMs can be used as automatic metrics (Liu et al., 2023b) to generate a single quality score. In particular, Kocmi and Federmann (2023) showed that LLMs can be prompted to assess the quality of machine-generated translations, even achieving state-of-the-art performance on assessing system-level quality. However, previous work only provides a limited view of the capabilities of LLMs for machine translation evaluation: the focus has predominantly been on *score prediction* (i.e. predicting a numerical value for quality), without considering the use of *any* annotated data (either through in-context learning or finetuning), and only in *high-resource* language pairs.

We provide a large-scale study of the capabilities of LLMs (from the PaLM and PaLM-2 families; Chowdhery et al., 2022; Anil et al., 2023) for machine translation evaluation (both with and without a reference translation), provide a novel comparison between prompting and finetuning, and investigate the performance in the low-resource scenario. Inspired by findings that the performance of LLMs can be improved by prompting them for *rationales* of their predictions (Wei et al., 2022; Lu et al., 2023), we also propose **AUTOMQM**, a prompting technique for MT evaluation that asks LLMs to identify error spans in a translation and to classify these errors according to the MQM framework, with a quality score derived automatically from the identified errors. A key advantage of AUTOMQM is its *interpretability*, as users can inspect the errors responsible for a score (Figure 1).

Our contributions can be summarized as follows:

- We confirm the finding of Kocmi and Federmann (2023) that LLMs are *zero-shot* state-of-the-art system-level evaluators, but show low correlation with human judgment compared to *learned* metrics at the segment-level.
- We show that *finetuning* an LLM with human judgment mitigates its low segment-level performance (particularly for smaller LLMs), showing similar correlations with human judgment at both the system-level and segment-level to state-of-the-art learned metrics.
- We are the first to evaluate LLM-based evaluation methods on low-resource language pairs. We find that their performance is promising, but lags behind state-of-the-art learned metrics.
- We find that, with AUTOMQM, PaLM-2 models can be prompted to generate rich MQM-like annotations, outperforming their score prediction counterparts at the segment-level.
- Furthermore, annotations predicted by PaLM-2 models correctly identify over 50% of words that are part of *major* errors, and are comparable to the ones produced by state-of-the-art *supervised* word-level evaluators.

Our findings might have significant implications for not only MT evaluation, but evaluation of machine-generated text in general, and further highlight the potential of using LLMs to provide *AI Feedback* (Fernandes et al., 2023).

The outputs of our models prompted with AUTOMQM are available at github.com/google-research/google-research

2 Background: MT Evaluation

Machine translation evaluation is one of the most well-studied evaluation problems in NLP (Callison-Burch et al., 2008; Freitag et al., 2022). In this task, given

1. a *source* sentence in a (source) language
2. a *candidate* translation in a (target) language

an evaluation metric assesses the quality of the candidate translation by how well it conveys the meaning of the source sentence while considering other factors like *fluency*. Like many other natural language generation evaluation problems, this task is difficult because the set of correct translations for a given source sentence is often very large and not entirely known in advance. To simplify the problem of machine translation evaluation, often (3) a *reference* translation (typically created by a professional human translator) is included as additional information when assessing the candidate translation. This sub-problem is known as *reference-based* evaluation (as opposed *reference-less* evaluation or *quality estimation*).

Up until recently, human evaluation of machine translation was carried out predominantly with the aim of assigning a single quality score to a candidate translation. Consequently, *learned* metrics, which leverage collected human judgment data, are trained for and evaluated on the same task of *score*

prediction (i.e., assigning a single quality score to a candidate translation), and can achieve high correlation with human-provided scores (Freitag et al., 2022).

However, framing machine translation evaluation as a score prediction task is problematic: any scoring or ranking of translations is implicitly based on an identification of errors in the candidate translations, and asking raters to solely provide a single score can lead to rushed and noisy judgments (Freitag et al., 2021a).

This insight has led to the adoption of the Multidimensional Quality Metrics (MQM) framework (Lommel et al., 2014; Freitag et al., 2021a) as the gold standard for evaluating machine translation. The MQM framework asks human evaluators to identify error spans in candidate translations and classify those errors according to various dimensions, e.g., *fluency*, *accuracy*, ... (see Appendix A for a more detailed description of MQM). Importantly, the MQM framework *does not* ask annotators to provide a quality score for each translation, and instead derives one automatically from the identified error spans and their classifications. However, despite its richness, *most* automatic metrics that leverage MQM data only use the final quality score produced by the framework and discard the error span information and classification.

3 Related Work

The success of *learned* machine translation metrics (Sellam et al., 2020; Rei et al., 2022a; Freitag et al., 2022; Qin et al., 2022), which finetune neural network models pretrained on large amounts of (unsupervised) data, highlighted the importance of leveraging *transfer learning* to achieve metrics with better correlation with human judgments. More recently, *generative* LLMs (OpenAI, 2023; Anil et al., 2023) have consistently demonstrated impressive results in natural language understanding and *zero-* and *few-shot* transfer and, naturally, interest in employing these models for (translation) evaluation has increased. Kocmi and Federmann (2023) first explored the use of GPT models for evaluating machine translation tasks, showing their potential as *zero-shot* evaluators, and others have since extended GPT-based evaluation to other generation problems (Jain et al., 2023; Liu et al., 2023b).

Perrella et al. (2022) first highlighted that MQM annotations could be leveraged to allow pretrained models to predict major and minor errors and, similarly to AUTOMQM, used the identified errors to automatically score translations. However, their approach relied on weaker encoder-only or encoder-decoder language models, required *supervised* data to work, and overall underperformed other top metrics. We compare against their *MaTASe* metric in our experiments. Lu et al. (2023) showed that doing *error analysis*, a prompting technique similar to AUTOMQM, could lead to better ChatGPT-based evaluators. However, they still relied on the LLM to provide a score once it identified errors (rather than do it automatically using something like the MQM framework). Furthermore, they provided a very limited meta-evaluation using only 40 examples per language pair. Concurrently with our work, Xu et al. (2023) proposed INSTRUCTSCORE, a LLaMA-based evaluator that asks models to identify and categorize errors in translation (as well as providing a natural language explanation for each error). However, the authors only explore a 7B parameter model and don't leverage zero- and few-shot capabilities of models as in this work. Instead, they rely on a more complex approach of distilling the knowledge of a more capable GPT-4 LLM.

Additionally, WMT Word-Level Quality Estimation shared tasks (Fonseca et al., 2019; Zerva et al., 2022) leverage MQM data by converting span-level annotations of errors (normally of *major* severity) to word-level tags and Task 2 in the WMT19 Quality Estimation shared task evaluation explicitly evaluated submissions of span-level annotations (although most submissions still consisted of models that predicted word-level tags which were converted to spans). We also compare against state-of-the-art word-level quality estimation models.

4 Using LLMs to Predict Quality Scores

Recent works have shown that large language models are versatile, general-purpose models that can be used to tackle many problems in NLP, including evaluation (Kocmi and Federmann, 2023; Jain et al., 2023; Liu et al., 2023b). We begin by exploring how LLMs can be used for machine translation evaluation through *score prediction*.

4.1 Prompting

We start by measuring how far we can push the performance of LLMs with just *prompting* (Liu et al., 2023a): by defining the task of MT evaluation and quality estimation as *textual templates* (with

a general description of the problem and "slots" for the inputs and outputs), we can use general-purpose LLMs to perform these tasks at inference-time, without any parameter updates.

Throughout the paper, we choose to use Kocmi and Federmann (2023)'s `GEMBA-SQM` prompt (Figure 9, Appendix C), which asks models to generate (a string representation of) a score from 0-100. We choose this prompt for two reasons: firstly, early explorations with various prompts showed that this generally performed well. Secondly, using a single prompt ensures a fairer comparison between the capabilities of different models.[1]

In-Context Learning A surprising emergent capability of LLMs is their ability to improve on prompting-based tasks by including a very small amount of labeled data as part of the prompt/context (Brown et al., 2020) and *without* parameter updates, a technique called *in-context learning* (ICL) or *few-shot prompting*. We thus investigate the impact that ICL has on LLMs' ability to assess translation quality. Recent works have shown that the impact of ICL is tightly tied with the exact examples included in the prompt, with a poor selection procedure leading to no improvements or even worse performance than the zero-shot case (Jain et al., 2023). We therefore explore two sampling approaches to select in-context examples from a pre-defined "pool" of translation quality assessments: **uniform** and **stratified sampling**, where the example pool is bucketed by score ranges and examples are sampled from each bucket.

4.2 Finetuning

It has previously been shown that LLMs are capable of zero-shot evaluation (Kocmi and Federmann, 2023), but the extent to which *finetuning* on human judgment data can further boost the performance of LLMs has not been studied. In the WMT'22 Metrics Shared Task (Freitag et al., 2022), all top submissions were learned metrics; that is, pretrained models finetuned on human judgment data[2].

Thus, we investigate whether LLMs are amenable to finetuning on human judgment data. LLMs used in top-performing metrics are generally much larger than the pretrained language models leveraged by previous learned metrics (which generally have fewer than 1 billion parameters). Moreover, most learned metrics leverage pretrained encoder-only rather than (decoder-only) prefix language models. We experiment with finetuning LLMs using two objectives:

- *Regression* (**R**): Commonly used for training learned metrics (Rei et al., 2022a), the objective here is a regression loss (e.g., mean squared error) between continuous scores obtained from the model (for example, with a *regression head*) and the human scores.
- *Generative Classification* (**GC**): We bucket scores into discrete classes (e.g. *"bad"*, *"ok"* and *"good"*) and treat the MT evaluation task as a text-to-text classification problem (Raffel et al., 2020) by having the model generate a template sentence with the class. See §6.1 for more details.

5 Using LLMs to Predict Error Spans

While producing quality scores that correlate with human judgments is an important part of translation quality assessment, metrics that solely do score prediction suffer from problems of **interpretability**: if a metric assigns a low score, the downstream users are left in the dark about which parts of the translation were responsible for the score and thus need to be corrected. This is especially problematic in cases where the metric assigns a *wrong* score to a translation, as it is much harder to diagnose why the evaluation model made a mistake, and identify and prevent similar mistakes in the future. In fact, reducing translation quality to a single score has proven problematic even for human annotators: asking raters to solely provide a single score can lead to rushed and noisy judgments (Freitag et al., 2021a) and the current gold standard for translation quality evaluation involving human annotators is instead based on methodologies like the MQM framework (see §2) , which provide richer feedback by identifying error spans, categorizing them, and evaluating their severity.

Interestingly, another emergent phenomenon in LLMs is the success of *chain-of-thought* prompting (Wei et al., 2022): when defining a prompt for a particular task, if we instruct the model to produce a series of intermediate reasoning steps (*"let's think step-by-step"*), it tends to generate a free-text *rationale* before generating an output, and this often improves the performance on the

[1] While this prompt wasn't the best for *system-level*, it led to the best *segment-level* performance in GEMBA.

[2] While these metrics all leverage powerful pretrained (language) models, these generally aren't considered LLMs

```
Based on the given source and reference, identify the major and minor errors in this
translation. Note that Major errors refer to actual translation or grammatical errors,
and Minor errors refer to smaller imperfections, and purely subjective opinions about
the translation.

{src_lang} source: "{source}"
{tgt_lang} human reference: "{reference}"
{tgt_lang} translation: "{candidate}"
Errors: {error1:span} - {error1:severity}/{error1:category}; {error2:span} - ...
```

Figure 2: The AUTOMQM prompt used in this paper. Parts in purple are only included for *reference-based* evaluation, while parts in orange represent slots for outputs, and are only included for in-context examples.

task at hand (Liu et al., 2023b). Furthermore, this *chain-of-thought* prompting can be used to obtain *structured* rationales from LLMs, and this can lead to better performance than with free-text rationales (Lu et al., 2023).

Motivated by these findings, we propose **AUTOMQM**, a prompting technique for translation quality assessment that instructs LLMs to *identify* errors in a translation, and *categorize* the type of error according to the MQM framework (Lommel et al., 2014). Furthermore, we *don't* ask the model to produce a score, as the MQM framework provides an algorithmic procedure to obtain one from identified errors: the total score is the sum of penalties for all errors identified, where (roughly) *major* errors get penalized with -5 and *minors* with -1 (see Appendix A for a more detailed description of the scoring algorithm).[3] Figure 2 shows the main AUTOMQM prompt used in this paper.

Importantly, obtaining meaningful AUTOMQM results in a zero-shot setting is a substantially more challenging task compared to score prediction: we found that, without any in-context examples, LLMs tend to produce outputs that are either uninformative or difficult to parse. Thus we only consider the AUTOMQM task in the *few-shot* scenario. Based on the findings from §6.2, we explore the impact of in-context learning by sampling from the example pool using stratified sampling extended with a set of *rejection criteria* (Appendix D), which ensures that the example set has a balance between major and minor errors as well as diversity in the categories of errors.

6 Experiments

6.1 Experimental Setup

Data The metrics in this work are evaluated on both *high-resource* and *low-resource* language pairs. The three high-resource language pairs come from the WMT'22 Metrics Shared Task (Freitag et al., 2022): en→de, zh→en, and en→ru. The ground-truth translation quality scores are derived from MQM ratings in which expert annotators marked error spans in the translations with different severity levels which are automatically converted to a numeric score (see §2). The four low-resource language pairs come from the WMT'19 Metrics Shared Task (Ma et al., 2019): en↔gu and en↔kk. Since MQM ratings are not available for the low-resource pairs, the ground truth quality scores are direct assessment (DA) scores. DA scores are quality assessments assigned by non-expert raters on a scale from 0-100, normalized per rater. See Table 9 (Appendix B) for statistics about the number of MT systems and segments for every language pair.

Additionally, in our experiments, AUTOMQM required in-context examples with MQM annotations to work, so we restrict our evaluation of AUTOMQM to en→de and zh→en because there are available MQM ratings from the WMT'21 Metrics Shared Task (Freitag et al., 2021b) that we can use as in-context learning example pools.

Models We base most of our experiments on the following LLMs:

- **PaLM**: A 540 billion parameter autoregressive Transformer model trained on 780 billion tokens of high-quality text (Chowdhery et al., 2022). It showed remarkable performance on a wide-range of NLP tasks, including Machine Translation (Vilar et al., 2022).
- **PaLM-2**: The successor to PaLM, the PaLM-2 family of LLMs (Anil et al., 2023) builds upon recent research insights, such as compute-optimal scaling, a more multilingual and diverse pre-training mixture, and architectural/optimization improvements. We mainly use two model sizes in the family: PaLM-2 BI-

[3] This is similar to methods that leverage external *executors* to improve the performance of LLMs (Gao et al., 2022)

SON and (the larger) PaLM-2-UNICORN.[4] In addition we explore the impact of instruction-tuning by using a UNICORN model finetuned on the FLAN dataset (Wei et al., 2021).

For *score prediction*, we compare PaLM and PaLM-2 against the GPT family of LLMs (Brown et al., 2020; OpenAI, 2023) by leveraging the results and outputs from the GEMBA evaluator (Kocmi and Federmann, 2023). We then evaluate the performance of AUTOMQM with only PaLM-2 models (which performed best in score prediction).

Additionally, for the high-resource languages, we compare to a set of strong baseline evaluation metrics, MetricX-XXL and COMET-22, which were the two top-performing metrics in the WMT'22 Metrics Shared Task. MetricX-XXL and COMET-22 are both finetuned regression models trained on DA data from WMT that are initialized with mT5 (Xue et al., 2021) and XLM-R (Conneau et al., 2020), respectively.

For the AUTOMQM experiments, we also compare against MATESE, a comparable submission to the WMT'22 Metrics Shared task that finetuned a XLM-R model to identify major and minor errors, and computed a score automatically. Since we were unable to obtain the span-level predictions for the MATESE submission, we also compare against the top submission to the WMT'22 Word-Level Quality Estimation Shared Task (Zerva et al., 2021): word-level COMETKIWI (COMET-WL) (Rei et al., 2022b), also based on an XLM-R model trained on a combination of sentence- and word-level data. To do so, we re-run this model on the WMT'22 Metrics Shared Task data, and convert the predicted *word-level* `OK/BAD` tags into spans.[5]

Finetuning For *regression* finetuning, we use a real-valued logit, extracted from a fixed index in the first target token's logit vector, as the quality signal. (In particular, we leverage a special, *unused*, vocabulary token.) This was the technique used to train MetricX-XXL in the WMT 2022 Shared Task submission (Freitag et al., 2022). The regression-based model was trained on WMT direct assessment (DA) data from the years 2015 through 2020.

For *generative* classification, we bucket the scores in the training data into five classes, where class boundaries are assigned so that each class contains an equal number of training examples. We then map labels to verbal ratings from the following set, based on their bucket: [*"very bad"*, *"bad"*, *"ok"*, *"good"*, *"very good"*]. To evaluate the model, predictions are mapped back to integer labels from 1 to 5. Any predictions not containing a substring in the label set are considered invalid and are mapped to 0. We experimented with finetuning on both DA and MQM 2020 (Freitag et al., 2021a) data, and found that the latter performed slightly better.

To assess the impact of *model size*, we also finetune two additional (smaller) PaLM-2 models, which we call S and M, comparing their finetuned and zero-shot performance.[6]

Metric Meta-Evaluation The quality of an automatic evaluation metric is estimated by comparing the agreement between the metric scores and ground-truth quality scores on a large number of translations from different MT systems, a process known as metric meta-evaluation. This work reports three different agreement scores, as follows.

The first is system-level accuracy, which calculates the percent of system pairs that are ranked the same by the metric and ground-truth scores, micro-averaged over a set of language pairs (Kocmi et al., 2021). System-level scores are defined as the average score across all segments.

At the segment-level, the standard correlation that is reported by WMT is Kendall's τ. However, recent work pointed out problems with Kendall's τ with respect to ties (Deutsch et al., 2023). In short, different variants of τ are inconsistent with respect to ties and even biased against metrics that predict ties, as our metrics do in this work. Deutsch et al. (2023) recommend reporting a pairwise accuracy score, which rewards metrics for correctly ranking translations as well as correctly predicting ties, in combination with a tie calibration procedure that automatically introduces ties into metric scores so that the meta-evaluation is fairer. This accuracy score, denoted acc*, ranges between 0 and 1, and a random metric would achieve 33% accuracy. We report the "group-by-item" variant of the pairwise accuracy score from Deutsch et al. (2023) in addition to Pearson's ρ, a complementary signal to rank-based correlations that measure the strength of the linear relationship between two variables (and one of the standard correlations reported in WMT).

[4] Information about exact number of parameters of PaLM-2 models is not publicly available.

[5] We consider a span as any maximal consecutive sequence of words marked as `BAD`, assigning every span the *major* severity.

[6] We use a small variation of the *zero-shot* prompt, asking models for scores from the same 5 buckets used in finetuning.

		System-Level	Segment-Level					
		All (3 LPs)	EN-DE		ZH-EN		EN-RU	
Model	Ref?	Accuracy	ρ	acc*	ρ	acc*	ρ	acc*
Baselines								
MetricX-XXL	✓	85.0%	0.549	61.1%	0.581	54.6%	0.495	60.6%
COMET-22	✓	83.9%	0.512	60.2%	0.585	54.1%	0.469	57.7%
COMET-QE	✗	78.1%	0.419	56.3%	0.505	48.8%	0.439	53.4%
Prompting								
PaLM 540B	✓	90.1%	0.247	55.4%	0.255	48.5%	0.180	48.6%
PaLM-2 BISON	✓	88.7%	0.394	56.8%	0.322	49.3%	0.322	52.8%
PaLM-2 UNICORN	✓	90.1%	0.401	56.3%	0.349	51.1%	0.352	55.3%
FLAN-PaLM-2 UNICORN	✓	75.9%	0.197	55.6%	0.139	46.1%	0.198	52.0%
PaLM 540B	✗	84.3%	0.239	56.1%	0.270	43.1%	0.300	51.8%
PaLM-2 BISON	✗	85.0%	0.355	57.0%	0.299	48.6%	0.303	53.1%
PaLM-2 UNICORN	✗	84.3%	0.275	56.1%	0.252	48.3%	0.209	49.8%
FLAN-PaLM-2 UNICORN	✗	69.7%	0.116	54.6%	0.112	43.8%	0.156	47.8%
Finetune								
PaLM-2 BISON (**R**)	✓	88.0%	0.511	61.0%	0.459	51.5%	0.458	59.5%
PaLM-2 BISON (**GC**)	✓	86.1%	0.400	59.2%	0.444	49.3%	0.365	56.0%
PaLM-2 UNICORN (**R**)	✓	87.6%	0.508	61.1%	0.412	52.6%	0.460	60.4%
PaLM 2 BISON (**R**)	✗	87.6%	0.490	59.9%	0.439	53.4%	0.437	59.2%
PaLM 2 BISON (**GC**)	✗	86.1%	0.368	57.5%	0.420	47.3%	0.390	54.9%
PaLM 2 UNICORN (**GC**)	✗	86.1%	0.407	57.9%	0.402	45.6%	0.411	55.3%

Table 1: Meta-evaluation results at system and segment-level for the *high-resource* language pairs. Finetuned **(R)** and **(GC)** represent the *regression* and *generative classification* objectives (§4.2). ✓and ✗ represent *reference-based* and *reference-less* metrics, respectively.

Span Meta-Evaluation Since AUTOMQM provides not only scores but also the identified error spans, we can compare the predicted spans with the errors marked by annotators in the MQM annotations. We evaluate quality of predicted spans using: (1) *Span Precision* (SP), which measures the overlap of predicted spans and gold (annotated) spans; and (2) *Major recall* (MR), which captures the percentage of gold major errors that were predicted as errors (either minor or major).

More formally, consider the set of ground truth spans $S^\star$, where each span consists of a sequence of words, i.e., $s_i = (w_{(a)}, w_{(a+1)}, \cdots)$. Let $S^\star_{\text{maj}} \subseteq S^\star$ be the subset containing only the major errors. Given a span set S, we define its positional set $P(S)$ as the set containing the positions of all the words in every span in S. For example, assuming a span $s_i = (w_{(n)}, w_{(n+1)}, \cdots)$ in S starts at the nth position in the text, its corresponding positional set will include the positions $\{n, n{+}1, ..., n{+}\text{len}(s_i){-}1\}$. Then for a set of *predicted* spans $\hat{S}$, SP and MR are defined as:

$$\text{SP}(\hat{S}) = \frac{|P(\hat{S}) \cap P(S^\star)|}{|P(\hat{S})|} \quad (1)$$

$$\text{MR}(\hat{S}) = \frac{|P(\hat{S}) \cap P(S^\star_{\text{maj}})|}{|P(S^\star_{\text{maj}})|} \quad (2)$$

Intuitively, we care for overall precision (regardless of severity) since we want to make sure predicted errors tend to be marked by annotators as well, but for recall we care mostly for *major* errors, as these have a larger impact on translation quality and are more critical to identify. Additionally, we also report the (3) *Matthews Correlation Coefficient* (MCC), one of the official metrics in the word-level quality estimation tasks (Zerva et al., 2022).

6.2 Results

6.2.1 Score Prediction

Table 1 summarizes the meta-evaluation results, at the *system* and *segment* level, for both the *zero-shot prompting* and *finetuning* settings.

Prompting A first observation is almost all zero-shot LLM evaluators have higher *system-level* performance than learned metrics (with and without references), with PaLM 540B and PaLM-2 UNICORN achieving the best performance. At the segment level, the story is more complicated: similarly to Kocmi et al. (2022), we find that none of the LLMs we explored was able to consistently outperform the baseline learned metrics. We see that PaLM-540B is a particularly poor reference-based evaluator, which is surprising given its system-level performance. Unexpectedly, instruction-tuning with FLAN seems to *degrade* performance, with FLAN-PaLM-2 UNICORN achieving poor performance at both the system and segment levels.[7]

Nevertheless, PaLM-2 models achieve high correlations with human judgments, and the *reference-*

[7]Note that this might be a problem with the FLAN dataset and not instruction-tuning in general, as the GPT models are also instruction-tuned and perform well.

		System	Segment acc*		
Model	Ref?	All	EN-DE	ZH-EN	EN-RU
GEMBA					
GPT-3.5	✓	85.4%	54.9%	49.5%	47.5%
GPT-4	✓	88.7%	57.8%	52.6%	55.0%
GPT-3.5	✗	82.5%	56.1%	49.7%	49.3%
GPT-4	✗	89.1%	56.4%	53.4%	54.8%
BISON	✓	88.7%	56.8%	49.3%	52.8%
UNICORN	✓	90.1%	56.3%	51.1%	55.3%
BISON	✗	85.0%	57.0%	48.6%	53.1%
UNICORN	✗	84.3%	56.1%	48.3%	49.8%

Table 2: Comparison between PaLM-2 and GPT-based GEMBA (Kocmi et al., 2022) at the system and segment levels for the *high-resource* language pairs.

less PaLM-2 BISON is competitive with the *learned* baselines, particularly at assessing alternative translations of the same sentence (acc*). When comparing PaLM-2 models with Kocmi et al. (2022)'s GPT-based GEMBA evaluator (Table 2), we see that both families of LLMs perform similarly, with PaLM-2 models exhibiting higher system-level performance than GPT-based GEMBA, while GEMBA achieves better segment-level accuracy, particularly in the reference-less setting.

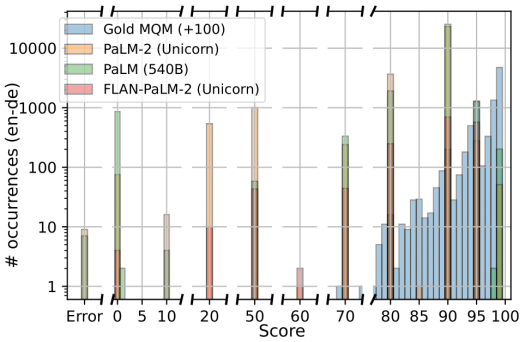

Figure 3: Distribution of scores for various LLM *reference-based* evaluators, on the EN-DE test set. Note that the y axis is in *log-scale*.

Figure 3 shows the distribution of scores produced by PaLM- and PaLM-2-based evaluators. We find that, despite being prompted to give a score in the 0-100 range, these models almost always output one of a very limited set of scores (e.g. 0, 50, 90, 95). Given Kocmi and Federmann (2023)'s similar findings with GPT models, it seems that this is a consequence of the pretraining objective.

Finetuning Despite their already-great performance in the zero-shot setting, we find that finetuning LLMs can further improve LLM evaluators' segment-level scores. This is particularly obvious for the *reference-less* evaluators, where a finetuned PaLM-2 BISON achieves state-of-the-art performance in segment-level correlations and comparable system-level accuracy across all language pairs. Moreover, when we look at how performance *scales* with parameter count (Figure 4), we observe an interesting trend: while smaller models are not capable of being effective zero-shot evaluators, finetuning them leads to competitive performance, and only a slight decrease when compared to their larger finetuned counterparts.

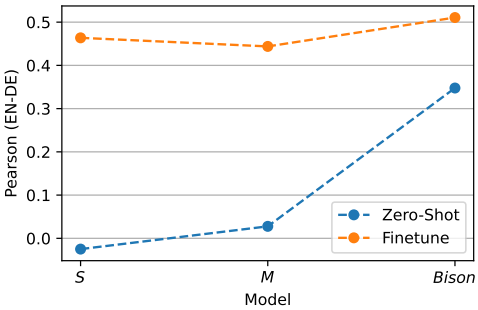

Figure 4: Behavior of *Pearson* as we scale the LLM's parameter count. Note that the x axis is not to-scale with regard to parameter count.

In-context Learning Figure 5 shows the mean and interquartile range (IQR) of the performance as we increase the number of in-context examples k (with 100 example sets per k) sampled with *stratified* sampling (see Appendix E for *uniform*). Surprisingly, despite evidence of the benefits of in-context learning for many tasks, we found that including in-context examples during evaluation (almost) never led to better performance, either with *uniform* or *stratified* sampling.

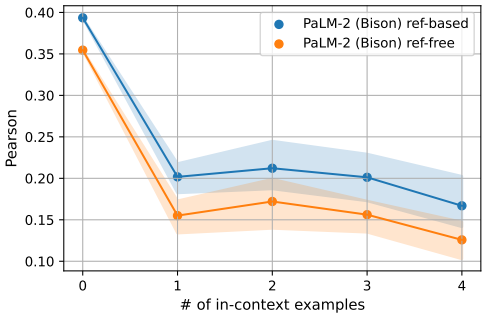

Figure 5: Mean *Pearson* and its interquartile range (IQR) in the WMT22 EN-DE test set, as we increase the number of in-context examples with *stratified* sampling

To investigate the cause of this disappointing performance, we looked at how *particular* in-context example sets affect the distribution of scores produced by LLM-based evaluators. Figure 6 shows the distribution of scores *over the whole test set* for the 1-shot and 2-shot settings, with different in-context examples sets. We can see that output distribution is heavily biased by the scores in the in-context examples: despite *never* predicting 79

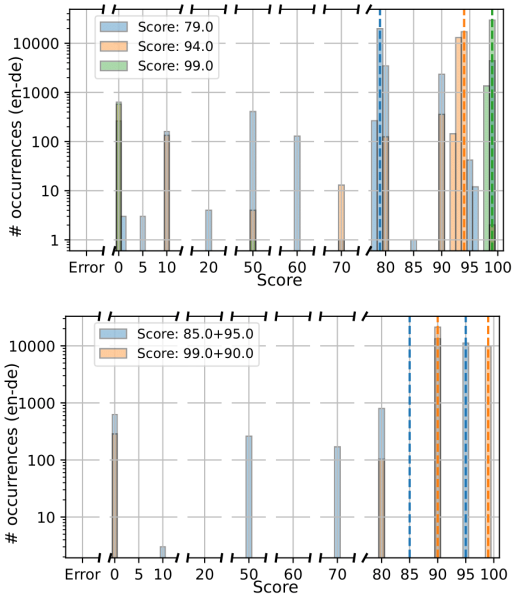

Figure 6: Distribution of scores for PaLM-2 (BISON) models for 1-shot (top) and 2-shot (bottom) setups, with various in-context learning sets for each (and their scores in the legend)

in the zero-shot setting, when a single example with that score is included, it starts to dominate the model predictions. This seems to hint that LLMs "overfit" to the specific scores provided as examples, rather than generalizing to the broader evaluation task, which could explain the lackluster performance of in-context learning.

6.3 Low Resource Languages

Table 3 shows the performance of PaLM-2 models at *score prediction* for *low-resource* translation. Overall, we find that similar to high-resource LPs, these models are good zero-shot evaluators, with system-level accuracies around 90%. However, *zero-shot* LLMs underperform *learned* metrics, even when these metrics also weren't exposed to data in these low-resource languages.

		System	Segment ρ			
Model	Ref?	All	EN-KK	EN-GU	KK-EN	GU-EN
Baseline						
MetricX-XXL*	✓	94.0%	0.666	0.701	0.539	0.409
Prompting						
BISON	✓	92.2%	0.605	0.540	0.462	0.339
UNICORN	✓	87.4%	0.609	0.621	0.495	0.384
BISON	✗	89.8%	0.567	0.478	0.381	0.313
UNICORN	✗	84.4%	0.536	0.523	0.433	0.334

Table 3: Meta-evaluation results for system-level *accuracy* and segment-level *Pearson* on the low-resource languages, using PaLM-2 for *score prediction*. *Note that the baseline is slightly different from the high-resource case, being trained on the same data but *without* these *low-resource* language pairs.

6.3.1 AUTOMQM

Figure 14 shows the mean and interquartile range (IQR) of the performance of PaLM-2 BISON with AUTOMQM, as we increase the number of in-context examples (again, with 100 example sets per k). Contrary to the performance with score prediction, we find that performance with AUTOMQM seems to (mostly) scale with the number of in-context examples: performance increases monotonically with up to 4 in-context examples and plateaus thereafter. Additionally, the variance across the in-context learning sets seems to be lower, with most example sets exhibiting less than 0.05 *Pearson* difference from the best-performing sets. All this suggests that LLM evaluators are much more robust to the choice of in-context examples when prompted for AUTOMQM rather than for score prediction. We also find that the behavior of in-context learning is quite similar for both reference-based and reference-less evaluation tasks. Finally, we observe that the example sets that perform well for one task generally work well for the other, with performance on both settings given a fixed in-context set being highly correlated, as shown in Figure 7.

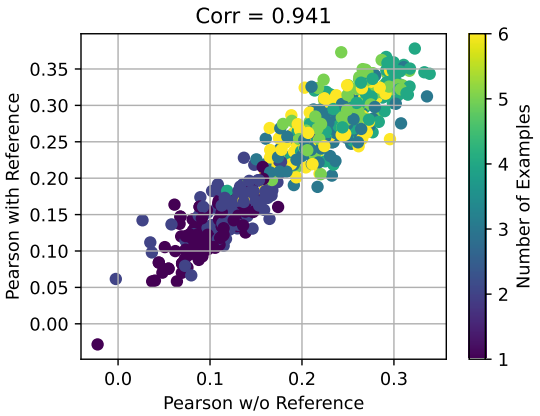

Figure 7: Scatter plot of the *Pearson* of PaLM-2 (BISON) models, with/without including the *reference* in the prompt, for each in-context learning setting tried.

Table 4 shows the meta-evaluation results for PaLM-2 BISON and UNICORN prompted with AUTOMQM (using the best-performing in-context learning sets in Figure 14). For ease of comparison, we also report their performance when prompted for *score prediction*, as well as the performance of the baselines. Overall, prompting LLMs with AUTOMQM seems to lead to significant improvements in evaluating machine translation quality, particularly for larger models: UNICORN achieves better performance (across all meta evaluations) with it than when prompted for *score prediction*,

		System-Level	Segment-Level			
		All (2 LPs)	EN-DE		ZH-EN	
Model	Ref?	Accuracy	ρ	acc*	ρ	acc*
Baselines						
MetricX-XXL	✓	81.1%	0.549	61.1%	0.581	54.6%
MATESE	✓	79.9%	0.391	58.8%	0.528	51.5%
COMET-QE	✗	76.9%	0.419	56.3%	0.505	48.8%
MATESE-QE	✗	73.4%	0.298	57.9%	0.468	50.1%
COMET-WL	✗	71.6%	0.418	57.1%	0.406	51.5%
Score Prediction						
PaLM-2 BISON	✓	86.4%	0.394	56.8%	0.322	49.3%
PaLM-2 UNICORN	✓	86.4%	0.401	56.3%	0.349	51.1%
PaLM-2 BISON	✗	84.0%	0.355	57.0%	0.299	48.6%
PaLM-2 UNICORN	✗	80.5%	0.275	56.1%	0.252	48.3%
AutoMQM						
PaLM-2 BISON	✓	84.0%	0.369	59.2%	0.355	48.4%
PaLM-2 UNICORN	✓	87.6%	0.432	59.1%	0.442	51.8%
PaLM 2 BISON	✗	87.6%	0.297	55.2%	0.331	48.0%
PaLM 2 UNICORN	✗	83.4%	0.368	56.4%	0.429	50.2%

Table 4: Meta-evaluation results for PaLM-2 models using *AutoMQM* and score prediction, at the system and segment levels for multiple language pairs.

and its reference-less version is competitive with the best learned metric even at the segment level. However, for the smaller BISON, the benefits of AUTOMQM are less clear, with both techniques performing comparably. This hints that *scale* is necessary for *zero-* and *few-* shot fine-grained evaluation (like with AUTOMQM). We also find that the *distribution* of scores produced by LLMs prompted with AUTOMQM is much closer to the gold MQM distribution, with models outputting a much larger set of scores, and in the same ranges as annotators do (see Figure 8).

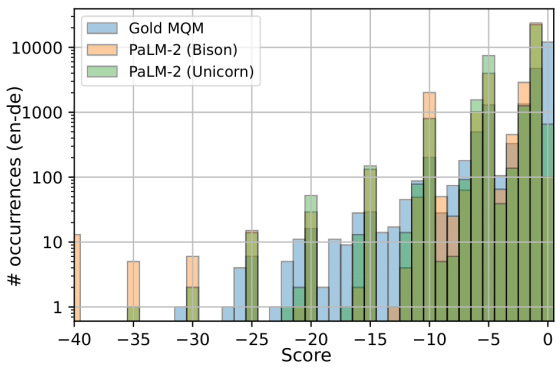

Figure 8: Distribution of scores for PaLM-2 models using AUTOMQM, on WMT22 EN-DE

Finally, when evaluating the error spans produced by LLMs prompted with AUTOMQM (Table 5), we find that PaLM-2 models are able to identify most of the *major* errors. However, it does seem to *over-predict* errors (with errors predicted by UNICORN having on average ∼5 words per span vs ∼2 words in the ground truth) and have overall low span precision. Similarly to overall *score* correlations, *scale* also seems to be important for the quality of spans produced by AUTOMQM, with UNICORN outperforming BISON at most metrics. Additionally, UNICORN prompted with AutoMQM predicts spans of comparable quality to the ones produced by current state-of-the-art *learned* word-level evaluators (trained on a considerable number of fine-grained annotations derived from MQM): while word-level models are more precise, their overall span correlation (MCC) is comparable, and they miss considerably more *major* errors than LLMs (despite only leveraging a handful of annotations).

		EN-DE			ZH-EN		
Model	R?	SP	MR	MCC	SP	MR	MCC
Baselines							
COMET-WL	✗	0.267	0.250	0.161	0.364	0.178	0.152
AutoMQM							
BISON	✓	0.095	0.749	0.060	0.252	0.255	0.109
UNICORN	✓	0.175	0.628	0.193	0.238	0.476	0.143
BISON	✗	0.119	0.520	0.092	0.224	0.311	0.091
UNICORN	✗	0.150	0.580	0.150	0.229	0.488	0.133

Table 5: Span-level meta-evaluation on WMT22 for PaLM-2 models using *AutoMQM*. **SR** and **MR** represent *span precision* and *major recall*, respectively.

7 Conclusion

In this study, we have systematically investigated the capabilities of large language models for machine translation evaluation through *score prediction*, and proposed AUTOMQM, a novel

prompting technique that leverages the Multidimensional Quality Metrics (MQM) framework for interpretable MT evaluation using LLMs.

We demonstrated that just prompting LLMs for score prediction leads to state-of-the-art system-level evaluators, but still falls short of the best *learned* metrics at the segment-level (with finetuning being necessary to close this gap). Then we showed that AUTOMQM can further improve the performance of LLMs without finetuning while providing interpretability through error spans that align with human annotations.

Our findings surrounding finetuning LLMs for *score prediction* hint that LLMs' performance in machine translation evaluation could be further improved by finetuning these models on fine-grained human judgment data (like MQM) and is a direction we are actively pursuing. Additionally, the general-purpose nature of LLMs may enable the application of similar prompting techniques (leveraging some fine-grained evaluation schemes) to other evaluation problems (Wu et al., 2023).

Acknowledgements

We would like to thank Ricardo Rei, Marcos Treviso and Chryssa Zerva for helping run the word-level QE baselines, and George Foster who provided feedback on an earlier version of this work. This work was partially supported by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the project DECOLLAGE (ERC-2022-CoG 101088763), by the Portuguese Recovery and Resilience Plan through project C645008882- 00000055 (Center for Responsible AI), and the Fundação para a Ciência e Tecnologia through contracts SFRH/BD/150706/2020 and UIDB/50008/2020.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Josh Schroeder, and Cameron Shaw Fordyce, editors. 2008. *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Columbus, Ohio.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pil-

lai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties Matter: Modifying Kendall's Tau for Modern Metric Meta-Evaluation.

Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José G. C. de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, and André F. T. Martins. 2023. Bridging the gap: A survey on integrating (human) feedback for natural language generation.

Erick Fonseca, Lisa Yankovskaya, AndrÃ© F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the wmt 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–12, Florence, Italy. 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, 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.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality.

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.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Revista Tradumàtica: tecnologies de la traducció*.

Qingyu Lu, Baopu Qiu, Liang Ding, Liping Xie, and Dacheng Tao. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

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.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo, and Roberto Navigli. 2022. MaTESe: Machine translation evaluation as a sequence tagging problem. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 569–577, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yiwei Qin, Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2022. T5score: Discriminative fine-tuning of generative evaluation metrics. *ArXiv*, abs/2212.05726.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

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, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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.

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

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. Instructscore: Towards explainable text generation evaluation with automatic feedback.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 69–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online. Association for Computational Linguistics.

A Multidimensional Quality Metric (MQM)

The Multidimensional Quality Metrics (MQM) framework is a flexible human-evaluation framework developed to evaluate and categorize errors in translations. Annotators are instructed to identify all errors within each segment in a document, paying particular attention to document context. See Table 6 for the annotator guidelines provided.

Annotators are asked to assign both an error *severity* and *category*. Error *severity* (either *major* or *minor*) is assigned independently of category. Spans with no marked errors have *neutral* severity and no category. Possible error categories are displayed in Table 7.

You will be assessing translations at the segment level, where a segment may contain one or more sentences. Each segment is aligned with a corresponding source segment, and both segments are displayed within their respective documents. Annotate segments in natural order, as if you were reading the document. You may return to revise previous segments.

Please identify all errors within each translated segment, up to a maximum of five. If there are more than five errors, identify only the five most severe. If it is not possible to reliably identify distinct errors because the translation is too badly garbled or is unrelated to the source, then mark a single *Non-translation* error that spans the entire segment.

To identify an error, highlight the relevant span of text, and select a category/sub-category and severity level from the available options. (The span of text may be in the source segment if the error is a source error or an omission.) When identifying errors, please be as fine-grained as possible. For example, if a sentence contains two words that are each mistranslated, two separate mistranslation errors should be recorded. If a single stretch of text contains multiple errors, you only need to indicate the one that is most severe. If all have the same severity, choose the first matching category listed in the error typology (eg, *Accuracy*, then *Fluency*, then *Terminology*, etc).

Please pay particular attention to document context when annotating. If a translation might be questionable on its own but is fine in the context of the document, it should not be considered erroneous; conversely, if a translation might be acceptable in some context, but not within the current document, it should be marked as wrong.

There are two special error categories: *Source error* and *Non-translation*. Source errors should be annotated separately, highlighting the relevant span in the source segment. They do not count against the five-error limit for target errors, which should be handled in the usual way, whether or not they resulted from a source error. There can be at most one *Non-translation* error per segment, and it should span the entire segment. No other errors should be identified if *Non-Translation* is selected.

Table 6: MQM annotator guidelines

Since MQM doesn't ask annotators for quality scores, those scores are derived automatically from the identified error spans and their classifications, based on a *weighting* of each error severity and category. Table 8 summarizes this weighting scheme, in which segment-level scores can range from 0 (perfect) to 25 (worst). The final segment-level score is an average over scores from all annotators. In some settings (e.g. calculating correlation for learned metrics), the scores are negated.

We use the same weighting to obtain scores from errors identified by AUTOMQM.

B Datasets' Statistics

See Table 9 for a summary of the number of systems and annotated segments per system in the evaluation datasets used in this work.

C Score Prediction Prompt

Figure 9 contains the GEMBA-SQM prompt that we used for our 0-shot experiments.

D Sampling in-context learning examples for AutoMQM

Figure 10 shows the rejection criteria used when sampling example sets as discussed in §4.

E Additional Results

Figures 11, 12, 13 and 8 present additional experimental results.

Error Category		Description
Accuracy	Addition	Translation includes information not present in the source.
	Omission	Translation is missing content from the source.
	Mistranslation	Translation does not accurately represent the source.
	Untranslated text	Source text has been left untranslated.
Fluency	Punctuation	Incorrect punctuation (for locale or style).
	Spelling	Incorrect spelling or capitalization.
	Grammar	Problems with grammar, other than orthography.
	Register	Wrong grammatical register (eg, inappropriately informal pronouns).
	Inconsistency	Internal inconsistency (not related to terminology).
	Character encoding	Characters are garbled due to incorrect encoding.
Terminology	Inappropriate for context	Terminology is non-standard or does not fit context.
	Inconsistent use	Terminology is used inconsistently.
Style	Awkward	Translation has stylistic problems.
Locale convention	Address format	Wrong format for addresses.
	Currency format	Wrong format for currency.
	Date format	Wrong format for dates.
	Name format	Wrong format for names.
	Telephone format	Wrong format for telephone numbers.
	Time format	Wrong format for time expressions.
Other		Any other issues.
Source error		An error in the source.
Non-translation		Impossible to reliably characterize distinct errors.

Table 7: MQM hierarchy.

```
Score the following translation from {src_lang} to {tgt_lang} with respect to the
human reference on a continuous scale from 0 to 100 that starts with "No meaning
preserved", goes through "Some meaning preserved", then "Most meaning preserved
and few grammar mistakes", up to "Perfect meaning and grammar".

{src_lang} source: "{source}"
{tgt_lang} human reference: "{reference}"
{tgt_lang} translation: "{candidate}"
Score (0-100): {score}
```

Figure 9: The *score prediction* prompt used in this paper. Equivalent to the `GEMBA-SQM` prompt in Kocmi and Federmann (2023). Parts in purple are only included for *reference-based* evaluation, while parts in orange represent slots for outputs and are only included for in-context examples.

Severity	Category	Weight
Major	Non-translation	25
	all others	5
Minor	Fluency/Punctuation	0.1
	all others	1
Neutral	all	0

Table 8: MQM error weighting.

LP	#Sys	#Seg
en→de	13	1315
zh→en	14	1875
en→ru	15	1315

LP	#Sys	#Seg
en→kk	11	998
kk→en	11	1000
en→gu	11	998
gu→en	11	1016

Table 9: The number of systems and segments that have MQM scores (left) and DA scores (right) used as ground-truth in this work.

```python
def check_icl_set(
    examples: pd.DataFrame,
    min_errors=3,
    majmin_threshold=2,
    cat_diversity=2,
    min_clen=20,
    max_clen=400,
):
    # Check if they have the same number of spans as severity/category
    if not examples.apply(
        lambda r:
          len(r['span']) == len(r['severity']) and len(r['span']) == len(r['category']),
        axis=1
    ).all():
      return False

    # Check if there are at least min_errors
    if examples['severity'].apply(lambda svs: len(svs)).sum() < min_errors:
      return False

    # Check that there's a balance of major and minor errors.
    major_count = examples['severity'].apply(lambda svs: sum([s=='major' for s in svs])).sum()
    minor_count = examples['severity'].apply(lambda svs: sum([s=='minor' for s in svs])).sum()
    if abs(major_count - minor_count) > majmin_threshold:
        return False

    # Check that at least cat_diversity error types are represented.
    categories = examples['category'].apply(lambda cs: [c.split("/")[0] for c in cs])
    represented_error_types = set().union(*categories.tolist())
    if len(represented_error_types) < cat_diversity:
        return False

    top_clen = examples.apply(
        lambda row: max(len(row[s]) for s in ('source', 'reference', 'candidate')
    ), axis=1).max()
    bot_clen = examples.apply(
        lambda row: min(len(row[s]) for s in ('source', 'reference', 'candidate')),
    axis=1).min()

    if top_clen > max_clen or bot_clen < min_clen:
      return False

    # All checks passed.
    return True
```

Figure 10: Rejection criteria used when sampling *in-context learning* examples for AUTOMQM.

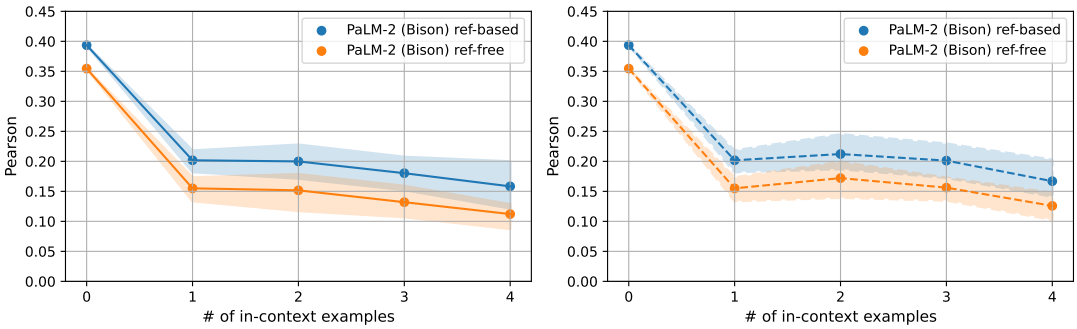

Figure 11: Mean *Pearson* and its interquartile range (IQR), as we increase the number of in-context examples in the *score prediction* prompt, sampled with *uniform* (left) and *stratified* (right) sampling, for WMT22 EN-DE.

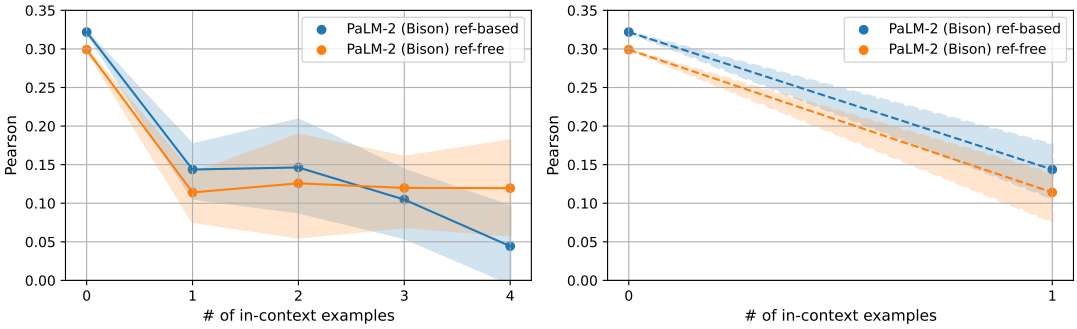

Figure 12: Mean *Pearson* and its interquartile range (IQR), as we increase the number of in-context examples in the *score prediction* prompt, sampled with *uniform* (left) and *stratified* (right) sampling, for WMT22 ZH-EN.

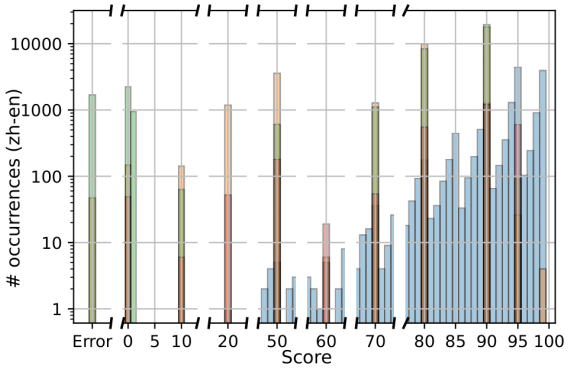

Figure 13: Distribution of scores for various LLM *reference-based* evaluators, on the ZH-EN test set. Note that the y axis is in *log-scale*.

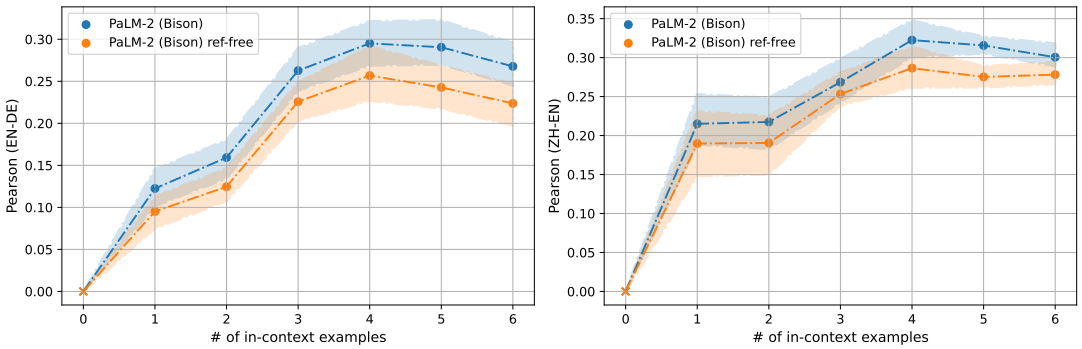

Figure 14: Mean *Pearson* and its interquartile range (IQR), as we increase the number of in-context examples in the AUTOMQM prompt, for EN-DE (left) and ZH-EN (right).