MetricX-25 and GemSpanEval: Google Translate Submissions to the WMT25 Evaluation Shared Task

Juraj Juraska, Tobias Domhan, Mara Finkelstein, Tetsuji Nakagawa, Geza Kovacs, Dan Deutsch, Pidong Wang, and **Markus Freitag**
Google Translate
{jjuraska,domhant,freitag}@google.com

Abstract

In this paper, we present our submissions to the unified WMT25 Translation Evaluation Shared Task. For the Quality Score Prediction subtask, we create a new generation of MetricX with improvements in the input format and the training protocol, while for the Error Span Detection subtask we develop a new model, GemSpanEval, trained to predict error spans along with their severities and categories. Both systems are based on the state-of-the-art multilingual open-weights model Gemma 3, fine-tuned on publicly available WMT data. We demonstrate that MetricX-25, adapting Gemma 3 to an encoder-only architecture with a regression head on top, can be trained to effectively predict both MQM and ESA quality scores, and significantly outperforms its predecessor. Our decoder-only GemSpanEval model, on the other hand, we show to be competitive in error span detection with XCOMET, a strong encoder-only sequence-tagging baseline. With error span detection formulated as a generative task, we instruct the model to also output the context for each predicted error span, thus ensuring that error spans are identified unambiguously.

1 Introduction

Large language models (LLMs) have been evolving at a breakneck speed over the past couple of years, achieving performance in machine translation (MT) that matches or even exceeds that of humans for certain languages and domains (Kocmi et al., 2024a). While human evaluation is still the most reliable way to assess the quality of MT models, in this fast-paced environment of state-of-the-art models being released on a monthly basis, the cost and duration make it infeasible to run human evaluation studies to benchmark models regularly. Improving automatic metrics is therefore instrumental to making further progress in MT, especially as the field expands to low-resource languages and more difficult domains.

Currently, the most successful automatic MT evaluation metrics are trained neural models themselves, following one of two main paradigms: (1) regression models predicting a scalar quality score (Rei et al., 2022a; Juraska et al., 2024), and (2) sequence-tagging or generative models, providing fine-grained quality feedback, including error spans, severities and categories (Fernandes et al., 2023; Guerreiro et al., 2024). In this work, we push the performance of automatic metrics in both categories higher by leveraging a state-of-the-art multilingual open-weights LLM, Gemma 3 (Gemma Team et al., 2025), resulting in two separate submissions to the WMT25 Evaluation Shared Task.

For the Quality Score Prediction subtask, we develop *MetricX-25*, a successor to MetricX-24 (Juraska et al., 2024), updated to use an encoder-only architecture and trained on a combination of publicly available direct assessment (DA) and Multidimensional Quality Metrics (MQM) scores from WMT shared tasks between 2015 and 2023. For the Error Span Detection subtask, we introduce *GemSpanEval*, a generative model that identifies and categorizes error spans in JSON format, trained exclusively on MQM error span annotations from WMT20–24. We enable GemSpanEval to uniquely identify short, non-unique error spans by training the model to also indicate the error span context where necessary.

The key takeaways from our experiments, detailed in this report, include:

1. Gemma 3 offers a strong multilingual foundation for an automatic MT evaluation metric, and adapting it to an encoder-only architecture proves highly effective for score prediction.
2. It is possible to train an automatic metric to effectively predict different types of score using a single regression head by simply mixing training examples of different scores, with a score type indication included in the input.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 510–521
November 8–9, 2025. ©2025 Association for Computational Linguistics

3. Fine-tuning a strong multilingual decoder-only model can be competitive with encoder-only error span detection models.
4. Predicting error span context can be used to uniquely identify error spans when formulating span detection as a generative task.

2 Data

The systems we developed for both the quality score prediction and the error span detection are trained solely on publicly available data from the WMT Metrics shared tasks between 2015 and 2024. Quality ratings for system translations across those 10 years were collected using 3 different methods:

1. Direct assessment (DA) scores, provided mostly by non-expert raters, on a scale from 0 to 100. WMT data with DA scores is available for years between 2015 and 2023, and covers nearly 50 language pairs.

2. MQM scores (Lommel et al., 2014; Freitag et al., 2021) on a scale from 0 to 25 (or uncapped in the most recent years), where lower is better. The scores are derived from professional translator annotations of error spans, including error severities and categories, where, generally, each minor error and each major error contribute 1 and 5 to the score, respectively. MQM annotations are only available for a limited number of language pairs (en-de, en-es, en-ru, en-zh, he-en, zh-en, ja-zh) for WMT data starting from 2020.

3. ESA scores (Kocmi et al., 2024b), which combine the first two approaches in that the expert raters annotate error spans and severities, yet they provide an overall quality score on a scale from 0 to 100 as well. These were only introduced in WMT24, so there is only one year worth of ESA data.

We use both DA and MQM scores for training MetricX, our score prediction system, and MQM error span annotations for training GemSpanEval. In general, we reserved data from WMT24 – both MQM (Freitag et al., 2024) and ESA (Kocmi et al., 2024a) – for validation of our models, with the exception of one submission to the error span detection task. We provide more details on the training and validation sets in §3 for MetricX and in §4 for GemSpanEval.

3 Quality Score Prediction: *MetricX-25*

Our MetricX-25 submissions to the Quality Score Prediction subtask are based on the successful MetricX-24 (Juraska et al., 2024; Freitag et al., 2024), with several modifications and improvements. The biggest one among them is the switch from mT5 (Xue et al., 2021) to Gemma 3 as the backbone model. We start this section by providing an overview of the similarities and differences between MetricX-24 and MetricX-25, then give more details on the training and evaluation data, and finally describe our experiments and the MetricX-25 systems we submitted to the shared task.

3.1 MetricX Overview

MetricX is a regression model trained to predict a quality score for a machine translation given the source segment and/or a reference translation. As of the '24 version, there are no separate models for reference-based and reference-free prediction; instead, a single model is trained on a mixture of examples: (1) with both the source and a reference, (2) with the reference omitted, and (3) with the source omitted.

The model is trained on translation evaluation data in two stages. It is first fine-tuned on z-normalized DA scores, then further fine-tuned on a mixture of MQM scores and raw DA scores. In both stages, a small proportion of synthetic training data, generated from WMT data, is included to help MetricX models recognize certain types of bad translations that are insufficiently represented in the standard WMT data. These synthetic examples cover cases such as over- and undertranslation, fluent but unrelated translation, and missing punctuation. We refer the reader to Juraska et al. (2024) for the full list of synthetic example categories and details on how the data was constructed.

3.2 What Is New in MetricX-25?

Initialization model. The model we initialize MetricX-25 from is Gemma 3 12B, a state-of-the-art multilingual open-weights model similar in size to the 13B-parameter mT5-XXL used in all previous versions of MetricX. In contrast to mT5-based MetricX models, MetricX-25 uses an encoder-only architecture with a regression head on top. Specifically, MetricX-25 is a fine-tuned Gemma Encoder (Suganthan et al., 2025) with mean pooling and no uptraining, and with the encoder's weights initialized from the corresponding

Czech source:
```Připadalo mi, že na mě dýchnul závan z hrobu.```

English (United Kingdom) reference:
```It was like having felt a draught from a grave.```

English (United Kingdom) translation:
```It was like having felt a draft from a grave.```

Score type: MQM

Figure 1: Example MetricX-25 model input.

decoder weights of Gemma 3. Besides the major differences in architectures and pretraining corpora and strategies, Gemma 3 also has the advantage of supporting significantly longer context windows (up to 128K tokens) than mT5 and has an even wider language support (over 140 languages, compared to mT5's 101).

Language indication. For MetricX-25, we augment the model input with source and target language information. The motivation behind this is to help MetricX recognize when an untranslated source (or portion of it) is appropriate, as well as help it handle quality assessment of translations from one language dialect to another without incorrectly assuming the translation is largely untranslated. Thus, we include the country information too if the locale is indicated in the data (e.g., ar_EG) and the language has multiple major dialects, such as Arabic (Egyptian, Modern Standard Arabic, etc.) or Portuguese (Brazilian, European, etc.). Figure 1 shows a full example of a model input.

Input format. Given the dual nature of human evaluation in the WMT25 shared task – MQM for some language pairs and ESA for others – we also add a score type indication to the model input, so that it learns to predict both types of quality score. We use the "MQM" score type for the MQM training and evaluation data, and "ESA" for the DA training and ESA evaluation data. Considering the possibly multi-paragraph segments in the official test sets, we also enclose each segment between triple backticks and separate input segments with double newline characters (see Figure 1).[1]

2-way hybrid input mode. We changed the training recipe for MetricX-25 by only including reference-only examples in the first stage of fine-tuning. In the second stage, we fine-tune the model on two types of examples only: (1) source-only, and (2) with source and reference both. The reason is twofold. First, MQM scores were produced in a source-based fashion, without a reference being available at all, so training examples with an MQM score but no source segment might not provide an accurate signal to the model. We verified experimentally that omitting these examples in the second stage does not have a significant negative impact on reference-only prediction performance. Second, in all the evaluation scenarios of the quality score prediction task, source segments are available, so there is little reason to distract the model with source-free training examples in the second stage of fine-tuning.

Score clipping. Earlier versions of MetricX had MQM scores in the training data, as well as the output scores, clipped to the [0, 25] range. However, with the switch to document-level segments over the past two years of the shared task, and the fact that MQM scores in human evaluation are therefore no longer capped at 25, we also drop the score clipping in MetricX-25, allowing for output scores greater than 25, which we expect to improve the correlation with MQM scores for long segments.[2]

3.3 Experimental Setup

Training data. In the first stage of fine-tuning MetricX-25, we use z-normalized DA scores from WMT15–23, with the into-English subset of WMT21 omitted due to its low quality (Juraska et al., 2023). Furthermore, the DA z-scores are aggregated per segment, negated, and finally clipped to the $[-1.0, 1.0]$ range, as shown by Juraska et al. (2023) to yield the best performance. We also incorporate a small proportion of the synthetic training data introduced in Juraska et al. (2024). In this stage, we do not include any score type indication in the input. In the second fine-tuning stage, we mix an equal proportion of the same DA data as above (with "ESA" indicated as the score type) and MQM data from WMT20–23 (with "MQM" score type), along with synthetic data included in each group of examples. In contrast to the first stage, however, we use raw DA scores rescaled to the MQM scale here, so the model does not have to learn two different scales, only distributions. We

[1] We experimented with including a preamble with instructions on the quality assessment task in the input too, but it did not improve the performance, perhaps except for the first few steps of fine-tuning.

[2] This causes a small discrepancy with the synthetic training data, which uses a fixed range of [0, 25], with many of the examples having a score of 25 assigned to them. Nevertheless, most of the synthetic examples are sentence-level, so an MQM score of 25 is reasonable for very bad translations.
```

then rescale the output scores to their respective ESA and negative MQM scales, as expected for the evaluation on the official test set, in postprocessing.

Meta-evaluation. Our validation set, which we use to pick the best model checkpoints, consists of both the MQM and ESA data from WMT24. To evaluate our models' performance, we calculate to what degree their predicted scores agree with the human judgments of translation quality. For segment-level correlation we use the tie-calibrated pairwise accuracy introduced by Deutsch et al. (2023), while at the system level we calculate soft pairwise accuracy (SPA; Thompson et al. 2024). These were the two primary meta-evaluation metrics used in the WMT24 Metrics Shared Task (Freitag et al., 2024). We use the same checkpoint selection strategy as for MetricX-24, averaging over all three MQM language pairs of WMT24, and downweighting the system-level component due to its larger variance.

Implementation details. MetricX-25 is implemented in TensorFlow (Abadi et al., 2015), and all of our submitted MetricX-25 systems are based on the Gemma 3 variant with 12B parameters. We defer further implementation details to Appendix A.

3.4 Results and Submission Details

Here we present the results of our experiments, focusing on assessing the impact of the combined MQM/ESA score prediction and comparing the performance of the Gemma-based MetricX-25 with that of the similarly-sized mT5-based MetricX-24. Due to limited resource availability, we were only able to run each experiment with one random seed.

3.4.1 Combining MQM/ESA Score Prediction

We start by examining the effects of mixing DA and MQM training data, together with using the score type indicators in the input. As a reminder, raw DA scores are used to train the model to predict ESA scores, since they both use the same scale and follow similar distributions. The first 3 rows of Table 1 compare the combined fine-tuning with fine-tuning on DA data only and MQM data only. We can see that the "DA + MQM" model (row 3) performs on par with the "MQM only" model (row 2) on the MQM validation sets and on par with the "DA only" model (row 1) on the ESA validations sets. This demonstrates that the model can effectively learn to predict both types of scores without sacrificing performance in either of them.

Our next observation is that simple two-stage fine-tuning (DA first then MQM) also achieves this goal, except for the system-level performance on ESA, which is around 2 points behind both the "DA + MQM" and the "DA only" model (compare row 4 against rows 3 and 1). Finally, we show that by combining two-stage fine-tuning and DA/MQM data mixing (row 5), we significantly boost the system-level performance on the ESA sets, while maintaining or further improving the performance on the MQM sets, as well as maintaining the segment-level performance.

3.4.2 MetricX-25 Submissions

Table 2 summarizes our MetricX-25 submissions to the quality score prediction task and compares them against MetricX-24 – one of the three top-performing systems in last year's shared task – as a baseline. The submissions only differ in the combination of examples they were trained on (with or without source/reference) and thus their expected input: the primary submission is a hybrid system, whereas the two secondary submissions are a purely quality estimation (QE) and a purely reference-based system, respectively.

As the table shows, all the MetricX-25 submissions (rows 3–6) significantly outperform the MetricX-24 baseline (rows 1–2) at the segment level, but the results are mixed at the system level. The ja-zh language pair exhibits by far the largest improvement, suggesting that Gemma 3 has a stronger understanding of Japanese and/or Chinese than mT5. Our expectation is that this is true for many more languages, making MetricX-25 a more robust automatic evaluation metric than its mT5-based predecessors.

We chose the hybrid model as our primary submission, despite being slightly outperformed by the reference-based variant (compare rows 6 and 4), because of its input flexibility. Since the official test set consists of language pairs both with and without references available, the reference-based model would likely perform poorly on the latter. Moreover, the majority of the challenge sets also do not provide references, so MetricX-25-Ref would not be able to participate in them.

4 Error Span Detection: *GemSpanEval*

In this section, we describe our submission to subtask 2: Error Span Detection. We denote our system *GemSpanEval*, as it is a span-level prediction model based on Gemma 3.

Training protocol	Segment-level pairwise accuracy				System-level soft pairwise accuracy			
	en-de	en-es	ja-zh	Avg(ESA)	en-de	en-es	ja-zh	Avg(ESA)
DA only	50.41	68.51	54.48	54.72	85.62	**82.13**	92.62	87.65
MQM only	54.71	68.80	56.36	54.20	85.55	78.56	89.94	86.45
DA + MQM	54.71	68.92	56.16	54.91	85.21	78.89	90.67	**87.80**
DA→MQM	55.40	69.11	57.90	55.00	85.91	78.12	**93.64**	85.74
DA→(DA + MQM)	**55.66**	**69.25**	**58.24**	**55.14**	**86.60**	77.92	92.88	87.08

Table 1: Meta-evaluation scores of reference-based models (non-hybrid) on the WMT24 validation set, using a variety of one- and two-stage fine-tuning protocols, with the → symbol indicating two stages. "DA + MQM" denotes the combination of DA and MQM scores, with a score type indication provided in the input. The correlation scores are shown for all 3 MQM language pairs individually, and for the 9 ESA language pairs averaged.

MetricX variant	Segment-level pairwise accuracy				System-level soft pairwise accuracy			
	en-de	en-es	ja-zh	Avg(ESA)	en-de	en-es	ja-zh	Avg(ESA)
24-Hybrid-QE	52.60	68.50	53.00	–	**87.80**	78.90	87.50	–
24-Hybrid	53.20	68.50	53.90	–	87.40	**79.90**	89.70	–
25-QE†	54.97	**69.42**	57.21	54.34	85.45	78.29	91.34	84.91
25-Ref†	**55.66**	69.25	**58.24**	**55.14**	86.60	77.92	**92.88**	87.08
25-Hybrid-QE	54.83	69.31	56.66	54.11	85.42	77.74	91.59	86.32
25-Hybrid*	55.45	69.14	57.72	54.87	85.82	77.00	92.00	**87.61**

Table 2: Performance of our MetricX-25 submissions compared to the MetricX-24 baseline on WMT24. "Hybrid" and "Hybrid-QE" rows correspond to the same model, only evaluated with and without references, respectively. The correlation scores are shown for all 3 MQM language pairs individually, and for the 9 ESA language pairs averaged. *Primary submission. †Secondary submissions.

4.1 Overview

Last year's shared task on error span detection (Zerva et al., 2024), despite low participation, showed that span-level error prediction remains a challenging task. Specifically, Shan et al. (2024) found that generative methods based on LLMs, such as GPT-4o-mini (Hurst et al., 2024) or Tower-Instruct-7B (Alves et al., 2024), still lag behind encoder-only models like COMETKIWI (Rei et al., 2022b). Our shared task submission aims to explore how far we can push generative models at the error span detection task when using a recent strong multilingual open-weights model, in our case Gemma 3 27B, fine-tuned for the error span detection task.

4.2 Error Span Detection

We adapt the AutoMQM setup (Fernandes et al., 2023) by predicting MQM error spans in JSON format similar to Finkelstein et al. (2024). LLMs tend to be good at producing valid JSON output, making parsing the structured output straightforward. Each error contains the text span of the machine translation or the source segment (for omissions) along with a severity and a category. Note that for the error span detection task the category and source errors, such as omissions, are not used.

While models are able to identify and extract spans, we also need to find the spans in the original text for this task. A simple way to do so is to perform a string search. This is successful for most error spans, as they are unique substrings of the source or the machine translation. However, there is also a considerable portion of error spans that are not unique, often related to punctuation or short frequent words. For example, in the WMT24 en-de MQM data, 21% of error spans are not unique. Note though, that the problem is less pronounced for the shared task evaluation, as it is based on the character F1 score, to which short spans contribute less than long spans.

To be able to uniquely identify short spans, we modify the model response to include additional context for any span that is not unique. We expand the context to the previous and next word by looking for the previous and next space character. For Chinese and Japanese, we extend the context by one character at a time. The context is extended until we find a unique substring. See Figure 2 for an example of translation error spans with context in JSON format, and Figure 3 in the Appendix for the full prompt and output. The rest of the prompt and format follows Finkelstein et al. (2024). We do not use ICL examples, as we train a dedicated model. All data is presented twice: once with the

```
English source:
‘‘‘I have not made use of the timer,
    preferring to turn them on and off
    myself. I can see this feature as
    useful in an office setting with
    houseplants or if on vacation‘‘‘
German machine translation:
‘‘‘Ich benutze den Timer nicht, sondern
    schalte ihn lieber selbst ein und
    aus. Ich sehe diese Funktion als
    nützlich im Büro mit Zimmerpflanzen
    oder im Urlaub.‘‘‘
```

Response:

```
[
  {"span": "im", "span_with_context":
      "nützlich im Büro", ...},
  {"span": "ihn", ...},
  {"span": "mit", ...}
]
```

Figure 2: Example translation with non-unique error spans, where span context text is included.

reference translation and once without. This allows us to evaluate the model as a QE and as a reference-based model both.

4.3 Experimental Setup

For development, we train on MQM data from WMT20–23 and evaluate on WMT24 MQM data. We evaluate using character-level F1 score, which was used as the shared task metric in previous years (Zerva et al., 2024) and also the current year. The metric takes into account error severities and gives partial credit for predicting an error span with the wrong error severity. For the submission system, we include the en-de and ja-zh WMT24 MQM data in training as well, holding out en-es for evaluation. We include the latest data in order to fine-tune the model on longer segments too. Before WMT24, only WMT23 en-de provided paragraph-level data, while all other data is at the sentence level. Additionally, we hope to increase the coverage of translation errors of modern LLM-based translation models.

We fine-tune a 27B Gemma 3 model using Kauldron SFT tooling.[3] We use the Adafactor (Shazeer and Stern, 2018) optimizer with a learning rate of 0.0001 and a batch size of 64, running for 20K steps, which covers a little under 2 epochs of the training data. As baselines we report XCOMET-XXL (Guerreiro et al., 2024) scores. XCOMET is a strong encoder-only model that was trained to rate segments and also label translation error token spans. Additionally, we evaluate Gemma 3 27B without fine-tuning, prompting it to produce JSON error spans, as shown in Figure 3. Note that this baseline does not include the span context for short spans.

[3] https://kauldron.readthedocs.io/en/latest/

4.4 Results and Submission Details

The experimental results for the span-level error prediction task are shown in Table 3. GemSpanEval-QE v1 denotes an initial training run for just 10K steps and without span context for short spans. GemSpanEval-QE and GemSpanEval are the QE and reference-based evaluations, respectively, of a model trained without WMT24 data. The final row adds WMT24 en-de and ja-zh data. Therefore, for the last row, we should only take the results as an indication of how much of WMT24 has been memorized, not of how good the model is, except for en-es. The final shared task submission is based on the model that was trained *with* WMT24 data. The primary submission uses the reference while the secondary submission is reference-free. Both submissions use the same model and just differ in whether references are shown at inference time. Note that, when preparing the submission data, we ran into model repetition problems that resulted in invalid JSON output. For these cases we fell back to the original model, GemSpanEval-QE v1. For the shared task submission we find 22% of errors spans to not be unique, while this corresponds to only 5% of the characters of error spans, as non-unique spans tend to be short. Consequently, for WMT24 en-de this leads to a marginal F1 improvement of 0.08.

System	en-de	en-es	ja-zh	Avg.
XCOMET-XXL-QE	24.28	10.11	14.30	16.23
XCOMET-XXL	**25.43**	11.02	24.94	**20.46**
Gemma 3 27B	17.94	8.19	**28.42**	18.18
GemSpanEval-QE v1	17.51	14.43	22.75	18.23
GemSpanEval-QE	20.85	13.06	24.72	19.54
GemSpanEval	21.79	13.73	25.28	20.27
+ WMT24 train	27.26	**14.37**	37.09	26.24

Table 3: WMT24 character level F1 scores for the error span prediction task. Numbers where we train on the development set are grayed out but kept for reference.

From the experimental results during development in Table 3, we see that the encoder-only XCOMET-XXL model is a strong baseline showing the best result for en-de. The Gemma 3 baseline that was not fine-tuned also shows generally good performance, even achieving the best

result of all evaluated systems for ja-zh, better than the fine-tuned model. This is likely due to the task being difficult and highly dependent on rater behavior, which varies significantly across years and languages. Using references consistently achieves higher character F1 across all three language pairs. Surprisingly, the model trained with WMT24, while showing the best results in terms of character F1, still shows a significant gap, despite seeing the test data during training for en-de and ja-zh. This shows that ∼2 epochs in the current training setup was not enough to completely memorize the training data. For the held-out language, en-es, we see the best score across all settings, leading to the decision to use the model trained with WMT24 as our submission to the shared task.

5 Related Work

For decades, right until the recent advent of LLMs, the most widely adopted automatic evaluation metrics would express the predicted machine translation quality as a scalar score. This is the case for metrics ranging from simple lexical overlap metrics, such as BLEU (Papineni et al., 2002) and ChrF (Popović, 2015), to learned metrics including BLEURT (Sellam et al., 2020; Pu et al., 2021), COMET (Rei et al., 2020, 2022a) and MetricX (Juraska et al., 2023, 2024). The feasibility of prompting general-purpose LLMs to score translations has also been studied (Kocmi and Federmann, 2023b; Leiter et al., 2023; Leiter and Eger, 2024) and this approach has been shown to be competitive with fine-tuned dedicated models, especially in system-level performance. Among recent methods, there has been an increasing proportion of metrics providing structured (Perrella et al., 2022; Kocmi and Federmann, 2023a; Fernandes et al., 2023; Guerreiro et al., 2024) or natural language explanations (Xu et al., 2023) for the predicted scores, most of which are based on LLMs.

Since the era of lexical metrics, which require one or more reference translations to evaluate a machine translation, metrics have evolved to rely increasingly more on the source segment (Rei et al., 2020, 2022a; Juraska et al., 2024). This is enabled by the multilingual pretrained models they are typically built on top, such as XLM-R (Conneau et al., 2020) or mT5 (Xue et al., 2021). In fact, *reference-free* (or *quality estimation*; QE) metrics do not lag far behind their *reference-based* counterparts anymore, as evidenced by the most recent WMT Metrics shared tasks (Freitag et al., 2023, 2024). That being said, high-quality references do provide added value to automatic metrics in most cases, helping them make even more accurate quality predictions. Therefore, metrics nowadays typically employ a unified (or hybrid) input approach, allowing them to make a reference-based prediction whenever a reference is available, and a QE prediction otherwise (Wan et al., 2022; Guerreiro et al., 2024; Juraska et al., 2024). This is the case with our primary MetricX-25 and GemSpanEval submissions as well.

Current approaches to error span annotation are often based on encoder-only models predicting error severities per token. One such approach is COMETKIWI (Rei et al., 2022b, 2023) or, more recently, XCOMET (Guerreiro et al., 2024). In last year's Error Span Detection shared task, COMETKIWI was shown to be a competitive baseline, coming in ahead of the (single) submitted system (Zerva et al., 2024). Kocmi and Federmann (2023a) showed that GPT-4 can simply be prompted to produce MQM error spans, denoting the method GEMBA-MQM. Fernandes et al. (2023) showed that fine-tuning for the generative error span prediction can improve performance compared to prompting LLMs alone. Recent translation-oriented LLMs, such as Tower (Alves et al., 2024; Rei et al., 2025) also include generative error span prediction as part of the supported tasks in the training data.

6 Conclusion

We introduced MetricX-25 and GemSpanEval, our submissions to the WMT25 Evaluation Shared Task, both built upon the Gemma 3 foundation model, but in very different ways. We demonstrated that MetricX-25, adapting Gemma 3 to an encoder-only architecture, can be trained to effectively predict ESA and MQM quality scores and significantly outperforms its predecessor, MetricX-24, in segment-level performance. For error span detection, our generative model, GemSpanEval, proved to be competitive with a strong sequence-tagging baseline. Additionally, we showed how error span context can be used to identify unique error spans. Our work demonstrates that a strong multilingual foundation model, such as Gemma 3, can successfully be used for both regression-based and generative translation evaluation metrics.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks.

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: 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.

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.

Mara Finkelstein, Dan Deutsch, Parker Riley, Juraj Juraska, Geza Kovacs, and Markus Freitag. 2024. From jack of all trades to master of one: Specializing llm-based autoraters to a test set. *arXiv preprint arXiv:2411.15387*.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. 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, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.

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.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil

Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024a. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. 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, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024b. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453, Miami, Florida, USA. Association for Computational Linguistics.

Christoph Leiter and Steffen Eger. 2024. PrExMe! large scale prompt exploration of open source LLMs for machine translation and summarization evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11481–11506, Miami, Florida, USA. Association for Computational Linguistics.

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.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica*, (12):0455–463.

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.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the*

Tenth Workshop on Statistical Machine Translation, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. 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, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Ricardo Rei, Nuno M Guerreiro, José Pombal, João Alves, Pedro Teixeirinha, Amin Farajian, and André FT Martins. 2025. Tower+: Bridging generality and translation specialization in multilingual llms. *arXiv preprint arXiv:2506.17080*.

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

Weiqiao Shan, Ming Zhu, Yuang Li, Mengyao Piao, Xiaofeng Zhao, Chang Su, Min Zhang, Hao Yang, and Yanfei Jiang. 2024. HW-TSC 2024 submission for the quality estimation shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 535–540, Miami, Florida, USA. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Paul Suganthan, Fedor Moiseev, Le Yan, Junru Wu, Jianmo Ni, Jay Han, Imed Zitouni, Enrique Alfonseca, Xuanhui Wang, and Zhe Dong. 2025. Adapting decoder-based language models for diverse encoder downstream tasks. *arXiv preprint arXiv:2503.02656*.

Brian Thompson, Nitika Mathur, Daniel Deutsch, and Huda Khayrallah. 2024. Improving statistical significance in human evaluation of automatic metrics via soft pairwise accuracy. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1222–1234, Miami, Florida, USA. Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore. Association for Computational Linguistics.

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, Frederic Blain, José G. C. De Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M. Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and André Martins. 2024. Findings of the quality estimation shared task at WMT 2024: Are LLMs closing the gap in QE? In *Proceedings of the Ninth Conference on Machine Translation*, pages 82–109, Miami, Florida, USA. Association for Computational Linguistics.

A MetricX-25 Implementation Details

With the input context length of 4096 SPM tokens, each training run requires 64 TPUs. Using a batch size of 128, we train our models for 10K steps in the first stage, using a learning rate of 0.00005 with a cosine decay following 100 steps of linear warm-up. We then fine-tune the best checkpoint for another 10K steps in the second stage, lowering the peak learning rate to 0.00001.

Prompt

```
You are an annotator for the quality of machine translation. Your task is to
    identify errors and assess the quality of the translation.
Based on the source segment, human-generated reference translation, 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 severities: 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.

Make sure your response is a strict and valid json object that could be parsed with
    json.loads() in python.

English source:
‘‘‘The lights are dimmable, but I use the strongest setting only. I have not made
    use of the timer, preferring to turn them on and off myself. I can see this
    feature as useful in an office setting with houseplants or if on vacation‘‘‘
German machine translation:
‘‘‘Die Lichter sind dimmbar, aber ich benutze nur die stärkste Einstellung. Ich
    benutze den Timer nicht, sondern schalte ihn lieber selbst ein und aus. Ich sehe
    diese Funktion als nützlich im Büro mit Zimmerpflanzen oder im Urlaub.‘‘‘
```

Response:

```
[
  {"span": "im", "severity": "minor", "category": "accuracy/mistranslation",
      "span_with_context": "nützlich im Büro"},
  {"span": "ihn", "severity": "minor", "category": "accuracy/mistranslation"},
  {"span": "mit", "severity": "minor", "category": "accuracy/mistranslation"}
]
```

Figure 3: Example prompt and response for AutoMQM error span identification. We omit the error span attribute `is_source_error` for brevity. Each span that is not unique receives an additional attribute `span_with_context`.