

Vertical: Quality Estimation of Machine Translation from Université Grenoble Alpes

Mariam Nakhle^{1,2}, Marco Dinarelli¹, Raheel Qader², Hervé Blanchon¹

¹Université Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

²OVHai LLM, 2 rue Kellerman, 59100 Roubaix, France

contact: mariam.nakhle@univ-grenoble-alpes.fr

Abstract

This paper describes the submission of Université Grenoble Alpes and OVHai LLM to the Eleventh Conference on Machine Translation (WMT26) Shared Task on Automated Translation Quality Evaluation Systems. We participate in the Task 2 - Segment-Level Quality Score Prediction. We present our system that is trained to predict a single quality score per input segment. Our contributions are the following: 1) we present a metric that uses a Large Language Model (LLM) as backbone, namely gemma-3-1b-it, thus leveraging its long context size and its large-scale training on 2 trillion tokens, 2) we propose a method to adapt a decoder-only model to a downstream task by training a special token and using it as the sequence representation and 3) we propose a training curriculum designed to boost metric performance on long inputs. Our metric is trained using publicly available data and it can predict a score with or without a reference. Preliminary results show that our method outperforms baseline models when provided with a reference on the system level. The reference-free variant ranks second on the segment level.

1 Introduction

Fast and reliable evaluation is crucial to the development of Machine Translation (MT) models. While MT models have quickly advanced during the last years, with great focus being placed on document-level translation, the evaluation metrics somewhat lag behind (Peng et al., 2025). One of the challenges was the context limit, which was traditionally 512 tokens and the difficulty of models to use the relevant parts of context from long texts. With the arrival of Large Language Models (LLMs), the context size has been increased dramatically: from the traditional 512 tokens input (Devlin et al., 2019; Bao et al., 2021) to thousands of tokens (Team et al., 2024). LLMs are versatile models capable of solving different tasks, and

are trained on trillions of tokens. Because of their long context size, these models can natively support document-level translation and their extensive pre-training and post-training makes them more capable of making use of longer context. However, a reliable evaluation of document-level MT is still a challenge, since most of the best performing metrics rely on encoder-only models that maintain the 512 token limit. That means that the performance of models on inputs exceeding this limit remains hard to measure. This gives a surprising advantage to the superficial n-gram matching metrics, such as BLEU (Papineni et al., 2002) and ChrF (Popović, 2015) that have no context size limitation and are computed equally regardless of length. However, their limitations and weaknesses are well known (Mathur et al., 2020), therefore an alternative capable of evaluating longer inputs is needed.

In the past years, the WMT conference provided test sets at the sentence level. In 2025, the conference introduced "multi-sentence" units for the first time, thus parting from the sentence-level paradigm. In this work, we will refer to this unit as "paragraphs". Therefore, the systems participating in this year's Shared Task must be able to evaluate these longer sequences. That is why in the design of our proposed system we concentrate on boosting performance on longer inputs. First, we choose an LLM as backbone for our model, namely gemma-3-1b-it (Team et al., 2025), which offers both a longer context size, necessary for paragraph-level inputs, and greater overall capabilities thanks to its extensive pre-training and post-training on 2 trillion tokens, compared to the encoder-based models used in existing metrics, such as COMET. Second, we propose a method to adapt this decoder-only model to the downstream task by using the hidden state of a special token inserted after the input sequence. Third, to further improve performance on long inputs, we design a training curriculum split into three stages: sentence-level data,

reconstructed paragraphs, obtained by recombining sentence-level test sets following the methodology of Deutsch et al. (2023b), and real paragraphs.

2 Data

2.1 Sources

For the training of the MT evaluation metric, we need translation data paired with a quality score, ideally coming from human judgements. We used the following sources of data:

WMT 2019-2024: The main part of the data, issued from the MT-metrics-eval framework (MTME)¹, containing 1.1 million sentence-level samples.

Reconstructed paragraphs from MTME 2019-2024: Following Deutsch et al. (2023b), we used the document ids of the segments to reconstruct paragraphs from the sentence-level data, by concatenating all consecutive segments that have a human score annotation. The score aggregation was computed as a text-length weighted average (Haq et al., 2025), meaning the longer segments contribute more to the final multi-sentence score. This resulted in 125 thousand samples of reconstructed paragraphs.

Natural paragraph-level data, MTME 2025: the last year of WMT is the first (and currently the only) one that does not use sentences as the main unit but rather multi-sentence (paragraph) segments. They contain 54 thousand paragraph-level samples.

Synthetic data: We experiment with adding synthetic data. We corrupt the MTME data to provide more low-quality samples. We produce the following failure categories: empty translation, over-translation (repeated translation), gibberish (random words sampled from the references), unrelated (translation replaced with a random but fluent reference), hallucination loop (repeated loop on the end of low-quality translations - only produced from paragraph-level data), truncation (drop last 60% of words - only produced from paragraph-level data). These samples are assigned the lowest score. We also generate a reference-match category with a perfect score. We add 7% of synthetic data to every stage of training.

2.2 Score Scaling and Normalisation

The human evaluation protocol employed at WMT has been changing over the years. Therefore, the

data contains different types of human scores. The types of scores that we use for training are the following: 'da-sqm', 'esa', 'esa-merged', 'mqm', 'wmt-appraise-z', 'wmt-z' (using the MTME naming convention). We don't use the raw direct assessment scores (named 'wmt-appraise' and 'wmt') since we use the z-normalized scores instead: these are normalized per rater and are provided by the WMT organizers.

These scores all follow "the higher the better" logic but they are on different scales:

- z-normalized DA scores: unbound, with mean at 0, can span from -15 to +15
- ESA, DA-SQM scores: on a 0 to 100 scale
- MQM scores: can be unbound from 0 to -inf, or capped at -25²

In order to be able to train on all of the available corpora, we normalized the data to be on a shared scale from 0 to 1 with 1 being the perfect score. We decided on this scale for three reasons: it follows the ESA protocol, we find it easy to interpret, and it aligns with the BLEU and COMET scales. We performed these data normalizations:

- z-scores: first, the scores are clipped to a practical minimum and maximum value. The minimum value (-1.5347) was calculated by averaging the z-scores for translations where the raw DA score is equal to 0 (as proposed in xComet (Guerreiro et al., 2024)), and we set the maximum value to 1³. Finally, a min-max normalisation is performed using the cited values.
- ESA and DA-SQM: rescaled to 0-1 by dividing by 100.
- MQM: we follow the convention of -25 being the worst possible value, and we divide the given score by 25 and add the result to 1.

3 Model architecture

The architecture of this model is represented in Figure 1. We use the model google/gemma-3-1b-it as the backbone for our system, due to its broad

²We dropped the wmt21.tedtalks_en-ru and wmt21.news_en-ru test sets that use a different scale.

³The calculated maximum value on segments having a raw score of 100 was 0.77. However, looking at the data plots, we decided to keep it at 1, in order to preserve the score variations in the upper bound.

¹<https://github.com/google-research/mt-metrics-eval>

language support of over 140 languages and strong performance in translation. We feed the inputs to the backbone model jointly, using the following prompt:

Evaluate the quality of this machine translation.

Source ({src_lang}):
{source}

Machine translation ({tgt_lang}):
{translation}<sep1>

Reference translation ({tgt_lang}):
{ref}<sep2>

We employ a very minimal prompt inspired by MetricX-25 (Juraska et al., 2025) that also provides the model with a very minimal input (only segment and language indication together with the expected score type).

The hidden states produced by all layers (including the embedding layer) are combined using a layer-wise attention mechanism (Guerreiro et al., 2024)⁴. We mimic the method of using the <cls> token in encoder-only models when fine-tuning on downstream tasks. Since our backbone model is a decoder, we cannot use a prepended token as it is only informative when using the encoder’s self-attention mechanism. Instead, we add a special token at the end of the sequence. The hidden state of this token is fed to a two-layer feed-forward network that is trained to predict the segment quality score.

We follow the unified training approach (Guerreiro et al., 2024; Wan et al., 2022), where we train the model both using the reference and without it. In order to do so, we run one forward pass through the backbone using the full sequence (including the reference). We then compute the sentence embedding to create two scenarios: 1) src_ref: for evaluation of the hypothesis using the source and reference as input, and 2) src: for quality estimation-style evaluation, i.e. without a reference.

Since the backbone is a decoder-only model, we can optimize the training and only run the forward pass through the backbone once and still keep the two scenarios, thanks to the fact that the inserted special tokens can only attend to previous tokens.⁵

⁴We use softmax transformation for the per layer weights

⁵In contrast, if using an encoder-only model we would be obliged to compute the forward pass twice.

For the src_ref scenario, we use the hidden state of the second separator (sep2), which attends to the full sequence. For the src scenario, we use the hidden state of the first special token (sep1) that only attends to the source and translation. The placement of the two special tokens is shown in the prompt.

For both scenarios we obtain the prediction scores $\{y^{src}, y^{src_ref}\}$, and the loss $\mathcal{L}$ is computed as a mean squared error between the human score (y) and the predicted score for each scenario. The final loss is computed like the weighted average of the two losses:

$$\mathcal{L} = (1 - w) * \mathcal{L}^{src} + w * \mathcal{L}^{src_ref} \quad (1)$$

where w is the weight of the src_ref variant, which can be tuned. In our experiment w was set to 0.5.

In the case of reference-free data, these can also be exploited at training time, by only computing the $\mathcal{L}^{src}$ loss. At inference time, the reference is optional. This is a property of decoder-only models where the tokens of the source-only variant are not impacted by the reference tokens, hence we can still use the model without a reference. When available, the scores obtained from the two variants are combined as follows:

$$\hat{y} = w \cdot y^{src_ref} + (1 - w) \cdot y^{src} \quad (2)$$

where w is the same weight coefficient as in Equation 1. Therefore, at inference time, we have three kinds of scores for the reference-based segments: the y^{src_ref} score, the y^{src} score and the combined $\hat{y}$ score.

4 Experimental setup

4.1 Validation Data Selection

We do not keep the full latest year of WMT data for the validation set, as is common practice at this shared task, because that would leave us with no authentic paragraphs to train on. Therefore, the evaluation set used during all of our training phases is a 20% unused portion of WMT 2025 data (12 523 samples). This way, we ensure we have similar distributions in the training and evaluation sets, as shown in Figure 2. When selecting the data for evaluation, we make sure that for every language pair, every domain is present in both training and evaluation. When possible, we don’t include the

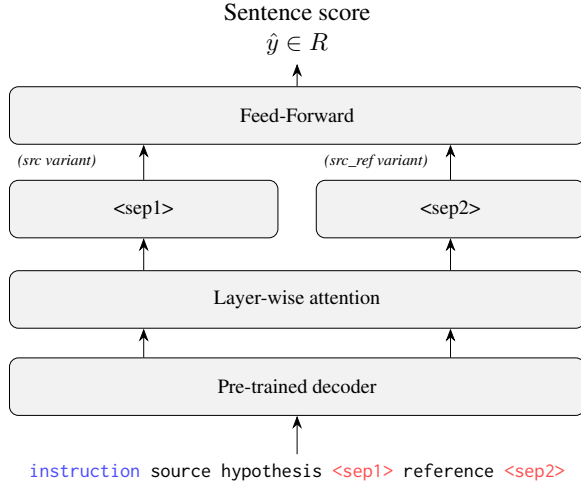


Figure 1: Architecture of the proposed model. The full input passes through the backbone decoder model. All hidden states obtained from the backbone are mixed into one representation using a layer-wise attention mechanism. The layer-wise attention is computed separately for the src and src_ref variants. We use the embeddings of separator 1 and separator 2 as representations of the full sequence. They are fed to a feed-forward network that is trained to predict a sentence score.

segments coming from the same document in both training and evaluation sets as they can be quite similar (in topics, writing style, tone, etc.) and could lead to overly confident evaluation scores. For some domains, such as the literary domain, this was not possible since for some language pairs, all segments belonging to a domain were sourced from the same document.

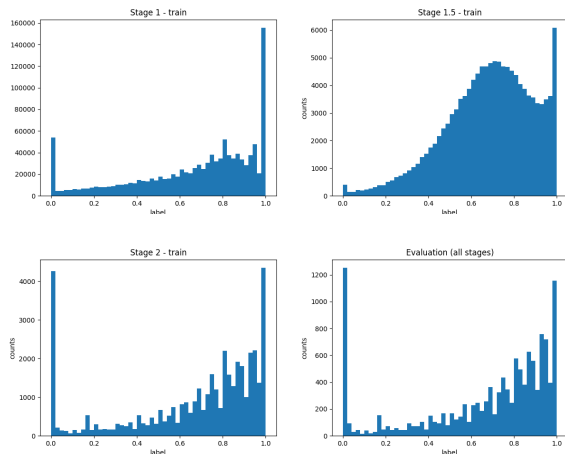


Figure 2: The data distribution across the stages. We can see that last stage’s training data has the most similar distribution with the evaluation set.

Hyper-parameter	VERTICAL
Backbone	gemma-3-1b-it
Optimizer	adafactor
Learning rate	5e-6
Learning rate scheduler	cosine
Loss function	MSE
Layer-wise dropout	0.1
Feed-Forward hidden units	3072, 1536
Feed-Forward activations	Tanh
Feed-Forward dropout	0.1

Table 1: Hyper-parameters used to train our metric.

4.2 Curriculum Training

During training, we faced the difficulty of paragraph-level data being underrepresented compared to the sentence-level data. Since the shared task aims at evaluating paragraphs, we combined two strategies to mitigate this difficulty: 1) recreating the paragraphs (on the data side), and 2) defining a curriculum-learning strategy (on the training side). Figure 2 shows the data distribution for every training stage. We divide the training data into 3 stages:

- stage 1: only sentence-level data (WMT data up to 2024) (1 128 732 samples)
- stage 1.5: reconstructed paragraphs (125 629 samples)
- stage 2: 80% of WMT 2025 as training data. (42 314 samples)

At each stage, the model’s parameters are initialized with parameters of the model trained at the previous stage (except for stage 1).

4.3 Training Setup

The details of our training hyperparameters are shown in Table 1. We apply the tokenizer’s chat template to all training samples. As for the special tokens, we used two of the tokenizer’s reserved tokens: <unused0> and <unused1>; these are chosen arbitrarily but remain constant during training and inference. In contrast to xComet training, we do not freeze the backbone. The special tokens we use as sequence representation carry no meaningful pre-trained representation, therefore their embeddings must be trained. A frozen backbone would pool a near-random vector⁶.

⁶Contrarily to xComet that pools a pre-trained CLS token.

The checkpoint selection is done using the Kendall correlation computed between the model’s prediction and true labels. The training was performed on 4 GPUs per node across 2 nodes using DeepSpeed configuration.

4.4 Shared Task submission

We submit two sets of scores. In both submissions, we use predictions from the source-reference variant of our model whenever possible, falling back to the reference-free variant otherwise. We later discovered a training time bug affecting the reference-free predictions, so we expect the scores on reference-free segments to be of lower quality. The first submission, Vertical_1, was trained entirely under this error. The second, Vertical_2, uses a checkpoint whose stage 2 was retrained after fixing the bug. For the remainder of the paper, we report results using the corrected and fully retrained method.

5 Results

5.1 Meta-evaluation

At the time of writing of this article, the official results of the WMT26 Shared Task are not available. Therefore, we report the scores on our held out evaluation set, described in Section 4.1. For the baseline metrics, we used the scores as reported in the MTME framework (using only the scores relating to our evaluation set’s segments in order to have fully comparable scores). We compute system level soft pair-wise accuracy (SPA) (Thompson et al., 2024) and segment-level pairwise accuracy with tie calibration (acc23) (Deutsch et al., 2023a).

Table 2 reports both measures under three aggregation settings: averaged over all 16 language pairs (_all), averaged over the 14 reference-based pairs only (_ref), and averaged over the 2 reference-free pairs only (_reffree), with each pair weighted equally.

We report the results of our method Vertical under its two variants: 1) Vertical_srcref, the reference-based variant and 2) Vertical_src, the reference-free variant.

First, we notice that there is a significant drop between Vertical_srcref and Vertical_src, which suggests that despite the unified training, the reference is still proving useful for the model. Vertical (srcref) ranks first overall in the SPA_all setting against all of the baselines. Although Vertical_src’s SPA_all score is comparatively low, it ranks second

on acc23_reffree, behind only TASER, suggesting that on the segment level, it is competitive compared to other reference-free metrics.

We notice that the LLM-based metrics (prompting) perform better on the system level than the segment level, while the surface-based and learned metrics follow the opposite pattern, tending to perform better at the segment level. Our method follows the same behaviour as the other learned metrics.

We remind the reader that our method’s scores may be overly optimistic, as the model was exposed to similar texts during training. As explained in Section 4.1, for some language pairs the literary domains only contained one document, therefore, the training and evaluation data contain similar segments sourced from the same document. This may lead to a bias since the model saw stylistically and topically similar texts during training.

Limitations

The scores submitted for the Shared Task were affected by a bug in the formulation of the reference-free variant: the layer-wise attention was computed over the entire sequence, so reference tokens leaked into the reference-free representation. We corrected this by keeping the single backbone pass but running the layer-wise attention twice: once per variant, each time using a mask truncated at that variant’s special token. So the reference-free representation is normalised over only the source and the hypothesis. The reference-based path is unaffected. Due to time constraints, we were not able to submit scores produced by the correctly retrained model, however we use the corrected checkpoint for the result section of this paper. Therefore, we expect the official scores on reference-free samples to be of lower quality than those reported here.

We test the effectiveness of this architecture mainly using public WMT data. Future work will test different mixes of paragraph and sentence samples, as well as synthetic data. Due to scarcity of paragraph-level data, we test our method only on a selected portion of WMT25 data, which despite our careful evaluation data selection method can introduce bias into the evaluation.

Acknowledgments

This work was performed using HPC resources from GENCI-IDRIS (Grant 2026-AD011014738R2). This work was supported by

Metric	SPA_all	acc23_all	SPA_ref	acc23_ref	SPA_reffree	acc23_reffree
Vertical_srcref	0.8347	0.628	0.8447	0.6356	—	—
Vertical_src	0.7735	0.5973	0.7816	0.6041	0.7166	0.5496
Gemini2_5-Pro	0.8283	0.583	0.8439	0.5896	0.7191	0.5369
GPT-4_1	0.8268	0.4989	0.8418	0.5014	0.7218	0.4817
TASER	0.8244	0.5831	0.8369	0.5873	0.7367	0.5534
Claude-4	0.8105	0.3524	0.823	0.3465	0.7235	0.3934
GEMBA-v2	0.7942	0.5644	0.8048	0.5744	0.7199	0.4944
CommandA	0.7883	0.3874	0.7989	0.3849	0.7137	0.4052
DeepSeek-V3	0.7784	0.3616	0.7881	0.3596	0.7105	0.3754
chrF	0.7714	0.5892	0.7714	0.5892	—	—
YiSi-1	0.7685	0.5982	0.7685	0.5982	—	—
Qwen3-235B	0.7668	0.3776	0.7768	0.3749	0.6967	0.3962
spBLEU	0.7647	0.5899	0.7647	0.5899	—	—
BERTScore	0.7539	0.5891	0.7539	0.5891	—	—
BLEU	0.7488	0.5783	0.7488	0.5783	—	—
MetricX-25	0.6983	0.5619	0.71	0.5707	0.6164	0.5007
COMET22	0.6944	0.5721	0.6944	0.5721	—	—

Table 2: System-level SPA scores and segment-level acc23 scores, across aggregation settings.

ANRT (Association Nationale de la Recherche et de la Technologie) under a CIFRE agreement.

References

- Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. [G-transformer for document-level machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3442–3455, Online. Association for Computational Linguistics.
- Daniel Deutsch, George Foster, and Markus Freitag. 2023a. [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.
- Daniel Deutsch, Juraj Juraska, Mara Finkelstein, and Markus Freitag. 2023b. [Training and meta-evaluating machine translation evaluation metrics at the paragraph level](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 996–1013, Singapore. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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.
- Sami Ul Haq, Chinonso Cynthia Osuji, Thiago Castro Ferreira, Brian Davis, and Sheila Castilho. 2025. [Long-context reference-based MT quality estimation](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 905–912, Suzhou, China. Association for Computational Linguistics.
- Juraj Juraska, Tobias Domhan, Mara Finkelstein, Tetsuji Nakagawa, Geza Kovacs, Daniel Deutsch, Pidong Wang, and Markus Freitag. 2025. [MetricX-25 and GemSpanEval: Google Translate submissions to the WMT25 evaluation shared task](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 957–968, Suzhou, China. Association for Computational Linguistics.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. [Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Ziqian Peng, Rachel Bawden, and François Yvon. 2025. [Investigating length issues in document-](#)

level machine translation. In *Proceedings of Machine Translation Summit XX: Volume 1*, pages 4–23, Geneva, Switzerland. European Association for Machine Translation.

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.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *Preprint*, arXiv:2403.05530.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Riviére, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.

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.