Quality-Aware Translation Models: Efficient Generation and Quality Estimation in a Single Model

Christian Tomani[*1,2,3], **David Vilar**[1], **Markus Freitag**[1], **Colin Cherry**[1], **Subhajit Naskar**[1], **Mara Finkelstein**[1], **Xavier Garcia**[1], **and Daniel Cremers**[2,3]

[1]Google
[2]Technical University of Munich
[3]Munich Center for Machine Learning

Abstract

Maximum-a-posteriori (MAP) decoding is the most widely used decoding strategy for neural machine translation (NMT) models. The underlying assumption is that model probability correlates well with human judgment, with better translations getting assigned a higher score by the model. However, research has shown that this assumption does not always hold, and generation quality can be improved by decoding to optimize a utility function backed by a metric or quality-estimation signal, as is done by Minimum Bayes Risk (MBR) or quality-aware decoding. The main disadvantage of these approaches is that they require an additional model to calculate the utility function during decoding, significantly increasing the computational cost. In this paper, we propose to make the NMT models themselves quality-aware by training them to estimate the quality of their own output. Using this approach for MBR decoding we can drastically reduce the size of the candidate list, resulting in a speed-up of *two-orders of magnitude*. When applying our method to MAP decoding we obtain quality gains similar or even superior to quality reranking approaches, but with the efficiency of single pass decoding.

1 Introduction

Most state-of-the-art models for natural language processing tasks are probabilistic, with the most frequent parameterization being based on neural networks. Once these models are trained, the prevailing decoding strategy for natural language generation is MAP decoding, i.e. select the hypothesis that maximizes the conditional probability given an input. As an exact maximization is computationally intractable, typically beam search or greedy decoding are used to approximate the search for the best hypothesis. Neural Machine translation is a prominent example of these types of models, where the system is trained to generate a sentence in a target language given a source sentence in another language. Nonetheless, Eikema and Aziz (2020) have demonstrated that MAP decoding methods may be suboptimal due to the presence of misaligned probability distributions. Moreover, NMT models often assign human translations lower probabilities than their own beam search outputs due to calibration issues (Ott et al., 2018; Freitag et al., 2020).

Eikema and Aziz (2020, 2022) applied MBR decoding for NMT models as an alternative generation approach. MBR decoding follows a self-consistency approach by sampling from the model distribution and giving preference to hypotheses that exhibit greater similarity to all other hypotheses. In contrast to MAP decoding, MBR decoding's objective is not centered on generating the translation with the highest estimated model probability, instead it selects the translation that exhibits the highest quality based on a utility metric. Subsequent research conducted by Freitag et al. (2022a) showed that MBR decoding with *neural* utility metrics leads to significant improvements over beam search decoding. However, MBR is computationally intensive, with a time complexity of $O(M^2)$ for a candidate list containing M samples, ideally $M = 100$ to $1\,000$ according to Freitag et al. (2022a). Note than when using neural metrics, each "computation step" in the quadratic complexity is itself computationally expensive, requiring a forward pass through a large neural network.

As an alternative to MBR decoding, we can use a quality-aware decoding strategy, generating a list of candidate translations and reranking them using a neural quality estimation (QE) metric that computes a quality score for the translation conditioned only on the source sentence. This method offers the advantage of being more efficient than MBR decoding, as its inference speed scales linearly with

[*]Work carried out while interning at Google. Correspondence to: Christian Tomani (christian.tomani@tum.de), David Vilar (vilar@google.com).

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15660–15679
August 11-16, 2024 ©2024 Association for Computational Linguistics

the number of candidate translations. A study conducted by Fernandes et al. (2022) showed that employing neural metrics for QE reranking exhibits comparable advantages to those seen with MBR decoding. However, this approach still demands the use of a separate, computationally expensive QE model to evaluate the quality of each candidate.

In our work, we propose a novel method that moves quality awareness inside the model itself, enabling us to guide the generation process towards higher-quality translations, and eliminating the need for an external QE model during decoding. Specifically, we investigate two key strategies: Quality-Aware Prompting, where we use quality prompts that explicitly encourage the generation of high-quality translations, and Quality-Aware Prediction, where we enable an NMT model to judge the quality of its own translations. Both strategies add special quality tokens to each NMT training example. The strategies differ only in whether the token is included in the source or the target sentence.

Our main scientific contributions are the introduction of quality-aware translation models, and their application for improved, more efficient decoding strategies. We analyze two use cases:

- We propose a novel reranking approach that eliminates the necessity for external QE models during decoding while maintaining the same level of translation quality. We can achieve similar or even superior results with single pass decoding, eliminating the need of a costly reranking step.
- We show that pre-filtering the candidate list according to the model's quality prediction can dramatically boost decoding performance of MBR, by up to two orders of magnitude, while increasing translation quality.

2 Related Work

Machine Translation (MT) metrics can be divided into two high-level categories: reference-based and reference-free, also known as quality estimation (QE) metrics. QE metrics compute a quality score for the MT output conditioned only on the source text. Reference-based metrics, on the other hand, require a human-generated reference translation to compare the MT output with. A multitude of reference-based metrics are available to evaluate the quality of translated content. Some metrics rely on lexical overlap, such as BLEU (Papineni et al., 2002), Meteor (Banerjee and Lavie, 2005) or ChrF (Popović, 2015). The WMT metrics task (Freitag et al., 2022b) demonstrated that the new generation of metrics – neural fine-tuned metrics like BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020a) – have significantly higher correlation with human judgement than traditional word overlap metrics. Consequently, we focus on neural fine-tuned metrics in this study. Quality estimation for MT began as confidence estimation (Blatz et al., 2004), but has recently shifted to embrace close kinship with reference-based metrics, with recent neural examples including OpenKiwi (Kepler et al., 2019), TransQuest (Ranasinghe et al., 2020), and COMET-QE (Rei et al., 2020b).

A prominent example of a reference-based neural metric is BLEURT (Sellam et al., 2020), and its extension to MetricX, which was the winning entry in the WMT22 metrics task (Freitag et al., 2022b). Building on this foundation, Juraska et al. (2023) recently introduced MetricX-QE, which we use as our primary QE metric in this work.

Reranking has a long history in translation, starting with Shen et al. (2004), where a discriminative model is learned to rank a candidate list to maximize a reference-based metric, with recent examples including Bhattacharyya et al. (2021) and Lee et al. (2021), who both train using BLEU. Our approach is closest to that of Fernandes et al. (2022), who rerank using various translation quality-estimation metrics, as opposed to training a special-purpose discriminative reranker. We differ from these works in that we do not need any external quality signal, which is instead provided by the NMT system itself.

While conventional MT research often relies on MAP decoding or generating k-best lists through beam search for MBR decoding, Eikema and Aziz (2020) proposed an approximation of MBR decoding via unbiased sampling. Their method aims to address the limitations of MAP decoding (Eikema and Aziz, 2020; Müller and Sennrich, 2021; Eikema and Aziz, 2022) by demonstrating that samples drawn from the NMT model align more faithfully with training data statistics when compared to beam search. Freitag et al. (2022a) showed that using neural metrics instead of overlap metrics results in significant improvements in translation quality. As a follow up, Freitag et al. (2023a) reported that the choice of sampling approach is important and epsilon sampling (Hewitt et al., 2022) is ideal for MBR decoding and reranking. Cheng and Vlachos (2023) introduced an orthogonal method to our proposed approach. They speed up MBR decoding by gradually increasing the number of samples used to estimate the utility while pruning hypotheses.

Our Quality-Aware Prompting approach extends a long line of methods where tagged training data has been used to control NMT output for different properties, including target language (Johnson et al., 2016), formality level (Yamagishi et al., 2016), politeness (Sennrich et al., 2016), domain (Kobus et al., 2017), gender (Vanmassenhove et al., 2018), syntactic structure (Shu et al., 2019), complexity (Agrawal and Carpuat, 2019) and reading level (Marchisio et al., 2019). The approaches closest to ours identify attributes related to translation quality, such as tagging back-translated examples to control away from synthetic data (Caswell et al., 2019), or tagging target-original examples to control toward natural-sounding output (Freitag et al., 2022c). To the best of our knowledge, we are the first to use quality estimation to tag training data, allowing NMT to discriminate between different translation qualities, and allowing us to prompt the model to generate high quality translations.

3 Preliminaries

We are given a NMT model $P_\Theta(y|x)$ which serves to estimate the probability of a hypothesis segment $y \in \mathcal{Y}$, given a source segment x. Here, Θ denotes the learned parameters of the neural network and $\mathcal{Y}$ the set of all possible hypotheses. There are two widely used approaches for generating the translations of a given sentence.

MAP decoding: This method involves searching for the most probable translation under $P_\Theta(y|x)$. However, determining the hypothesis with the maximum probability is computationally intractable due to the expansive and combinatorially complex search space $\mathcal{Y}$. Consequently, approximations like beam search (Graves, 2012; Sutskever et al., 2014) are often employed.

Sampling: In many applications we want to generate diverse hypotheses, e.g. in generative tasks where creativity is desired. In this case, instead of selecting the candidate with the highest probability (or an approximation thereof), we sample the output sentence following the probability distribution defined by the model. For NMT, this approach is used for generating a list of candidate translations, e.g. for MBR decoding. Specifically, epsilon sampling, as outlined by Hewitt et al. (2022), has emerged as the leading sampling technique for MBR. It was shown by Freitag et al. (2023a) to outperform other methods such as ancestral, top-k or nucleus sampling (Holtzman et al., 2020). Epsilon sampling prunes away any token with a probability lower than a given threshold ε, thereby guaranteeing that each token within a sample is allocated a fair probability mass. The likelihood of selecting token $y^{(\tau)}$ in the sampling process at time τ is governed by

$$P'_{\Theta,\varepsilon}(y^{(\tau)}|x, y^{(1:\tau-1)}) \sim \begin{cases} p_\tau^{\frac{1}{T}} & \text{if } p_\tau \geq \varepsilon \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

with

$$p_\tau = P_\Theta(y^{(\tau)}|x, y^{(1:\tau-1)}).$$

T denotes the sampling temperature. Epsilon sampling proves to be a highly effective strategy for the selective removal of unreliable, low-probability tokens.

3.1 External QE Reranking

External QE-Reranking involves generating a candidate list of size N through sampling and then reordering these samples, according to a quality estimation (QE) model. In our experiments, we employ MetricX-QE[1] (Juraska et al., 2023), a modification of MetricX, to compute a quality score $q = f(x, y)$. Here, f is parameterized by a transformer-based neural network and x and y denote the source segment and the translation, respectively.

3.2 Minimum Bayes Risk Decoding

In MBR decoding (Bickel and Doksum, 1977; Berger, 1985), given a set of candidate hypotheses $\mathcal{Y}$, the goal is to select the optimal hypothesis based on its expected utility concerning the distribution over human references within the space of all references Ω. This can be expressed mathematically as:

$$h^{\text{best}} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{r \in \Omega} u(y, r) P_{\text{human}}(r|x), \quad (2)$$

where $u(y, r)$ is a utility metric that is being used to gauge the quality of a candidate translation y with respect to a reference translation r.

[1] https://github.com/google-research/metricx

Since $P_{\text{human}}(r|x)$ remains unknown, we resort to sampling from the model instead, which relies on the assumption that the model provides a reliable approximation for the true underlying distribution over human translations. Furthermore, the integration over the vast space of all possible references Ω is computationally intractable. Therefore, MBR adopts a finite sample estimate by sampling a set of pseudo-references $\mathcal{Y}_{\text{model}}$ from $P_{\text{model}}(\cdot|x)$. This approximation can be expressed as:

$$h^{\text{MBR}} = \underset{y \in \mathcal{Y}}{\text{argmax}} \frac{1}{|\mathcal{Y}_{\text{model}}|} \sum_{r \in \mathcal{Y}_{\text{model}}} u(y, r), \quad (3)$$

where $\mathcal{Y} = \mathcal{Y}_{\text{model}}$, as the same set of model hypotheses serves both as the candidate list $\mathcal{Y}$ as well as the pseudo-reference list $\mathcal{Y}_{\text{model}}$. The computational time complexity of MBR decoding is $O(M^2)$ with M the size of the candidate list.

Note that this quadratic expression refers to *each sentence* to translate, i.e. for a corpus of size S, the total cost will be $O(S \cdot M^2)$. Also there is a hidden (multiplicative) constant, namely the cost of the computation of the utility function. For surface level metrics (e.g. BLEU, ChrF), this cost is negligible, but for neural metrics it involves computing the forward pass of a large neural network, therefore, any reduction in the number of metric computations has an important effect on the total running cost. In this paper, we focus on using BLEURT as utility function during MBR decoding.

4 Method

4.1 Quality-Aware Model

In contrast to External QE Reranking, which uses a separate QE model for assessing the quality of translations, we propose a novel method that integrates quality awareness directly into the translation model, making an independent QE model unnecessary during decoding. We present two approaches: in the first, we prompt the model to produce translations with a high QE score. In the second, the model is designed to provide a quality score alongside the translation. To achieve this, we initially assess the quality of samples within the training dataset, employing MetricX-QE. In the training phase we train our NMT model simultaneously on source and target samples, as well as their associated quality scores.

4.1.1 Assessing Quality

We first prepare the training dataset by computing the translation quality of each training sample and labelling each sentence pair with the corresponding quality score. Since the distribution over translation qualities is not necessarily uniform, we discretize the scores via equal mass binning into B bins, which are then mapped to single tokens of the vocabulary of the translation model. This binning strategy leads to a balanced training set w.r.t. quality scores. To achieve this, we consider the set of quality scores Q from all samples in the training dataset, denoted as $q_1, q_2, \ldots, q_N$, in order to determine bin boundaries or cut-off points $c_1, c_2, \ldots, c_{B+1}$ such that $\forall 1 \leq i \leq B$

$$|\{x \mid q \in Q,\ c_i \leq q < c_{i+1}\}| \approx \frac{N}{B}. \quad (4)$$

In this way each bin contains approximately the same number of samples. This is to avoid sample imbalances per bin when training the model, which in preliminary experiments proved to be important as to not bias the model towards the most frequent label. Next, each bin is assigned a bin identifier b, which is represented by a single token in the model vocabulary. E.g. if we define 10 bins (the actual number we used in our experiments[2]), we can just use the numbers between 0 and 9. Lastly, the quality score is inserted into the data pipeline during training to associate the source and target pair with the respective bin identifier b. To mark the token b as a QE value, we employ a special string format by surrounding b with square brackets: [b]. In the following we outline two distinct methods for integrating our quality score string into the model.

4.1.2 Quality-Aware Prompting (QA Prompting)

During the training process, we append the quality score string [b] to the source segment. This enables the model to associate the discretized quality score b with a translation example with the same level of quality. As the quality token is attached to the *input* to the system, it provides us with the capability to prompt for different quality levels during decoding. I.e. at translation time we can append the token corresponding to the highest quality level to prompt the system to generate a sentence of the highest quality.

[2]For an exploration on other number of bins see Appendix D.

4.1.3 Quality-Aware Prediction (QA Prediction)

Instead of prompting the model explicitly for high-quality translations, an alternative approach is to design a model that jointly predicts a hypothesis and a quality score. This design allows us to leverage our translation model to also function as a QE model. To achieve this, we append the quality score string `[`b`]` to the target sentence during training, enabling the model to learn to predict the quality during inference.

If using a reranking approach these quality scores can be used to reorder samples within a candidate list. However, due to our use of discretized bins, it is possible that the model predicts the same scores for multiple samples. To address this, we also consider the log probabilities z associated with the bin identifier tokens. This additional information allows for a more precise reranking of samples in the candidate list. Specifically, we sort samples with respect to b as the primary sorting criterion and use the log probabilities z as the secondary criterion. Given a candidate list of size M, we sort the samples into $y_1, y_2, \ldots, y_M$ with corresponding discretized quality scores $b_1, b_2, \ldots, b_M$ and log probabilities $z_1, z_2, \ldots, z_M$ such that

$$\begin{aligned} &\forall 1 \leq i < j \leq M : \\ &\quad (b_i > b_j) \vee \ (b_i = b_j \wedge z_i > z_j) \end{aligned} \tag{5}$$

With the sorted candidate list in hand, we can proceed by either selecting the top-ranked sample as our final translation or further processing the top-k samples in the context of MBR decoding.

5 Experimental Setup

Model: Our model is a transformer consisting of 6 encoder and 6 decoder layers, 16 attention heads with a dimension of 128, a hidden dimension of 8192, and a model dimension of 1024, resulting in 551M parameters. We employ a shared vocabulary of 32k tokens and impose a maximum sentence length of 128 tokens. We utilize GELUs with gated activation functions. The baseline system is trained on the entire available dataset. All models are trained on TPUs (v3) until they reach convergence. The MetricX-QE model used for quality estimation is a transformer based model with a total of 2B parameters, as described in Juraska et al. (2023). To assess the applicability of our approach to LLMs, we also fine-tune and evaluate a quality-aware LLM, with results in Appendix C.

Data: We choose two high-resource language pairs from the WMT 2022 shared task (Kocmi et al., 2022): English to German (en $\rightarrow$ de) and in the Appendix we additionally show results for English to Japanese (en $\rightarrow$ ja). While we filter out sentences exceeding 128 tokens, we perform no further data filtering or preprocessing. The training dataset for (en $\rightarrow$ de) comprises 295.8M samples, while the (en $\rightarrow$ ja) dataset consists of 33.9M samples. Our evaluation is based on the WMT 2022 general translation task test sets.

Given that we have available MetricX-QE scores for the whole training data, one natural question to ask is what would happen if we limit the training data to the only the best quality training sentence pairs. Peter et al. (2023) showed that this is indeed a very effective way to reduce the training data size, while at the same time improving translation performance. We also report experiments on this data condition, which represents a stronger baseline with which to compare our methods. For these experiments we follow (Peter et al., 2023) and keep only the top 50% scoring sentence pairs.

Metrics: We use neural metrics for evaluation, with a focus on COMET (Rei et al., 2020a) (COMET 22 version). We also report MetricX scores, but as MetricX-QE is based on it and our methods directly optimize this metric, there is the danger of overfitting for this particular metric (Amrhein and Sennrich, 2022; Yan et al., 2023). In addition, for selected experiments we conduct expert-based human evaluations using MQM (Freitag et al., 2021), a human evaluation scheme centered on marking errors present in the translations. Details can be found in Appendix I. For completeness we also report BLEU scores in Appendix B, but we do not analyze them here, as it has been shown that they do not correlate well with human judgement of NMT systems (Freitag et al., 2022b).

6 Results

6.1 Control Experiments

6.1.1 Can NMT Models Learn to Estimate Quality?

In a first experiment, we train a model using the proposed Quality-Aware Prediction method (§ 4.1.3). We then use this model to force-decode the translations provided in the WMT23 QE shared task (Blain et al., 2023) and, once the model has reached the end of the translation, we select the quality la-

System	Pearson ρ sys	seg	acc-t
XCOMET-Ensemble	0.993	0.695	0.603
MetricX	0.977	0.585	0.602
MetricX-QE	0.969	0.626	0.596
QA Prediction (Ours)	0.932	0.412	0.524
Model PPL	0.722	0.213	0.504
BLEU	0.916	0.192	0.520
ChrF	0.866	0.232	0.519

Table 1: Correlation of different QE metrics with human judgement on the WMT23 en → de task.

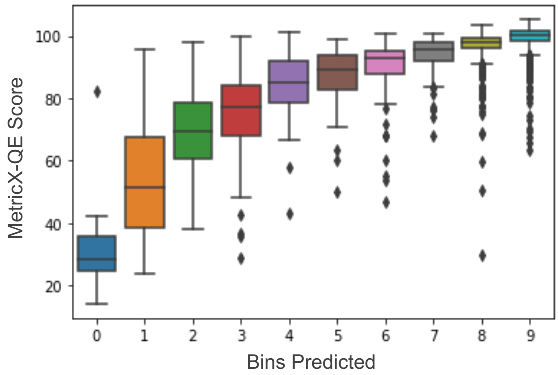


Figure 1: Alignment between predicted quality scores from the QA Prediction model and actual MetricX-QE scores of translations in the en → de test dataset. The boxplots show the distribution of actual scores across all samples assigned to each bin. The median ground truth quality score increases steadily in line with the predicted bins.

bel that gets assigned the highest probability by the model. As many different segments will be assigned the same discrete quality label, we also use model probability to break ties when calculating correlations (see Section 4.1.3). In this way we can check if our translation model can double as a QE model. We evaluate this approach using the same metrics as in the shared task. The results are shown in Table 1, together with some other representative metrics. The top block in Table 1 shows the top-scoring metrics in the shared task.[3] As expected, we see that our model does not achieve the performance of separate dedicated models (and specifically not that of the MetricX-QE metric it is based on), but it still outperforms the traditional BLEU and ChrF metrics, and these have access to the reference translations.

One could also consider using perplexity as an indicator of translation quality, thus eliminating the need of generating an explicit quality tag. Table 1 shows that this value alone is not enough to differentiate the quality of the translations.

In summary, our quality-aware method can predict translation quality better than perplexity scores and older string-matching metrics, but not as well as dedicated neural models. However, quality estimation is not the main goal of our work, rather a tool towards improving translation quality and efficiency, as we will show in Sections 6.2 and 6.3.

6.1.2 Can NMT Models Distinguish the Quality of Their Own Outputs?

Next we turn to the question of whether the system is able to judge the quality of *its own* translations. In order to test this, we translated our dev dataset with quality label predictions and then computed the actual MetricX-QE scores which the model is trained to predict. We find that the predicted quality score bins are well aligned with these "ground truth" scores[4], as demonstrated in Figure 1, where we show the distribution of ground truth scores (non discretized) across the predicted bins. In particular we note that the system is able to accurately detect bad quality translations, which will become important for the translation use case.

6.1.3 Can We Control Translation Quality?

As a last control experiment we train a QA Prompting model following Section 4.1.2 and generate translations using different quality labels. I.e. we are asking the model to generate good translations (high scores in the quality labels) as well as *bad* translations (low scores). The results are shown in Figure 2. It can be seen that the system is indeed able to adjust the quality of the translation output according to the given prompting. Example translation outputs are shown in Appendix F.

6.2 Translation Performance

Having confirmed that our model is indeed able to distinguish quality levels, we explore how to use this property to enhance the overall output quality of a NMT system. For this experiment, we evaluate both our Quality-Aware Prediction model and Quality-Aware Prompting model.

For the Quality-Aware Prediction model, we extract N=1024 hypotheses from the model via ep-

[3]Note that evaluation metrics and QE metrics were evaluated simultaneously.

[4]"Ground truth" in this context as these are the scores that the model was trained to predict. They are not necessarily ground truth for true quality measurement.

Method	Data	MetricX	COMET	MQM ↓
Baseline	Full Corpus	80.2	85.8	1.81
QA Prompting (Ours)	Full Corpus	**82.3**	**87.1***	**1.43***
QA Prediction (Ours)	Full Corpus	82.0	86.5*	2.07
External QE-Reranking	Full Corpus	83.3	86.9*	1.50*
Baseline	Filtered	81.8	87.0	–
QA Prompting (Ours)	Filtered	**82.6**	**87.3***	–
QA Prediction (Ours)	Filtered	82.5	86.7	–
External QE-Reranking	Filtered	83.7	86.9	–

Table 2: Comparison between quality-aware models and baseline models on the full and filtered training datasets. The quality-aware methods outperform the baseline model and perform similar to reranking without requiring an additional MetricX-QE model during decoding. * denotes statistically significant (pairwise permutation test (Koehn, 2004) with p=0.05) differences compared to the baseline with $p < 0.05$. No significance is computed for MetricX due to the methods optimizing this metric directly.

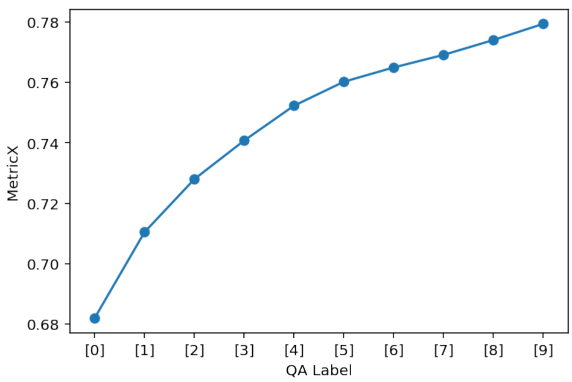


Figure 2: Translation quality dependent on the QA label used for prompting. Higher values in the label prompt the system to generate better translations.

silon sampling and retrieve the quality score string from each hypothesis. Subsequently, we rank these hypotheses using Equation 5 and select the highest-ranked sample as our final translation output. In the case of Quality-Aware Prompting, we directly retrieve the final translation through MAP-decoding by appending the highest quality score string as a suffix to the source sentence, i.e. we "ask the model" to produce high quality outputs. We compare these two approaches with an identical baseline NMT model that only differs by not using any quality score during training.

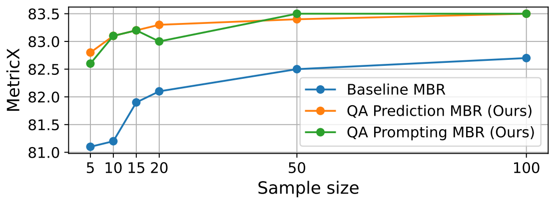
Table 2 shows that both quality-aware methods surpass the baseline model in terms of MetricX and COMET, with QA Prompting showing better results on both metrics. As expected, all the methods achieve big improvements in MetricX, as it is closely related to the MetricX-QE metric that we are directly optimizing. In fact, external QE-Reranking achieves the best MetricX score by a wide margin, however COMET puts QA Prompting on-par with QE-Reranking. Even though an improvement of COMET of 0.2-0.4 seems small, Lo et al. (2023) use human judgment to verify differences in COMET scores and conclude that even small differences in COMET scores can mean large quality improvements.

The human evaluation mostly confirms the trends shown by COMET. The MQM scores can be interpreted as the average number of errors in translation, i.e. lower numbers are better. The human evaluation shows that QA Prompting does indeed produce significantly better translations than the baseline systems, and it even outperforms the external QE-Reranking approach. This is specially noteworthy given that QA Prompting is a single pass approach, with no additional cost over the MAP decoding baseline, whereas QE-Reranking rescores a 1024 candidate list with an external (expensive) QE model. QA Prediction on the other hand does not outperform the baseline approach and in fact shows a degradation in performance.

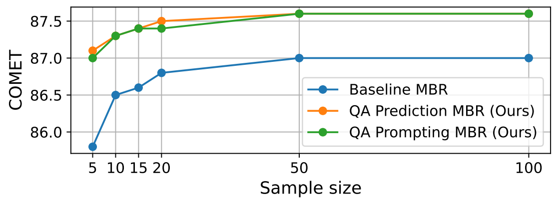
When compared to the stronger baseline with training data prefiltered with MetricX-QE scores, we see that QA Prompting is still able to obtain a slight improvement over both the baseline and QE-Reranking, which is still statistically significant. QA Prediction is not effective in this setting either.

6.3 MBR decoding

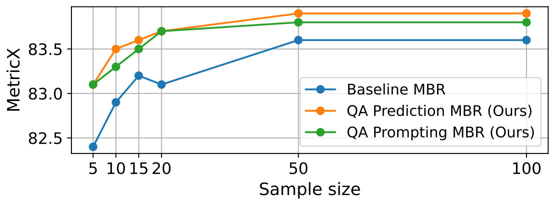
Next we turn our attention to improving the performance of MBR decoding. This depends heavily



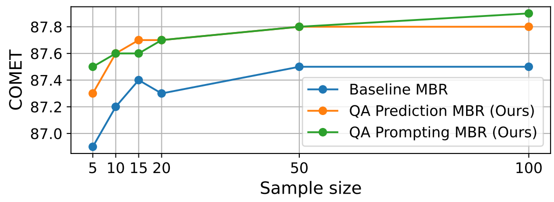
(a) MetricX- Full data



(b) COMET- Full data



(c) MetricX- Filtered data



(d) COMET- Filtered data

Figure 3: Performance of quality-aware approaches (QA Prediction and QA Prompting) compared to baseline MBR decoding across various candidate list sizes. MBR decoding with quality-aware models consistently outperforms baseline MBR decoding across candidate list sizes. The quality-aware approaches can achieve the same level of performance as baseline approaches while reducing the required utility function computations by up to two orders of magnitude.

on the length of the candidate list M, and Freitag et al. (2023a) showed that a large candidate size of several hundred candidates is needed for achieving good translation performance. However, this property makes MBR decoding computationally expensive as the utility function computations grow quadratically with the candidate list size M, see Section 3.2. Our investigation seeks to understand how improved candidate quality influences the performance and efficiency of MBR decoding.

For baseline MBR decoding we use epsilon sampling to generate a candidate list of M samples, as we do for the Quality-Aware Prompting approach. For Quality-Aware Prediction, in line with the previous section, we employ our Quality-Aware Prediction approach to sample $N = 1024$ hypotheses with quality score strings. Then we rank all samples and select the top M samples as our candidate list. Subsequently, we apply MBR decoding to the gathered candidate lists. We used BLEURT as utility function as a proxy for MetricX, due to the high computational cost of this last metric.

In Figure 3 we show the performance of our quality-aware approaches compared to baseline MBR decoding across various candidate list sizes M. Our proposed methods consistently outperform baseline MBR decoding in terms of MetricX and COMET scores, irrespective of the candidate list size. Notably, our quality-aware approaches combined with MBR decoding require substantially fewer candidates to achieve equivalent performance to baseline MBR decoding. For example, in models trained on the entire dataset, the Quality-Aware Prompting and Quality-Aware Prediction approaches obtain COMET scores of 86.9 and 87.0, respectively, with a candidate list size of 5 (needing just 20 utility function computations[5] per sentence). In contrast, baseline MBR decoding plateaus at 87 COMET starting at a candidate size of 50 (requiring 2450 utility function computations per sentence). This translates to a 100-fold increase in computations for the baseline model to achieve a similar score. We also note that the baseline model with a candidate list size of 1024 achieves a MetricX score of 82.9 and a COMET score of 87.0. This indicates that our approach, with a candidate list size of 50, outperforms a baseline model with even 1024 samples. For our experiments on filtered data we observe a similar improvement in the quality-aware models when compared to baseline MBR decoding.

Table 3 shows the translation performance of the MBR systems, including human evaluation with MQM, with a candidate size $M = 50$. A first observation is that baseline MBR decoding significantly outperforms the baseline system, coming close to the external QE reranking approach. When we combine MBR with quality-aware models, we

[5]The number of computations is $M \times (M - 1)$ as a hypothesis is not evaluated against itself.

Data	#Candidates	Method	MetricX	COMET	MQM
Full Corpus		Baseline (w/o MBR)	80.2	85.8	1.81
		External QE-Reranking	83.3	86.9*	1.50*
	$M = 50$	MBR Baseline	82.5	86.8*	1.41*
		MBR QA Prompting (Ours)	**83.5**	87.4*†	**1.36***†
		MBR QA Prediction (Ours)	83.4	**87.5***†	1.45*†
	$M = 5$	MBR Baseline	81.1	85.7	–
		MBR QA Prompting (Ours)	82.6	86.9*	–
		MBR QA Prediction (Ours)	**82.8**	**87.0***	–
Filtered		Baseline (w/o MBR)	81.8	87.0	–
		External QE-Reranking	83.7	86.9	–
	$M = 50$	MBR Baseline	83.6	87.4	–
		MBR QA Prompting (Ours)	83.8	**87.7***†	–
		MBR QA Prediction (Ours)	**83.9**	**87.7***†	–
	$M = 5$	MBR Baseline	82.4	86.7	–
		MBR QA Prompting (Ours)	**83.1**	**87.3***	–
		MBR QA Prediction (Ours)	**83.1**	87.2*	–

Table 3: MBR results with quality-aware decoding approaches. The symbol * denotes statistically significant differences compared to the baseline with $p < 0.05$, † denotes statistically significant differences compared to the *MBR baseline with* $M = 50$. No significance is computed for MetricX due to the methods optimizing this metric directly.

again obtain a significant improvement when compared the the MBR baseline, with the MQM score dropping from 1.81 to 1.36 for QA Prompting. In this condition, the QA Prediction approach does perform satisfactorily, and there is no significant difference when compared to QA Prompting.

When reducing the candidate list size to 5, we can see that the translation performance drops only slightly (e.g. only 0.3 COMET for the Filtered QA Prompting approach), but the number of utility function computations is drastically reduced from 2450 to 20, two orders of magnitude. This is not the case for baseline MBR, which actually performs worse than the non-MBR baseline with this reduced candidate size.

Note that each computation in MBR decoding involves evaluating a neural metric, an expensive operation. Thus, reducing the absolute number of computation has a direct effect on running time of the full MBR pipeline.

7 Conclusion

This paper introduces a novel approach to enhance NMT by making the models quality-aware. Our approach addresses the issue of misalignment between outputs generated via MAP decoding and human judgment. We achieve this by training NMT models to assess the quality of their own translations, effectively circumventing the limitations of conventional decoding methods. As a result this new approach yields improvements similar or superior to QE-reranking approaches, but with the efficiency of MAP-decoding, i.e. with single-pass decoding (for QA Prompting, the best performing method). QE-reranking in contrast needs a sampling step followed by a reranking step using an external, computationally expensive model.

By leveraging the model's quality signal internally during MBR decoding, not only does translation quality further improve, but computational efficiency is also dramatically enhanced, reducing inference time by two orders of magnitude. This improvement comes from a drastic reduction in the necessary size of the candidate list needed for producing good quality translations.

Overall this research opens up exciting possibilities for advancing the field of NMT, offering both improved translation quality and faster processing speeds without the need for additional, computationally intensive models.

Limitations and Risks

Our work is currently limited to two language pairs. We leave it for future work to explore the applicability of the proposed approach in multilingual as well as low-resource settings. Furthermore, especially in low-resource languages where there is less training data, overfitting to the QE metric used for training could be a limitation.

We acknowledge that, although we are using different metrics for optimizing our method (MetricX-QE) and evaluating it (COMET), both are neural metrics trained on the same data from the WMT evaluations. There might be biases that should be taken into account when considering the method. Nevertheless, neural metrics have proven to be the most reliable evaluation metrics for machine translation up to this date.

A potential risk of our method might be that training it is more resource-intensive than simple models, and thus might increase the quality difference with respect to low-resource languages, since they are unlikely to be allocated as many resources as the high-resource languages.

References

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1549–1564, Hong Kong, China. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2022. Identifying weaknesses in machine translation metrics through minimum Bayes risk decoding: A case study for COMET. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1125–1141, Online only. Association for Computational Linguistics.

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.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

James O Berger. 1985. *Statistical decision theory and Bayesian analysis; 2nd ed.* Springer series in statistics. Springer, New York.

Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum. 2021. Energy-based reranking: Improving neural machine translation using energy-based models. 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 4528–4537, Online. Association for Computational Linguistics.

Peter J Bickel and Kjell A Doksum. 1977. Mathematical statistics: Basic ideas and selected topics. *Holder-Day Series in Probability and Statistics, Holder-Day, San Francisco*.

Frederic Blain, Chrysoula Zerva, Ricardo Ribeiro, Nuno M. Guerreiro, Diptesh Kanojia, José G. C. de Souza, Beatriz Silva, Tânia Vaz, Yan Jingxuan, Fatemeh Azadi, Constantin Orasan, and André Martins. 2023. Findings of the WMT 2023 shared task on quality estimation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 629–653, Singapore. Association for Computational Linguistics.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum bayes risk decoding with confidence-based pruning.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

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, Behrooz Ghorbani, and Patrick Fernandes. 2023a. Epsilon sampling rocks: Investigating sampling strategies for minimum bayes risk decoding for machine translation. *arXiv preprint arXiv:2305.09860*.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022a. High quality rather than high model probability: Minimum bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

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. 2023b. Results of wmt23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022b. 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, David Vilar, David Grangier, Colin Cherry, and George Foster. 2022c. A natural diet: Towards improving naturalness of machine translation output. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3340–3353, Dublin, Ireland. Association for Computational Linguistics.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Vi'egas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *CoRR*, abs/1611.04558.

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.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

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, Vilém Zouhar, Christian Federmann, and Matt Post. 2024. Navigating the metrics maze: Reconciling score magnitudes and accuracies.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021. Discriminative reranking for neural 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 7250–7264, Online. Association for Computational Linguistics.

Chi-kiu Lo, Rebecca Knowles, and Cyril Goutte. 2023. Beyond correlation: Making sense of the score differences of new MT evaluation metrics. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 186–199, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Kelly Marchisio, Jialiang Guo, Cheng-I Lai, and Philipp Koehn. 2019. Controlling the reading level of machine translation output. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 193–203, Dublin, Ireland. European Association for Machine Translation.

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.

Mathias Müller and Rico Sennrich. 2021. Understanding the properties of minimum Bayes risk decoding in neural 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 259–272, Online. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

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.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. 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, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827, Florence, Italy. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium. Association for Computational Linguistics.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

Yiming Yan, Tao Wang, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Mingxuan Wang. 2023. BLEURT has universal translations: An analysis of automatic metrics by minimum risk training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5428–5443, Toronto, Canada. Association for Computational Linguistics.

Vilém Zouhar and Ondřej Bojar. 2024. Quality and quantity of machine translation references for automated metrics.

Appendices

A Additional Results for Language Pair: English to Japanese

Beyond the results highlighted in the main text, we present findings for an additional language pair, specifically, English to Japanese ($en \rightarrow ja$). Our experimental setup mirrors that of the English to German translation task, with the exception that we employ a BLEURT score threshold of 60 for training data filtering. Our results closely resemble those obtained for the English to German datasets.

Exploring the $en \rightarrow ja$ scenario, we compare the performance of quality-aware models against baseline models using both the complete and filtered training datasets. Evaluation metrics include MetricX, BLEURT, COMET, and BLEU. Notably, the quality-aware methods achieve consistently better results than the baseline model, all without the need for an additional BLEURT model during decoding (Tab. 4).

In Fig. 4 we also examine the effectiveness of the proposed approaches in contrast to baseline MBR decoding across various candidate list sizes for $en \rightarrow ja$. Our findings demonstrate that MBR decoding with quality-aware models consistently surpasses baseline MBR decoding across different candidate list sizes. When trained on the entire dataset our methods achieve MetricX and COMET scores with only 5 samples that are clearly better than the baseline model regardless of the sample size. For this language pair we can dramatically increase the performance while at the same time requiring more than two orders of magnitude less computation time. When trained on the full dataset, our methods exhibit MetricX and COMET scores that outperform the baseline model for all candidate list sizes with a noticeable advantage, even when considering a limited number of samples. Importantly, this improved performance is accompanied by a similar reduction in computation time as for $en \rightarrow de$, with our approach requiring two orders of magnitude less computational resources for this language pair.

B BLEU scores

For completeness we include in Tables 5 and 6 a copy of the main results of our paper, including BLEU scores. However, as demonstrated in numerous previous work (Mathur et al., 2020; Freitag et al., 2022b; Lo et al., 2023; Freitag et al., 2023b; Kocmi et al., 2024; Zouhar and Bojar, 2024, inter alia) BLEU scores are not longer representative of translation quality with current systems, hence we do not base the analysis of our results on them. Even more, Tables 5 and 6 provide further evidence of this fact, as BLEU scores show *negative correlation* with our available human evaluations.

C Quality-Aware LLMs

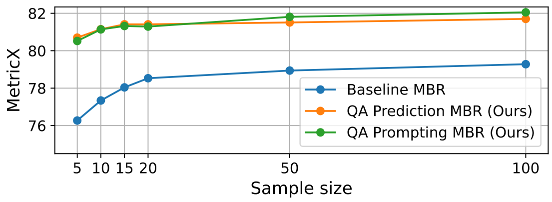
In this section we explore the potential of enhancing LLMs with quality awareness through our proposed method. Specifically, we employ our most efficient approach, Quality-Aware Prompting, to finetune PaLM-2 Bison (Anil et al., 2023). To this end, we finetune the pretrained model for 10k steps with the same data as in the main text (en-de) using our QA Prompting approach. As baseline we also finetune the LLM with identical configurations on the same data, but without incorporating any quality signal. In Table 7, we observe that QA Prompting substantially outperforms standard finetuning by 1.3 BLEURT points. Our results suggest that substantial performance improvements can be achieved with minimal data and steps. This opens the door to the possibility of utilizing more costly QE methods or even human evaluations in the future to curate finetuning datasets and align models directly with human preferences.

D Sensitivity Analysis of Number of Bins

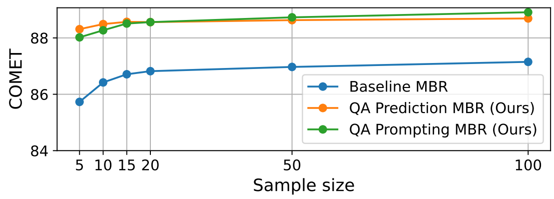
In this section, we examine the selection of the number of bins for discretizing the quality score. To do this, we employ our Quality-Aware Prediction approach and train five models with varying numbers of bins, specifically 2, 3, 5, 10, and 20. We then evaluate their performance on MetricX, BLEURT, COMET, and BLEU. Figure 5 illustrates that increasing the number of bins yields improvements on our quality metrics, in particular in the range of 2 to 5 bins. This aligns with our overarching concept of instilling quality awareness in the model, as a higher number of bins allows for a finer distinction between quality levels within the model, which is evident in our findings.

E Influence of Bin Identifier Choice

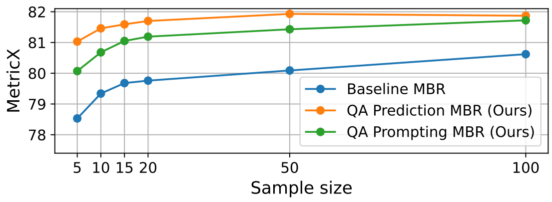
Tokens in a language model are mapped to an embedding, representing a specific meaning or relation to other tokens within the embedding space. In this context, we aim to explore whether mixing token meanings from the primary translation task



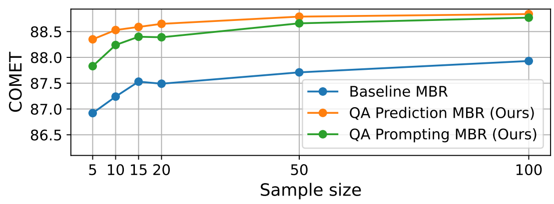
(a) MetricX- Full data



(b) COMET- Full data



(a) MetricX- Filtered data



(b) COMET- Filtered data

Figure 4: Performance of quality-aware approaches (Quality-Aware Prediction and Quality-Aware Prompting) compared to baseline MBR decoding across various candidate list sizes for $en \rightarrow ja$. MBR decoding with quality-aware models consistently outperforms baseline MBR decoding across candidate list sizes. The quality-aware approaches can achieve the same level of performance as baseline approaches while reducing the required utility function computations by one to two orders of magnitude.

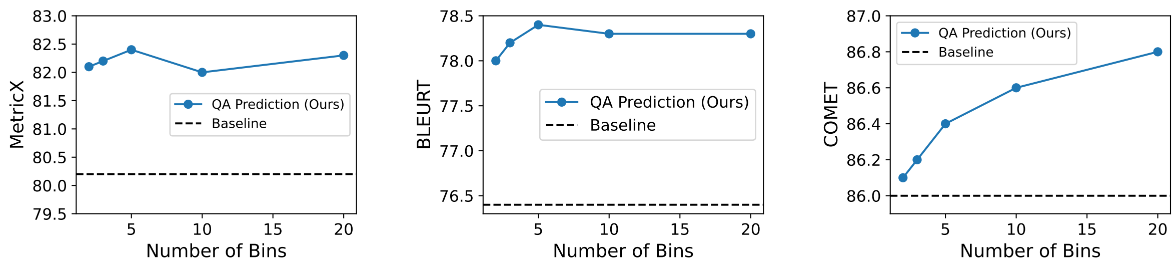


Figure 5: Sensitivity concerning performance of the Quality-Aware Prediction approach w.r.t. the number of quality score bins. Increasing the number of quality score bins yields generally improvements on our quality metrics, specifically in the range of 2 to 5 bins.

Method	Data	MetricX	COMET
Baseline	Full Corpus	76.3	85.7
Quality-Aware Prompting (Ours)	Full Corpus	80.3	**87.8**
Quality-Aware Prediction (Ours)	Full Corpus	**80.9**	87.7
Baseline	Filtered	77.8	86.2
Quality-Aware Prompting (Ours)	Filtered	**80.0**	**87.7**
Quality-Aware Prediction (Ours)	Filtered	79.6	87.5

Table 4: Comparison $en \rightarrow ja$ between quality-aware models (Quality-Aware Prediction and Quality-Aware Prompting) and baseline models on the full and filtered training dataset evaluated on MetricX, BLEURT, COMET and BLEU. The quality-aware methods outperform the baseline model without the need of an additional BLEURT model during decoding.

with the scoring task has a detrimental effect on either one of them. To investigate this, we assess the translation quality of the baseline model without the inclusion of quality score strings, as well as our proposed Quality-Aware Prediction approach.

Firstly, we evaluate the translation quality between the baseline and our model. We find that our model performs equivalently to the baseline when treating it as a basic translation model and disregarding the quality score strings appended to our model's hypotheses. This suggests that the model successfully disentangles a token's actual meaning in the text from its role as a quality score bin identifier.

Furthermore, we investigate whether the use of tokens that appear frequently in the training data corpus, such as numbers and letters, as opposed to tokens that are the least likely to appear in the training data, has an adverse impact on quality scoring. To investigate this, we train our model with 5 bins but employ different bin identifiers, including numbers $[0, 1, 2, 3, 4]$, letters $[a, b, c, d, e]$, and the 5 least frequently occurring tokens from the vocabulary. Tab. 8 shows that varying the choice of bin identifiers demonstrates a high degree of robustness.

We further investigate possible cases where the model does not produce a quality score at all. However, we find that the model is able to learn to predict a quality score as early as after 500 steps. We only observe a few corner cases where the model does not predict a quality score when we use epsilon sampling and sample 1000 sentences. In the case where no quality score is produced we assign the lowest bin identifier to a sample.

F Example Translations for QA Prompting

In Table 9 we provide different translation examples when varying the label for quality prompting (see Section 4.1.2). It can be seen, that the prompting has a critical influence in the quality of the translation output.

G Combining Quality-Aware Prompting and Quality-Aware Prediction

Throughout our experiments we frame Quality-Aware Prompting and Quality-Aware Prediction as two separate approaches. One might wonder whether both approaches are orthogonal to each other and might benefit each other when combined. To this end we add the quality score string to the source and the target sentence. To avoid that the model just learns copying the quality score string from the input to the output, we choose a multiple of the bin number from the prompting approach for the prediction approach. This way we make sure that the model is required to first of all learn to provide a high quality translation when prompted for it and learns to fine grained distinguish the quality in the quaility prediction tasks in the output. However, we observe that the combination of both approaches results in inferior performance compared to each approach individually, regardless of whether the full or filtered dataset is employed for training (Tab. 10). We hypothesize that this may be attributed to the model becoming excessively fixated on predicting the score in the output based on the input score, potentially leading to overfitting, where the prediction score becomes overly conditioned on the prompting score.

Method	Data	MetricX	COMET	BLEU	MQM ↓
Baseline	Full Corpus	80.2	85.8	35.4	1.81
QA Prompting (Ours)	Full Corpus	**82.3**	**87.1***	**36.6**	**1.43***
QA Prediction (Ours)	Full Corpus	82.0	86.5*	27.1	2.07
External QE-Reranking	Full Corpus	83.3	86.9*	28.1	1.50*
Baseline	Filtered	81.8	87.0	36.2	–
QA Prompting (Ours)	Filtered	**82.6**	**87.3***	**36.7**	–
QA Prediction (Ours)	Filtered	82.5	86.7	28.1	–
External QE-Reranking	Filtered	83.7	86.9	27.8	–

Table 5: Copy of Table 2 including BLEU scores. We highlight in red the cases where BLEU shows *inverse correlation* with human judgements when comparing systems to the baseline.

Data	#Candidates	Method	MetricX	COMET	BLEU	MQM
Full Corpus		Baseline (w/o MBR)	80.2	85.8	**35.4**	1.81
		External QE-Reranking	83.3	86.9*	28.1	1.50*
	$M = 50$	MBR Baseline	82.5	86.8*	30.8	1.41*
		MBR QA Prompting (Ours)	**83.5**	87.4*†	31.9	**1.36***†
		MBR QA Prediction (Ours)	83.4	**87.5***†	31.1	1.45*†
	$M = 5$	MBR Baseline	81.1	85.7	29.6	–
		MBR QA Prompting (Ours)	82.6	86.9*	31.3	–
		MBR QA Prediction (Ours)	**82.8**	**87.0***	29.1	–
Filtered		Baseline (w/o MBR)	81.8	87.0	**36.2**	–
		External QE-Reranking	83.7	86.9	27.8	–
	$M = 50$	MBR Baseline	83.6	87.4	31.1	–
		MBR QA Prompting (Ours)	83.8	**87.7***†	33.0	–
		MBR QA Prediction (Ours)	**83.9**	**87.7***†	32.0	–
	$M = 5$	MBR Baseline	82.4	86.7	**30.6**	–
		MBR QA Prompting (Ours)	**83.1**	**87.3***	32.4	–
		MBR QA Prediction (Ours)	**83.1**	87.2*	30.2	–

Table 6: Copy of Table 3 including BLEU scores. We highlight in red the cases where BLEU shows *inverse correlation* with human judgements when comparing systems to the baseline.

Method (LLM)	BLEURT
Baseline	77.9
QA Prompting (Ours)	**79.2**

Table 7: Quality-Aware LLMs: The performance of LLMs can be enhanced via finetuning a pretrained LLM with Quality-Aware Prompting compared to standard finetuning.

H Details of Quality-Aware Approaches

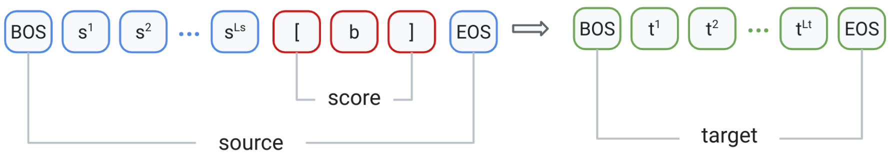
In the context of quality-aware translation, two approaches are explored. In Quality-Aware Prompting (QA Prompting), a quality score is appended to the source segment during training, allowing the model to associate quality score strings with examples of translation exhibiting the same level of quality (Figure 6a). Since this quality token appears in the input, it allows for direct prompting of high-quality translations during decoding. Conversely, Quality-Aware Prediction (QA Prediction) involves training a model that predicts both a hypothesis and a quality score concurrently. This approach transforms the translation model into a Quality Estimation (QE) model by appending the quality score string to the target sentence during training, enabling the model to predict quality during inference (Figure 6b).

I MQM Human Evaluations

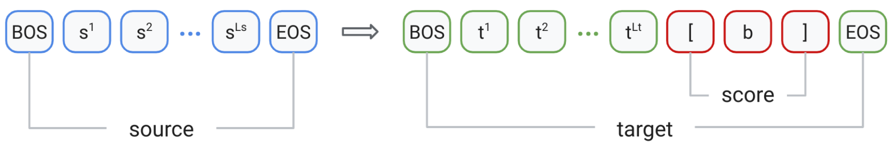
For the MQM evaluation we use the same annotator guidelines as specified in (Freitag et al., 2021). The annotators were compensated fairly and did not have to disclose any personal information during the annotation process. All of the test sets used in this study are publicly available.

Bin identifiers	MetricX	COMET
Baseline	80.2	86.0
Numbers	82.4	86.4
Letters	82.5	86.6
Least likely tokens	82.6	86.6

Table 8: Sensitivity of our Quality-Aware Prediction approach with respect to bin identifiers on the entire training dataset. For all experiments we set the number of bins to 5 and vary the choice of bin identifiers. Our approach is largely insensitive to bin identifiers.



(a) Quality-Aware Prompting



(b) Quality-Aware Prediction

Figure 6: Visualization of the Quality-Aware Prompting and Quality-Aware Prediction approaches.

Source		Needless to say, it was chaos.
QA Label	[0]	Es war ein sehr schöner Tag. *Translation: "It was a very nice day."*
	[5]	Es war ein Chaos. *Missing translation for "Needless to say".*
	[9]	Unnötig zu sagen, es war Chaos. *Nearly perfect translation.*
Reference		Es war natürlich ein Chaos.

Source		The total number of vaccines injected in the country reached 109,990,742 doses.
QA Label	[0]	Die Gesamtzahl der injizierten Impfstoffe in der Türkei erreichte 109.990.742 Dosen. *"in der Türkei" means "in Turkey"*
	[5]	Die Gesamtzahl der injizierten Impfstoffe erreichte 109.990.742 Dosen. *Missing the information about the country.*
	[9]	Die Gesamtzahl der injizierten Impfstoffe im Land erreichte 109.990.742 Dosen. *Perfect translation.*
Reference		Die Gesamtzahl der im Land verabreichten Impfungen beläuft sich auf 109.990.742 Dosen.

Source		Long story short, she comes back to the checkout, but goes to another line.
QA Label	[0]	Kurz gesagt, sie kommt zurück zur Kasse, aber geht zu einer anderen Zeile. *"Zeile" is an incorrect word in this context.*
	[5]	Kurz gesagt, sie kommt zurück zur Kasse, geht aber zu einer anderen Linie. *"Linie" is better, but still not natural.*
	[9]	Lange Rede kurzer Sinn, sie kommt zurück zur Kasse, geht aber zu einer anderen Schlange. *"Schlange" is the correct word. Also nice use of the idiom "Lange Rede kurzer Sinn".*
Reference		Um es kurz zu machen, sie kam zurück zur Kasse, stellte sich aber bei einer anderen Schlange an.

Table 9: Annotated example translations for different quality labels used for prompting.

Table 10: Performance of combining Quality-Aware Prompting and Quality Prediciton approaches on both full and filtered datasets. Two combined models are trained for each dataset: one with 10 and the other with 20 prediction quality score bins, while using 5 quality score bins for prompting. Combining both approaches yields no improvements across metrics.

Method	Data	MetricX	COMET
Quality-Aware Prompting	Full	82.3	87.1
Quality-Aware Prediction	Full	82.0	86.6
Combo: ProBins=5 and PreBins=10	Full	82.1	86.7
Combo: ProBins=5 and PreBins=20	Full	82.1	86.7
Quality-Aware Prompting	Filter	82.6	87.3
Quality-Aware Prediction	Filter	82.5	86.9
Combo: ProBins=5 and PreBins=10	Filter	81.9	86.6
Combo: ProBins=5 and PreBins=20	Filter	82.1	86.8