Disentangling Uncertainty in Machine Translation Evaluation

Chrysoula Zerva[1,4] **Taisiya Glushkova**[1,4] **Ricardo Rei**[2,3,4] **André F. T. Martins**[1,2,4]
[1]Instituto de Telecomunicações [2]Unbabel [3]INESC-ID
[4]Instituto Superior Técnico & LUMLIS (Lisbon ELLIS Unit)
{chrysoula.zerva, taisiya.glushkova, ricardo.rei, andre.t.martins}@tecnico.ulisboa.pt

Abstract

Trainable evaluation metrics for machine translation (MT) exhibit strong correlation with human judgements, but they are often hard to interpret and might produce unreliable scores under noisy or out-of-domain data. Recent work has attempted to mitigate this with simple uncertainty quantification techniques (Monte Carlo dropout and deep ensembles), however these techniques (as we show) are limited in several ways – for example, they are unable to distinguish between different kinds of uncertainty, and they are time and memory consuming. In this paper, we propose more powerful and efficient uncertainty predictors for MT evaluation, and we assess their ability to target different sources of aleatoric and epistemic uncertainty. To this end, we develop and compare training objectives for the COMET metric to enhance it with an uncertainty prediction output, including heteroscedastic regression, divergence minimization, and direct uncertainty prediction. Our experiments show improved results on uncertainty prediction for the WMT metrics task datasets, with a substantial reduction in computational costs. Moreover, they demonstrate the ability of these predictors to address specific uncertainty causes in MT evaluation, such as low quality references and out-of-domain data.[1]

1 Introduction

Trainable neural-based metrics, such as COMET or BLEURT (Rei et al., 2020a; Sellam et al., 2020a), hold great promise for MT evaluation (Freitag et al., 2021b). For system comparison, they surpass or complement traditional lexical metrics such as BLEU (Papineni et al., 2002), and at a segment level, they show higher correlations with human judgments, with and without access to references (Kepler et al., 2019; Thompson and Post, 2020; Ranasinghe et al., 2020).

[1]Our code and data is available at: https://github.com/deep-spin/uncertainties_MT_eval

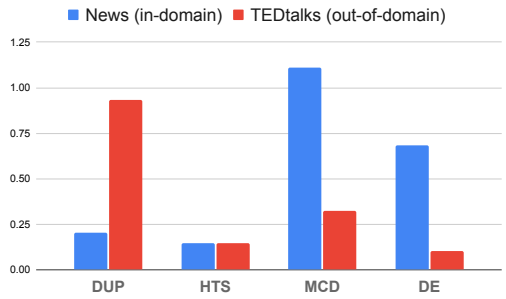


Figure 1: **Epistemic uncertainty caused by out-of-domain data.** We show sharpness (average uncertainty) on two English-Russian test sets from the WMT21 metrics task: an in-domain dataset (News) and an out-of-domain dataset (TED talks). Our proposed method that handles epistemic uncertainty (Direct Uncertainty Predictor – DUP) exhibits higher uncertainty on the out-of-domain dataset, as expected. The heteroscedastic (HTS) predictor, which detects aleatoric, but not epistemic uncertainty, has similar uncertainty in both datasets, and the MC dropout (MCD) and deep ensemble (DE) baselines, surprisingly, has the opposite behavior.

However, trainable MT evaluation metrics are not always trustworthy: For example, they can be unreliable in out-of-domain data and low resource languages, and sometimes they disregard specific error types, attributing high scores to low quality translations (Amrhein and Sennrich, 2022). Hence, we need a measure of **confidence** over their quality predictions for each segment, so that they can be better contextualized and interpreted. Recently, Glushkova et al. (2021) proposed **uncertainty-aware MT evaluation** by combining COMET with two simple uncertainty quantification methods, both based on model variance, namely, Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017). However, these two methods have two important shortcomings:

- They are costly in terms of inference time (MC dropout) and training time (deep ensembles).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8622 - 8641
December 7-11, 2022

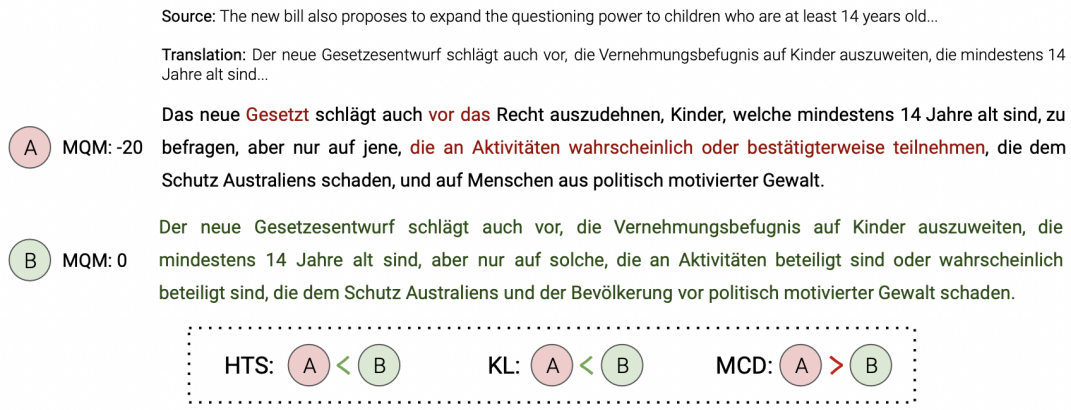


Figure 2: **Aleatoric uncertainty caused by noisy references.** We show a low quality reference (A) and a high quality reference (B) for an English-German translation (the translation in the example is high quality, identical to reference B). Errors in reference A are annotated in dark red; reference B has a perfect MQM score of 0 (no errors). Our two proposed methods that handle aleatoric (data) uncertainty, HTS and KL, are more uncertain when given the low-quality reference, as expected. The previously proposed MCD method (Glushkova et al., 2021) behaves in the opposite way. Full dataset statistics are shown in Figure 3.

- They are not able to detect or distinguish between different sources of uncertainty. For example, it is impossible to infer whether the predicted uncertainty stems from a noisy and ambiguous reference, an out-of-distribution example, or noisy annotations. More fundamentally, they are highly model-dependent and cannot distinguish between aleatoric (data) and epistemic (model) uncertainty, as illustrated in Figures 1–2.

In this paper, we address the limitations above by investigating more powerful and efficient uncertainty quantification methods: **direct uncertainty prediction** (Jain et al., 2021), a two-step approach which uses supervision over the quality prediction errors; **heteroscedastic regression**, which estimates input-dependent aleatoric uncertainty and can be combined with MC dropout (Kendall and Gal, 2017); and **divergence minimization**, which can estimate uncertainty from annotator disagreements, when multiple annotations are available for the same example. We examine the degree to which these predictors can improve segment-level uncertainty-aware MT evaluation and target phenomena related to specific types of uncertainty: (i) aleatoric uncertainty in the case of heteroscedastic regression and divergence minimization, and (ii) epistemic uncertainty in the case of direct uncertainty prediction.

We evaluate our newly proposed uncertainty estimators on 16 language pairs from the WMT20 and WMT21 metrics shared task, using two types of human annotations: direct assessments (DA) and multi-dimensional quality metric scores (MQM). The experiments show that our estimators compare favourably against model variance baselines (MC dropout and deep ensembles), while being considerably faster. We also show that we can address specific issues for MT evaluation, such as detecting potentially incorrect references and out-of-distribution examples in the data, by choosing the most suitable uncertainty predictor among our proposed methods.

2 Related Work

MT evaluation Traditional metrics for MT evaluation are based on lexical overlap, including BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009), and CHRF (Popović, 2015). More recent metrics leverage large pretrained models, either unsupervised, such as BERTSCORE (Zhang et al., 2019), YISI (Lo, 2019) and PRISM (Thompson and Post, 2020), or fine-tuned on human annotations, such as COMET (Rei et al., 2020a) and BLEURT (Sellam et al., 2020b). In recent studies it has become increasingly evident that supervised metrics exhibit higher correlations with human judgements (Mathur et al., 2020; Freitag et al., 2021a) and produce more reliable assessments of MT quality (Kocmi et al., 2021). Nonetheless, all these metrics output a single point estimate, with the exception of UA-COMET (Glushkova et al., 2021), which returns a confidence interval along with a quality estimate. Our work builds upon UA-COMET by proposing

improved uncertainty quantification.

Uncertainty quantification The problem of over-confident incorrect predictions affects neural models across tasks, and thus there are several works applying uncertainty quantification techniques to address this. Model variance methods such as MC Dropout (Gal and Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017) have been applied on a range of tasks to estimate the total uncertainty of a model. However, these methods are computationally costly to train and apply. Malinin et al. (2019) propose to address this shortcoming with ensemble distribution distillation via prior networks (addressing the inference cost of ensembling). They further investigate the adaptation of the aforementioned method to regression problems (Malinin et al., 2020), proposing two methods to estimate Regression Prior Networks (RPN), which however require either access to out-of-distribution data or the distillation of an ensemble of regression models into an RPN.

Recently, Ulmer and Cinà (2021) have shown that variance-based uncertainty estimation methods, which employ ensembling or MC dropout, can be unstable when applied to out-of-distribution data and often fail to provide accurate uncertainty estimates. Raghu et al. (2019), Hu et al. (2021), and Jain et al. (2021) corroborate these findings and propose to train a direct epistemic uncertainty predictor on the errors of the main model as a better method to estimate epistemic uncertainty. To the best of our knowledge, direct uncertainty prediction has not been examined on MT evaluation (or other NLP tasks). Contrary to epistemic uncertainty, aleatoric (data) uncertainty corresponds to the irreducible amount of prediction error(s), which is due to the noise present in the observed data. Kendall and Gal (2017) propose the use of heteroscedastic variance in the loss function. Wang et al. (2019) propose a test-time augmentation-based aleatoric uncertainty. They compare and combine it with epistemic uncertainty, and show that it provides more representative uncertainty estimates than dropout-based ones alone. Our paper takes inspiration from these techniques to estimate aleatoric noise in MT evaluation.

Annotator disagreement Several approaches have been proposed to understand and model annotator bias (Cohn and Specia, 2013; Hovy and Yang, 2021) and to leverage annotator disagreement in NLP applications (Sheng et al., 2008; Plank et al., 2014, 2016; Jamison and Gurevych, 2015; Pavlick and Kwiatkowski, 2019). Recently, soft-label multi-task learning objectives for classification tasks have been proposed by Fornaciari et al. (2021). Our Kullback-Leibler (KL) divergence minimization objective may be regarded as an extension of this approach for regression tasks, replacing (softmax) categoricals by Gaussian distributions.

Uncertainty in NLP There are several works applying uncertainty quantification techniques to NLP, most commonly for (structured) classification tasks. Fomicheva et al. (2020) use MC dropout to model MT confidence, and Malinin and Gales (2020) study structured uncertainty estimation in autoregressive tasks, including MT and speech recognition. Ye et al. (2021) model uncertainty in performance prediction of NLP systems. Mielke et al. (2019) apply heteroscedastic models to assess language difficulty, whereas Friedl et al. (2021) estimate aleatoric uncertainty in scientific peer reviewing. Recently, Wang et al. (2022) focus on calibration of regression models and show that uncertainty can be useful for data augmentation. Our paper also focuses on a regression task although some of our techniques and findings can apply more broadly to these problems.

3 Uncertainty in MT Evaluation

3.1 MT evaluation

Throughout, we denote by s a sentence in a source language, by t its translation into a target language, and by $\mathcal{R}$ a set of reference translations. A segment-level **MT evaluation system** $\mathcal{M}_{\mathrm{Q}}$ (also called a "translation quality metric") is a system that takes as input a triple $\langle s, t, \mathcal{R} \rangle$ and outputs a quality score $\hat{q} \in \mathbb{R}$, reflecting how accurate t is as a translation of s. [2]

Current state-of-the-art evaluation metrics, such as COMET (Rei et al., 2020a) or BLEURT (Sellam et al., 2020a), are trained with supervision on corpora annotated with human judgments $q^* \in \mathbb{R}$, such as direct assessments (DA; Graham et al. 2013) or scores from multi-dimensional quality metric annotations (MQM; Lommel et al. 2014). This supervision encourages their predicted quality scores $\hat{q}$ to approximate the human perceived quality q^*, in a way that generalizes to unseen data.

[2] We focus on reference-based MT evaluation.

3.2 Sources of uncertainty

While neural-based MT evaluation systems are more accurate than traditional lexical-based metrics such as BLEU, they are less transparent and may produce unreliable scores for out-of-domain inputs or when references are noisy (Rei et al., 2020b; Freitag et al., 2021b). Our goal is to mitigate this problem by quantifying the **uncertainty** associated with their predicted scores. This uncertainty can come from several sources:

- **Aleatoric (data) uncertainty** is primarily caused by noise in the data. Frequent sources of noise include inaccurate or inconsistent ground truth quality scores q^* (usually noticeable from low inter-annotator agreement scores) and noisy reference translations $\mathcal{R}$, which can mislead the MT evaluation system (Freitag et al., 2020).

- **Epistemic (model) uncertainty** reflects lack of knowledge from the model itself. This may be caused by limited training data, out-of-distribution examples (e.g., new languages, new domains, or diverse scoring schemes), or by complex, highly non-literal, translations which may trigger weak spots in the MT evaluation model.

Recently, Glushkova et al. (2021) proposed an **uncertainty-aware** evaluation metric (UA-COMET) by experimenting with two simple uncertainty quantification techniques, MC dropout (Gal and Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017). Both techniques compute estimates based on **model variance** – they estimate uncertainty by running multiple versions of the system (either produced on-the-fly with stochastic dropout noise or by using separate models trained with different seeds), and then computing the mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of the predicted scores. When given a triple $\langle s, t, \mathcal{R} \rangle$ as input, instead of returning a point estimate $\hat{q}$, UA-COMET treats the quality score as a random variable Q, modeled as a Gaussian distribution $p_Q(q) = \mathcal{N}(q; \hat{\mu}, \hat{\sigma}^2)$. After a calibration step, the variance parameter of the Gaussian $\hat{\sigma}^2$ is used as the uncertainty estimate.

4 Improving Uncertainty-Aware MT Evaluation

A limitation of UA-COMET is its reliance on model variance techniques that often produce poor estimates of uncertainty and conflate aleatoric and epistemic uncertainty, making it hard to accurately represent uncertainty related to out-of-distribution samples (Jain et al., 2021; Zhang et al., 2021). We therefore examine alternate methods to learn aleatoric and epistemic uncertainty directly from the available data. We assume that for each of the training scenarios and learning objectives described in the following sections, we can learn to predict the uncertainty of quality estimates $\hat{q}$ either as the noise variance σ in the case of aleatoric uncertainty, or as the generalization error ϵ in the case of epistemic (and total) uncertainty.

4.1 Predicting aleatoric uncertainty

Rather than a property of the model, aleatoric uncertainty is a property of the data distribution and thus it can be learned as a function of the data (Kendall and Gal, 2017). It corresponds to uncertainty induced due to noise and inconsistencies. In the case of MT evaluation, we identify low quality references and inconsistent human annotations as the main sources of aleatoric uncertainty. The uncertainty associated with each data instance can vary: references have shown to be of different quality levels (Freitag et al., 2020), while the quality scores depend largely on the annotators who sometimes have high disagreement (Toral, 2020).

Heteroscedasticity A common assumption in regression problems (of which MT evaluation is an example) is that the noise in the data has constant variance throughout the dataset – i.e., that the data is *homoscedastic*. The mean squared error loss, for example, corresponds to the maximum likelihood criterion under Gaussian noise with fixed variance. However, this is not a suitable assumption in several problems, including MT evaluation, where real data is often **heteroscedastic** – for example, complex sentences requiring specific background knowledge may be subject to larger annotation errors (higher disagreement among annotators) and higher chance for noisy references than simpler sentences. Therefore, the aleatoric uncertainty should be larger in those cases.

Heteroscedastic regression We model aleatoric uncertainty as observation noise by training a model to predict not only a quality score for each triple, but also a variance estimate $\hat{\sigma}^2$ for this score. Under our heteroscedastic assumption, we assume that the variance is specific to each data sample and can be learned as a function of the data. We follow Le et al. (2005) and Kendall and Gal (2017) and in-

corporate $\hat{\sigma}^2$ as part of the training objective, while learning the MT evaluation model parameters.

Formally, let $x := \langle s, t, \mathcal{R} \rangle$ denote an input triple, as described in §3. Our heteroscedastic uncertainty-aware MT evaluation system $\mathcal{M}_{\text{Q}}^{\text{HTS}}$ is a neural network that takes x as input and outputs a mean score $\hat{\mu}(x)$ and a variance score $\hat{\sigma}^2(x)$ – in practice, this is done by taking a COMET model and changing the output layer to output two scores ($\hat{\mu}(x)$ and $\log \hat{\sigma}^2(x)$) instead of one ($\hat{q}(x)$). This predicted mean and variance parametrize a Gaussian distribution $\hat{p}_Q(q|x;\theta) = \mathcal{N}(q; \hat{\mu}(x;\theta), \hat{\sigma}^2(x;\theta))$, where θ are the model parameters. Given a training set $\mathcal{D} = \{(x_1, q_1^*), \ldots, (x_N, q_N^*)\}$, the maximum likelihood training criterion amounts to maximize

$$\frac{1}{N}\sum_{i=1}^{N} \log \underbrace{\mathcal{N}(q_i^*; \hat{\mu}(x_i,\theta), \hat{\sigma}^2(x_i,\theta))}_{p_Q(q_i^*|x_i;\theta)} = \quad (1)$$

$$= -\frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\text{HTS}}(\hat{\mu}(x_i,\theta), \hat{\sigma}^2(x_i,\theta); q_i^*) + \text{const.},$$

where $\mathcal{L}_{\text{HTS}}$ denotes the **heteroscedastic loss**:

$$\mathcal{L}_{\text{HTS}}(\hat{\mu}, \hat{\sigma}^2; q^*) = \frac{(q^* - \hat{\mu})^2}{2\hat{\sigma}^2} + \frac{1}{2}\log \hat{\sigma}^2. \quad (2)$$

We can see that, if $\hat{\sigma}^2$ was constant and not estimated, the heteroscedastic loss $\mathcal{L}_{\text{HTS}}$ would revert to a standard squared loss; however, since this variance is predicted by the model and changes with the input, the model is trained to make a trade-off: the $\hat{\sigma}^2$ term in the denominator down-weights examples where the target q^* is assumed unreliable, decreasing the impact of highly noisy instances (a form of weighted least squares), while the $\log \hat{\sigma}^2$ term penalizes the model if it overestimates the variance. We show in §5.3 how this variance can be used to detect possibly noisy references.

KL divergence minimization While heteroscedastic uncertainty allows to estimate the observation noise, when we have multiple annotations for the same example we may have additional information on data uncertainty reflected in **annotator disagreement**. We assume that annotator disagreement in this case can be used as a proxy to data uncertainty.

Similarly to the estimation of heteroscedastic variance with the $\mathcal{L}_{\text{HTS}}$ objective, we assume that we can learn the variance $\hat{\sigma}(x;\theta)$ as an estimator of aleatoric uncertainty alongside the rest of the model, but now leveraging the supervision coming from the annotator disagreement – we denote this system by $\mathcal{M}_{\text{Q}}^{\text{KL}}$. We model the annotator scores as another Gaussian distribution $p_Q^*(q \mid x) = \mathcal{N}(q; \mu^*(x), \sigma^*(x))$, where $\mu^*(x)$ is the sample mean and $\sigma^*(x)$ the sample variance of the annotator scores for the example x, used as targets for our model predictions. We formalize this as a KL divergence objective between the target distribution p_Q^* and the predicted distribution $\hat{p}_Q$, which has the following closed form for Gaussian distributions:

$$\mathcal{L}_{\text{KL}}(\hat{\mu}, \hat{\sigma}^2; \mu^*, \sigma^{*2}) = \text{KL}(p_Q^* \| \hat{p}_Q)$$
$$= \frac{(\mu^* - \hat{\mu})^2 + \sigma^{*2}}{2\hat{\sigma}^2} + \frac{1}{2}\log\frac{\hat{\sigma}^2}{\sigma^{*2}} - \frac{1}{2}. \quad (3)$$

Note that Eq. 3 is a generalization of Eq. 2: if we assume a fixed zero-limit variance $\sigma^{*2} \to 0$, we recover Eq. 2 up to a constant.

4.2 Predicting epistemic uncertainty

Epistemic (model) uncertainty can be observed mainly on out-of-sample and out-of-distribution instances, and manifests as the *reducible* generalization error of the model – in the presence of infinite training data and suitable model and learning algorithm, epistemic uncertainty could be reduced to zero (Postels et al., 2021; Jain et al., 2021). We outline two procedures to estimate epistemic and total uncertainty, one combining MC dropout with the heteroscedastic loss (Kendall and Gal, 2017), and another which estimates uncertainty directly as the generalization error (Jain et al., 2021).

Heteroscedastic MC dropout Given a way to estimate aleatoric uncertainty $\hat{\sigma}$, e.g., using Eqs. 2 or 3, we can combine it with an estimator of epistemic uncertainty to obtain the total uncertainty over a sample. Assuming we have access to an MT evaluation model that is able to predict both a quality score $\hat{q}$ and an aleatoric uncertainty estimate $\hat{\sigma}$ – such as the system $\mathcal{M}_{\text{Q}}^{\text{HTS}}$ described in §4.1 – we can use a stochastic strategy such as MC dropout or deep ensembles to obtain a set $\mathcal{Q} = \{\hat{q}_1, \ldots, \hat{q}_M\}$ of quality estimates and $\Sigma = \{\hat{\sigma}_1^2, \ldots, \hat{\sigma}_M^2\}$ of variance estimates. Assuming $\mathcal{Q}$ is a sample drawn from a Gaussian distribution, the sample variance can be used as an estimator of epistemic uncertainty, and the sample mean of Σ can be used as an estimator of aleatoric uncertainty (Kendall and Gal, 2017). We can then estimate the total uncertainty

over the M samples as the sum of epistemic and aleatoric uncertainties:

$$\hat{\mathcal{U}}_{\text{total}} = \text{Var}[\mathcal{Q}] + \mathbb{E}[\Sigma] \quad (4)$$
$$= \underbrace{\frac{1}{M}\sum_{j=1}^{M}\hat{q}_j^2 - \left(\frac{1}{M}\sum_{j=1}^{M}\hat{q}_j\right)^2}_{\text{epistemic}} + \underbrace{\frac{1}{M}\sum_{j=1}^{M}\hat{\sigma}_j^2}_{\text{aleatoric}}.$$

For the experiments presented in §5 we use this strategy with MC dropout applied to a model trained with heteroscedastic regression.

Direct uncertainty prediction An alternative is to consider the total uncertainty $\hat{\mathcal{U}}_{\text{total}}$ as an approximation of the **generalization error** of the MT evaluation model $\mathcal{M}_{\text{Q}}$. In this case, assuming access to $\mathcal{M}_{\text{Q}}$'s predictions $\hat{q}$ and the ground truth quality scores q^* on a new (unseen) set of samples, we could learn to predict the total uncertainty **directly** as the error ϵ between the model predictions $\hat{q}$ and the true scores q^*, using the strategy recently proposed by Jain et al. (2021).

As opposed to the previously described uncertainty estimation approaches, direct uncertainty prediction (DUP) is a **two-step process**, as we need to first obtain the model $\mathcal{M}_{\text{Q}}$ that generates the predictions $\hat{q}$ that will allow us to estimate the target errors in a second stage. Hence, we need access to two distinct datasets on which two separate models have to be trained. We assume a dataset $\mathcal{D}_{\text{Q}}$ where $\mathcal{M}_{\text{Q}}$ is trained (we use the vanilla COMET system), and another, disjoint dataset $\mathcal{D}_{\text{E}}$ where we train a second system $\mathcal{M}_{\text{E}}$ to predict the uncertainty/error of $\mathcal{M}_{\text{Q}}$'s predictions. For this purpose, we use $\mathcal{M}_{\text{Q}}$ to annotate $\mathcal{D}_{\text{E}}$ with quality estimates $\hat{q}$, and then we calculate the ground truth error ϵ^* as the distance to the human quality scores q^* for each segment in $\mathcal{D}_{\text{E}}$, $\epsilon^* = |\hat{q} - q^*|$. We use ϵ^* as the target to train $\mathcal{M}_{\text{E}}$, given inputs $\langle s, t, \mathcal{R}, \hat{q}\rangle$. Letting $\hat{\epsilon}$ correspond to the uncertainty predicted by $\mathcal{M}_{\text{E}}$ on a given input, we define $\mathcal{L}^{\text{E}}_{\text{HTS}}$ function for $\mathcal{M}_{\text{E}}$:

$$\mathcal{L}^{\text{E}}_{\text{HTS}}(\hat{\epsilon}; \epsilon^*) = \frac{(\epsilon^*)^2}{2\hat{\epsilon}^2} + \frac{1}{2}\log(\hat{\epsilon})^2. \quad (5)$$

$\mathcal{L}^{\text{E}}_{\text{HTS}}$ is inspired by the heteroscedastic loss of Eq. 2, where the model is discouraged from predicting too high uncertainty values because of the term $\log(\hat{\epsilon})^2$, while it will still try to predict high $\hat{\epsilon}$ values for the samples where the MT quality score is not close to the human evaluation. Therefore, this choice is akin to a two-step approach to heteroscedastic regression: one step to train the "mean" predictor and another step for training the variance predictor given the mean predictions, where the two steps are performed on different partitions of the dataset, $\mathcal{D}_{\text{Q}}$ and $\mathcal{D}_{\text{E}}$. We show in Appendix F that $\mathcal{L}^{\text{E}}_{\text{HTS}}$ outperforms other loss functions.

5 Experiments

The main focus of our experiments is to investigate how the uncertainty estimators we explore in this paper compare to each other and against proposed variance-based methods. Our comparisons address the accuracy of uncertainty predictions on MT evaluation datasets (§5.2) as well as more specific concerns such as the performance on out-of-domain data (§5.2), the ability to detect low quality references (§5.3), and the computational costs (§5.4).

5.1 Experimental Setup

We follow Glushkova et al. (2021) and use COMET (v1.0) as the underlying architecture for our MT evaluation models, trained on the data from the WMT17-WMT19 metrics shared task (Freitag et al., 2021b). We consider two types of human judgments: direct assessments (DA) and multi-dimensional quality metric scores (MQM).

Experiments on DA scores We evaluate our models using 5-fold cross validation on the WMT20 dataset.[3] All single-step models are trained on the data from the WMT17-19 metrics shared task using the development folds (80%) for calibration. For DUP models, WMT17-19 is used to train the first step model $\mathcal{M}_{\text{Q}}$ and the development folds of WMT20 are used both for training the second step of the model $\mathcal{M}_{\text{E}}$ and for calibration. The data encompasses 16 language pairs (per-language results listed in Tables 2–3 in Appendix A), which we aggregate into two groups, EN-XX (out-of-English) and XX-EN (into-English). We report the balanced average across all language pairs (AVG).

Experiments on MQM scores We fine-tune all models on the entire WMT20 MQM dataset, which consists of MQM annotations for English-German (EN-DE) and Chinese-English (ZH-EN). For DUP, we finetune the $\mathcal{M}_{\text{E}}$ model on WMT20. For testing

[3]We ensure that triplets from the same document appear all in a single fold so that all folds are disjoint. All folds are balanced with respect to the percentage of documents/source segments available for each language pair.

and calibration we use WMT21 metrics shared task dataset, which contains MQM annotations for the same language pairs, but also with an addition of English-Russian (EN-RU). We evaluate using 5-fold cross validation on the WMT21 MQM data as well, where the development folds are used for calibrating models for each language pair. We also provide the performance on WMT21 without any finetuning on MQM scores in Appendix B.

Models In the experiments that follow we use as baselines the two variance-based methods proposed by Glushkova et al. (2021): a MC dropout model with 100 dropout runs (**MCD**) and a deep ensemble of 5 independent models (**DE**), as well as the fixed-variance simple baseline they proposed: $\sigma^2_{\text{fixed}} = \frac{1}{|\mathcal{D}|}\sum_{\langle s,t,\mathcal{R},q^*\rangle\in\mathcal{D}}(q^* - \hat{\mu})^2$.
We compare these baselines against our models:

- **HTS**: The heteroscedastic model $\mathcal{M}^{\text{HTS}}_{\text{Q}}$ trained with the loss in Eq. 2.
- **HTS+MCD**: The combination of **HTS** with MC dropout as described in Eq. 4.
- **DUP**: The direct uncertainty prediction model described in §4.2 using the $\mathcal{L}^{\text{E}}_{\text{HTS}}(\hat{\epsilon};\epsilon^*)$ loss described in Eq. 5.We use a vanilla COMET model as $\mathcal{M}_{\text{Q}}$ and a system with the same architecture for $\mathcal{M}_{\text{E}}$ which receives as an additional feature the predicted quality score $\hat{q}$ from $\mathcal{M}_{\text{Q}}$. This extra feature is added by inserting a bottleneck layer between two feed-forward layers in the original COMET architecture (see App. C).
- **KL**: The divergence minimization model $\mathcal{M}^{\text{KL}}_{\text{Q}}$ using the objective in Eq. 3. This model is used only for the experiments with MQM scores (Table 1), where multiple annotators for the same examples are available during training.[4]

Evaluation To compare the performance of uncertainty predictors, we report the same performance indicators as Glushkova et al. (2021): the predictive Pearson score $r(\hat{\mu}, q^*)$ (PPS), the uncertainty Pearson score $r(|q^* - \hat{\mu}|, \hat{\sigma})$ (UPS), the negative log-likelihood $-\log\mathcal{N}(q^*;\hat{\mu},\hat{\sigma}^2)$ (NLL), the expected calibration error (ECE; Naeini et al. (2015)), and the sharpness (Sha.), i.e., the average predicted variance in the test set (see Appendix D for details about these metrics). Note that we follow (Glushkova et al., 2021) in considering sharp confidence intervals desirable for all our in-domain experiments, however, for out-of-domain instances, the desired behaviour differs: we expect that the average predicted uncertainty on out-of-domain data would be higher compared to the average uncertainty observed on in-domain data. Hence higher sharpness values would be desirable in such cases (see also Figure 1 and Appendix E).

These indicators assess both quality prediction accuracy (PPS), uncertainty-related accuracy (UPS) and calibration (ECE, Sha.), and the prediction and uncertainty accuracy combined in a single score (NLL). We consider UPS as our main indicator of performance, but report the other uncertainty indicators for completeness. PPS is reported as well, to assert that the performance of the quality predictions $\hat{q}$ of the MT evaluation model is not compromised. Additionally, we consider changes in average sharpness to be more indicative of the interpretability of the uncertainty predictions and the sensitivity of the model to domain and distribution shifts. We illustrate this in Figure 1.

5.2 Comparison of uncertainty methods

The results of the DA and MQM experiments are shown in Table 1. As expected, the PPS values (which do not measure uncertainty, but accuracy of the quality predictions) are similar for all methods, since they are based either on a vanilla COMET model, or on an ensemble of COMET models, with an advantage for the **DE** method which benefits from the ensemble effect. **HTS** and **KL**, which have modified objectives that learn the mean and the variance simultaneously, also boost PPS, but not as much as **DE**. We focus our analysis on the uncertainty prediction accuracy, assessed primarily by UPS and also ECE, and Sharpness indicators.

For the DA experiments, we observe that our proposed methods, **HTS**, **DUP** and **KL**, show significantly [5] stronger uncertainty correlation (UPS) than the baseline estimates (**MCD** and **DE**), and obtain competitive scores for ECE, Sha. and NLL without significantly compromising PPS.

Enhancing $\mathcal{M}^{\text{HTS}}_{\text{Q}}$ with MC dropout (**HTS+MCD**) seems to further improve UPS and ECE, but produces less sharp uncertainty estimates and it negatively impacts the predictive accuracy. **DUP**'s main strength relates to provision

[4] Unlike the other models, the KL model is trained directly on the WMT20 MQM dataset (instead of being just fine-tuned there), since the WMT data with direct assessments does not include information on annotator disagreement that is used as target for the KL model training.

[5] $p<0.05$ using William's test.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
WMT20 DA	σ^2-fixed	–	0.019	0.415	1.236	0.444
	MCD	0.106	0.016	<u>0.377</u>	1.199	0.443
	DE	0.134	0.019	0.366	<u>1.156</u>	<u>0.460</u>
	HTS	0.177	0.015	0.450	1.201	0.444
	HTS+MCD	<u>0.254</u>	<u>0.013</u>	0.528	1.167	0.429
	DUP	0.182	0.014	0.437	1.190	0.444
WMT21 MQM	σ^2-fixed	–	0.055	0.371	2.090	0.377
	MCD	0.179	<u>0.024</u>	0.334	1.686	0.460
	DE	0.128	0.051	<u>0.236</u>	2.631	<u>0.479</u>
	HTS	0.307	0.041	0.284	2.264	0.445
	HTS+MCD	<u>0.311</u>	0.037	0.388	<u>1.614</u>	0.445
	KL	0.296	0.046	0.273	2.595	0.443
	DUP	0.285	0.039	0.634	1.778	0.377

Table 1: Results for segment-level DA and MQM predictions, averaged over all language pairs. <u>Underlined</u> numbers indicate the best result for each evaluation metric in each language pair.

of informative uncertainty intervals (changes in sharpness), while maintaining good performance for the other uncertainty metrics. As we can see in Figure 1, the sharpness of uncertainty predictions increases for out-of-domain data in the case of **DUP** and nicely captures the increased epistemic uncertainty in such cases. In contrast, we can see that variance-based epistemic uncertainty predictors are weaker in representing this domain shift and actually show the opposite behavior to the desired one, while aleatoric uncertainty (**HTS**) remains the same. We provide a more extended analysis of this aspect in Appendix E. Additionally, we provide the performance of the uncertainty predictors when applied to two other metrics, BLEURT (Sellam et al., 2020b) and UniTE (Wan et al., 2022) in Appendix G. We observe similar patterns, but notice that **HTS** approaches require access to source segments to provide meaningful uncertainty predictions.

The findings on DA data are further supported by the MQM results especially for UPS, and we can see that the models achieve good performance for EN-RU, which is not available in the WMT20 MQM data used for fine-tuning (see Appendix B). We also see that the **KL** model, despite using significantly less training data (see §5.1), achieves competitive results and even outperforms other metrics for EN-DE.

5.3 Identification of noisy references

As mentioned in §3.2, low quality references are a primary source of aleatoric uncertainty. Thus, we expect the uncertainty predictors that model aleatoric uncertainty (**HTS** and **KL**) to be more

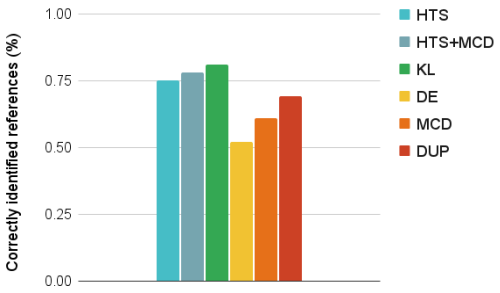


Figure 3: Percentage of correctly recognized references with higher quality (r_{good} versus r_{bad}) by different uncertainty predictors on the EN-DE dataset.

indicative of erroneous references compared to the other uncertainty predictors. To verify this hypothesis, we conduct an experiment on the WMT21 MQM EN-DE dataset, which includes 4 references, each annotated with MQM scores by a human annotator (Freitag et al., 2021b). For each $\langle s, t \rangle$ pair in the test split, we select the best reference r_{good} and the worst reference r_{bad} based on the respective MQM scores. We retain only the $\langle s, t, \{r_{\text{good}}, r_{\text{bad}}\} \rangle$ for which $|\text{MQM}(r_{\text{good}}) - \text{MQM}(r_{\text{bad}})| >= 10$, so that there is a considerable quality difference between the references.[6] We then apply the uncertainty predictors on the selected triples $\langle s, t, r_{\text{good}} \rangle$ and $\langle s, t, r_{\text{bad}} \rangle$ and obtain the predicted uncertainties, as shown in Figure 2. For each $\langle s, t \rangle$ pair, we check which reference leads to the lowest predicted uncertainty and compute how often that reference coincides with r_{good}. In Figure 3, we can see that all the **HTS**, **HTS+MCD** and the **KL** predictors are much more successful in choosing the correct reference compared to **MCD**, **DE** and **DUP**. This confirms the hypothesis that **HTS** and **KL** are more effective at capturing aleatoric uncertainty. Additionally, it is interesting to note that the combination of MC dropout with heteroscedastic loss provides a small boost to the accuracy of distinguishing the noisy reference.

5.4 Computational cost

We now turn to the computational cost associated with the different uncertainty quantification methods, both in terms of training and inference runtime. In Figure 4, we present the inference and training times for each of the models (we used the same maximum number of epochs for each model). The

[6] An MQM penalty of 10 points corresponds to at least 2 major errors (Freitag et al., 2021a).

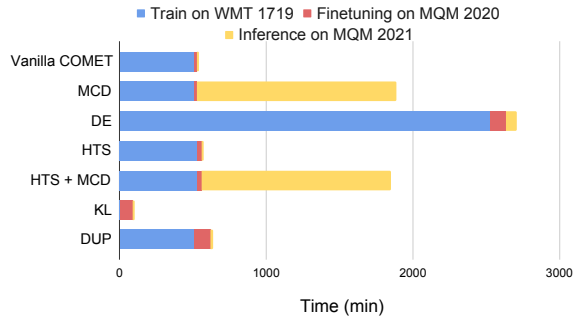


Figure 4: Combined training, fine-tuning and inference times for the experiments reported in Table 1. All experiments were performed on a server with 4 Quadro RTX 6000 (24GB), 12 Intel Xeon Silver 4214@2.20GHz CPUs, and 256 Gb of RAM; time calculated for training/inference on a single GPU.

large inference times for **MCD** and **HTS+MCD** stem from the need to perform 100 runs (the optimal number according to Glushkova et al. (2021)); for DE, 5 models are ensembled, increasing training and inference costs 5-fold (for training details see Table 6 in Appendix C). In contrast, **HTS**, **KL**, and **DUP** have much lower costs (with slightly higher costs for **DUP** due to the need to train/run a second system) without performance compromises.

6 Conclusions

We assessed the potential of different uncertainty predictors to capture different sources of uncertainty in MT evaluation. We demonstrated that methods modeling heteroscedasticity can detect noisy references as a source of aleatoric uncertainty, and that the direct epistemic prediction method reflects well the increased epistemic uncertainty under a domain shift. Besides providing more informative uncertainty estimates than MC dropout and deep ensemble methods, our proposed predictors are also computationally cheaper.

Overall, our work provides insight about which uncertainty predictors to choose for MT evaluation depending on the uncertainty source(s) to be addressed. The proposed uncertainty predictors that are able to target specific types of uncertainty are the first step towards mitigating the sources of such uncertainty, i.e. removing noisy instances from training to reduce aleatoric uncertainty, or identifying informative instances that would allow adapting to a new domain to reduce epistemic uncertainty. In future work, we are planning to further explore their properties and potential in improving MT and MT evaluation performance.

Acknowledgements

This work was supported by the European Research Council (ERC StG DeepSPIN 758969), by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by P2020 project MAIA (LISBOA-01-0247- FEDER045909), and Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

Limitations

Our work addresses an important limitation of existing MT evaluation metrics – the absence of reliable uncertainty estimates for their predictions and their inability to distinguish sources of uncertainty. However, our proposed approach has its own limitations as well. First, the scope of our work is limited by the availability of resources with human quality annotations covering multiple languages. Specifically, we are limited to the domains and language pairs addressed in the WMT metrics tasks (2017–2021) whose human assessments consist of DAs and MQM annotations. While these datasets include both high and low resource languages, most WMT datasets cover language pairs from or to English. While certainly experimenting with more language pairs and domains in future work might provide additional insights, the WMT datasets used in our paper encompass 16 language pairs for testing and 24 for training, which still provides valuable information of variability across languages. Second, the amount of sentences scored by more than one human annotator is scarce, and for this reason the experiments with the KL objective are limited to a relatively small scale, which prevents a thorough comparison with the other uncertainty quantification methods. Third, while the uncertainty-related training objectives we propose are fully general and can be applied to any supervised neural metric, we only experimented with COMET in this paper, due to limited computational resources. Experimenting with other base metrics to see if they exhibit the same patterns is an interesting topic for future research. Finally, our choice of uncertainty quantification techniques was guided by the desire to prioritize scalable and efficient methods that are applicable to different metrics and fit the MT evaluation task. Overall, we picked 6 different techniques (MCD, DE, HTS, HTS+MCD, KL, DUP) and left out other uncertainty quantifi-

cation methods with less favorable efficiency or scalability properties.

References

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*, Online. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, He-Yan Huang, and Ming Zhou. 2021. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32–42.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

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

Korbinian Friedl, Georgios Rizos, Lukas Stappen, Madina Hasan, Lucia Specia, Thomas Hain, and Björn Schuller. 2021. Uncertainty aware review hallucination for science article classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5004–5009, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602.

Shi Hu, Nicola Pezzotti, and Max Welling. 2021. Learning to predict error for mri reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 604–613. Springer.

Moksh Jain, Salem Lahlou, Hadi Nekoei, Victor Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. 2021. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*.

Emily Jamison and Iryna Gurevych. 2015. Noise or additional information? leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 291–297.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*.

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.

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

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. 2018. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2796–2804. PMLR.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alon Lavie and Michael Denkowski. 2009. The Meteor metric for automatic evaluation of Machine Translation. *Machine Translation*, 23:105–115.

Quoc V. Le, Alex J. Smola, and Stéphane Canu. 2005. Heteroscedastic gaussian process regression. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 489–496, New York, NY, USA. Association for Computing Machinery.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463.

Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales. 2020. Regression prior networks. *arXiv preprint arXiv:2006.11590*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. 2019. Ensemble distribution distillation. In *International Conference on Learning Representations*.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2901–2907. AAAI Press.

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.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 742–751.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. 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.

Janis Postels, Mattia Segu, Tao Sun, Luc Van Gool, Fisher Yu, and Federico Tombari. 2021. On the practicality of deterministic epistemic uncertainty. *arXiv preprint arXiv:2107.00649*.

Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. 2019. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*, pages 5281–5290. PMLR.

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. 2020a. 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.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020b. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Antonio Toral. 2020. Reassessing claims of human parity and super-human performance in machine translation at WMT 2019. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 185–194, Lisboa, Portugal. European Association for Machine Translation.

Dennis Ulmer and Giovanni Cinà. 2021. Know your limits: Uncertainty estimation with relu classifiers fails at reliable ood detection. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1766–1776. PMLR.

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.

Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. 2019. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Jing Zhang, Yuchao Dai, Mochu Xiang, Deng-Ping Fan, Peyman Moghadam, Mingyi He, Christian Walder, Kaihao Zhang, Mehrtash Harandi, and Nick Barnes. 2021. Dense uncertainty estimation. *arXiv preprint arXiv:2110.06427*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A DA experiments

Results per language pair are presented in Tables 2 and 3.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-CS	σ^2-fixed	–	0.005	0.408	1.019	0.699
	MCD	0.149	0.012	0.258	0.847	0.692
	DE	0.125	0.010	0.255	0.825	0.734
	HTS	0.105	0.003	0.456	1.025	0.699
	HTS+MCD	0.335	0.008	0.388	0.880	0.676
	DUP	0.146	0.003	0.419	1.010	0.699
EN-DE	σ^2-fixed	–	0.032	0.172	1.106	0.586
	MCD	0.284	0.013	0.260	0.916	0.582
	DE	0.209	0.025	0.198	0.949	0.616
	HTS	0.231	0.022	0.243	1.016	0.586
	HTS+MCD	0.386	0.004	0.369	0.924	0.612
	DUP	0.232	0.022	0.190	1.028	0.586
EN-JA	σ^2-fixed	–	0.011	0.186	0.714	0.636
	MCD	0.236	0.005	0.225	0.697	0.638
	DE	0.140	0.010	0.179	0.657	0.661
	HTS	0.184	0.007	0.218	0.691	0.636
	HTS+MCD	0.249	0.003	0.520	1.048	0.669
	DUP	0.220	0.005	0.201	0.679	0.636
EN-PL	σ^2-fixed	–	0.010	0.410	1.090	0.609
	MCD	0.167	0.010	0.326	0.947	0.610
	DE	0.129	0.011	0.315	0.960	0.645
	HTS	0.103	0.008	0.436	1.095	0.609
	HTS+MCD	0.189	0.004	0.397	0.969	0.631
	DUP	0.112	0.004	0.431	1.094	0.609
EN-RU	σ^2-fixed	–	0.018	0.287	0.989	0.534
	MCD	0.200	0.012	0.307	0.943	0.538
	DE	0.146	0.016	0.263	0.898	0.570
	HTS	0.156	0.012	0.321	1.006	0.534
	HTS+MCD	0.370	0.017	0.637	1.050	0.497
	DUP	0.154	0.009	0.305	0.991	0.534
EN-TA	σ^2-fixed	–	0.017	0.356	1.077	0.661
	MCD	0.077	0.004	0.406	1.035	0.658
	DE	0.111	0.010	0.361	1.030	0.676
	HTS	0.250	0.009	0.435	1.050	0.661
	HTS+MCD	0.231	0.038	0.921	1.223	0.617
	DUP	0.252	0.009	0.385	1.032	0.661
EN-ZH	σ^2-fixed	–	0.017	0.173	1.016	0.325
	MCD	0.083	0.018	0.152	1.367	0.330
	DE	0.560	0.021	0.152	0.813	0.327
	HTS	0.728	0.003	0.273	0.657	0.325
	HTS+MCD	0.504	0.020	0.380	0.865	0.562
	DUP	0.722	0.003	0.268	0.650	0.325
EN-XX	σ^2-fixed	–	0.015	0.288	1.000	0.566
	MCD	0.163	0.011	0.265	0.984	0.566
	DE	0.223	0.015	0.240	0.864	0.591
	HTS	0.272	0.008	0.344	0.919	0.566
	HTS+MCD	0.323	0.017	0.516	0.994	0.609
	DUP	0.285	0.007	0.320	0.910	0.566

Table 2: Results for segment-level DA prediction for En-Xx LPs. Underlined numbers indicate the best result for each evaluation metric in each language pair.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
CS-EN	σ^2-fixed	–	0.026	0.509	1.422	0.216
	MCD	0.099	0.012	0.462	1.319	0.215
	DE	0.134	0.019	0.366	1.156	0.460
	HTS	0.077	0.024	0.518	1.432	0.216
	HTS+MCD	0.229	0.006	0.502	1.276	0.195
	DUP	0.082	0.024	0.516	1.418	0.216
DE-EN	σ^2-fixed	–	0.025	0.403	1.398	0.573
	MCD	0.044	0.030	0.312	1.343	0.568
	DE	0.056	0.030	0.321	1.374	0.574
	HTS	0.099	0.024	0.425	1.389	0.573
	HTS+MCD	0.148	0.014	0.463	1.107	0.563
	DUP	0.100	0.023	0.432	1.382	0.573
JA-EN	σ^2-fixed	–	0.020	0.494	1.344	0.348
	MCD	0.064	0.008	0.532	1.280	0.349
	DE	0.079	0.012	0.502	1.305	0.360
	HTS	0.145	0.015	0.534	1.351	0.348
	HTS+MCD	0.215	0.007	0.611	1.322	0.339
	DUP	0.129	0.016	0.513	1.333	0.348
KM-EN	σ^2-fixed	–	0.007	0.618	1.246	0.452
	MCD	0.012	0.005	0.663	1.235	0.453
	DE	0.067	0.003	0.631	1.226	0.464
	HTS	0.147	0.004	0.661	1.255	0.452
	HTS+MCD	0.143	0.015	0.836	1.263	0.452
	DUP	0.144	0.004	0.638	1.239	0.452
PL-EN	σ^2-fixed	–	0.027	0.586	1.518	0.264
	MCD	0.063	0.029	0.472	1.450	0.265
	DE	0.029	0.029	0.500	1.490	0.271
	HTS	0.045	0.025	0.609	1.530	0.264
	HTS+MCD	0.139	0.008	0.502	1.424	0.268
	DUP	0.048	0.025	0.604	1.519	0.264
PS-EN	σ^2-fixed	–	0.005	0.735	1.319	0.325
	MCD	0.028	0.006	0.740	1.291	0.327
	DE	0.040	0.004	0.732	1.295	0.330
	HTS	0.110	0.005	0.735	1.315	0.325
	HTS+MCD	0.111	0.013	0.849	1.315	0.297
	DUP	0.097	0.005	0.732	1.317	0.325
RU-EN	σ^2-fixed	–	0.031	0.454	1.574	0.288
	MCD	0.058	0.033	0.373	1.528	0.281
	DE	0.056	0.034	0.401	1.526	0.300
	HTS	0.087	0.030	0.464	1.575	0.288
	HTS+MCD	0.161	0.013	0.493	1.520	0.209
	DUP	0.073	0.029	0.467	1.570	0.288
TA-EN	σ^2-fixed	–	0.028	0.588	1.416	0.346
	MCD	0.047	0.022	0.567	1.357	0.346
	DE	0.058	0.024	0.585	1.410	0.357
	HTS	0.081	0.023	0.577	1.468	0.346
	HTS+MCD	0.250	0.016	0.642	1.300	0.284
	DUP	0.079	0.025	0.602	1.420	0.346
ZH-EN	σ^2-fixed	–	0.021	0.518	1.510	0.303
	MCD	0.051	0.020	0.447	1.458	0.302
	DE	0.054	0.022	0.451	1.481	0.310
	HTS	0.082	0.020	0.533	1.508	0.303
	HTS+MCD	0.186	0.006	0.504	1.377	0.278
	DUP	0.089	0.020	0.523	1.502	0.303
XX-EN	σ^2-fixed	–	0.023	0.529	1.448	0.334
	MCD	0.055	0.020	0.477	1.392	0.332
	DE	0.053	0.022	0.480	1.418	0.342
	HTS	0.090	0.021	0.546	1.455	0.334
	HTS+MCD	0.176	0.011	0.600	1.323	0.276
	DUP	0.089	0.021	0.542	1.442	0.334

Table 3: Results for segment-level DA prediction for Xx-En LPs. Underlined numbers indicate the best result for each evaluation metric in each language pair.

B MQM experiments

We provide extended results for each language pair in the MQM 2021 test set in Table 4.

We also present results without fine-tuning on the MQM data in Table 5, to facilitate comparisons. For these experiments we use the models trained on the WMT DA data (performance for these models is also reported in Tables 2 and 3). We can see

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-DE	σ^2-fixed	—	0.053	0.228	2.543	0.342
	MCD	0.132	0.026	0.228	1.984	0.391
	DE	0.075	0.057	0.155	2.911	0.422
	HTS	0.236	0.029	0.192	2.274	0.370
	HTS+MCD	0.232	0.025	0.280	1.841	0.365
	KL	0.251	0.052	0.168	2.641	0.391
	DUP	0.186	0.051	0.273	2.215	0.342
ZH-EN	σ^2-fixed	—	0.058	0.516	1.611	0.439
	MCD	0.253	0.009	0.500	1.219	0.590
	DE	0.157	0.045	0.420	1.381	0.601
	HTS	0.365	0.036	0.514	1.338	0.564
	HTS+MCD	0.380	0.035	0.566	1.248	0.570
	KL	0.348	0.037	0.515	1.353	0.547
	DUP	0.386	0.039	0.949	1.419	0.439
EN-RU	σ^2-fixed	—	0.053	0.338	2.217	0.340
	MCD	0.136	0.039	0.243	1.944	0.376
	DE	0.144	0.052	0.100	3.803	0.392
	HTS	0.308	0.059	0.104	3.317	0.377
	HTS+MCD	0.304	0.049	0.285	1.822	0.375
	KL	0.279	0.050	0.095	3.976	0.372
	DUP	0.260	0.029	0.608	1.783	0.340
AVG	σ^2-fixed	—	0.055	0.371	2.090	0.377
	MCD	0.179	0.024	0.334	1.686	0.460
	DE	0.128	0.051	0.236	2.631	0.479
	HTS	0.307	0.041	0.284	2.264	0.445
	HTS+MCD	0.311	0.037	0.388	1.614	0.445
	KL	0.296	0.046	0.273	2.595	0.443
	DUP	0.285	0.039	0.634	1.778	0.377

Table 4: Results for segment-level MQM predictions with fine-tuning on MQM 2020 data. Underlined numbers indicate the best result for each evaluation metric in each language pair.

that without further finetuning on MQM scores all models with the exception of the ones based on variance (**MCD** and **DE**) have a significant drop in performance.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-DE	MCD	0.134	0.069	1.019	0.577	0.295
	DE	0.104	0.021	1.03	0.644	0.332
	HTS	0.094	0.039	0.274	2.567	0.326
	HTS + MCD	0.126	0.021	0.356	1.502	0.291
	DUP	0.038	0.054	0.241	2.248	0.302
ZH-EN	MCD	0.115	0.081	1.321	0.956	0.441
	DE	0.14	0.025	1.143	0.911	0.457
	HTS	0.082	0.013	0.595	1.615	0.436
	HTS + MCD	-0.006	0.013	0.637	1.42	0.433
	DUP	0.17	0.05	0.469	1.814	0.434
EN-RU	MCD	0.14	0.069	1.242	0.563	0.306
	DE	0.117	0.078	1.332	0.684	0.318
	HTS	0.134	2.035	0.306	0.021	0.337
	HTS + MCD	-0.042	0.016	0.459	1.492	0.333
	DUP	0.139	0.045	0.35	2.238	0.290
AVG	MCD	0.356	0.129	0.722	0.074	1.215
	DE	0.377	0.123	0.763	0.042	1.179
	HTS	0.289	0.079	0.012	1.34	0.341
	HTS + MCD	0.286	-0.017	1.076	0.011	0.41
	DUP	0.272	0.115	1.489	0.035	0.306

Table 5: Results for segment-level MQM prediction without fine-tuning. Underlined numbers indicate the best result for each evaluation metric in each language pair.

C Model implementation and parameters

Table 6 shows the hyperparameters used to train the following uncertainty prediction models: **MCD**, **DE**, **HTS**, **KL** and **DUP**. For deep ensembles we trained 4 models with different seeds and as a fifth model we used the *wmt-comet-da* available at https://github.com/Unbabel/COMET (in the table we refer to it as **Vanilla COMET**).

D Performance indicators

We briefly describe below each of the metrics reported for the experiments of this paper, provide the formulas for each one and the motivation for using them. For all described metrics we assume access to a test set $\mathcal{D} = \{\langle s_j, t_j, \mathcal{R}_j, q_j^* \rangle\}_{j=1}^{|\mathcal{D}|}$, consisting of samples paired with their ground truth quality scores.

Calibration Error To estimate how well-calibrated the methods are we compute expected calibration error (ECE; Naeini et al. 2015; Kuleshov et al. 2018), which is defined as:

$$\text{ECE} = \frac{1}{M} \sum_{b=1}^{M} |\text{acc}(\gamma_b) - \gamma_b|, \tag{6}$$

where each b is a bin representing a confidence level γ_b, and $\text{acc}(\gamma_b)$ is the fraction of times the ground truth q^* falls inside the confidence interval $I(\gamma_b)$:

$$\text{acc}(\gamma_b) = \frac{1}{|\mathcal{D}|} \sum_{\langle s,t,\mathcal{R},q^* \rangle \in \mathcal{D}} \mathbb{1}(q^* \in I(\gamma_b)). \tag{7}$$

We use this metric with $M = 100$, similarly to previous works.

Negative log-likelihood The negative log-likelihood (NLL) captures both accuracy- and uncertainty-related performance, since it essentially considers the log-likelihood of the true quality score q^* based on the distribution estimated by the predicted variance (uncertainty). Thus it penalizes predictions that are accurate but have too high uncertainty (since they will become flat distributions with low probability everywhere), and even more severely incorrect predictions with high confidence, but is more lenient with predictions that are inaccurate but have high uncertainty.

$$\text{NLL} = -\frac{1}{|\mathcal{D}|} \sum_{\langle s,t,\mathcal{R},q^* \rangle \in \mathcal{D}} \log \hat{p}(q^* \mid \langle s, t, \mathcal{R} \rangle). \tag{8}$$

Hyperparameter	MCD/DE/Vanilla COMET	HTS/KL	DUP
Encoder Model	XLM-R (large)	XLM-R (large)	XLM-R (large)
Optimizer	Adam	Adam	Adam
No. frozen epochs	0.3	0.3	0.3
Learning rate	3e-05	3e-05	3e-05
Encoder Learning Rate	1e-05	1e-05	1e-05
Layerwise Decay	0.95	0.95	0.95
Batch size	4	4	4
Loss function	Mean squared error	$\mathcal{L}_{\text{HTS}}$ / $\mathcal{L}_{\text{KL}}$	$\mathcal{L}^{\text{E}}_{\text{HTS}}$ [$\mathcal{L}^{\text{E}}_{\text{ABS}}$ / $\mathcal{L}^{\text{E}}_{\text{SQ}}$]
Dropout	0.15	0.15	0.15
Hidden sizes	[3072, 1024]	[3072, 1024]	[3072, 1024]
Encoder Embedding layer	Frozen	Frozen	Frozen
Bottleneck layer size	-	-	256
FP precision	32	32	32
No. Epochs (training)	2	2	2
No. Epochs (fine-tuning)	1	1	1

Table 6: Hyperparameters used to train uncertainty prediction methods.

Note that it is possible to calculate the optimal fixed variance that minimizes NLL by:

$$\sigma^2_{\text{fixed}} = \frac{1}{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{D}|}(q^*_j - \hat{\mu}_j)^2. \quad (9)$$

Sharpness To ensure informative uncertainty estimation, confidence intervals should not only be calibrated, but also sharp. We measure sharpness using the predicted variance $\hat{\sigma}^2$, as defined in Kuleshov et al. (2018):

$$\text{sha}(\hat{p}_Q) = \frac{1}{|\mathcal{D}|}\sum_{\langle s,t,\mathcal{R}\rangle \in \mathcal{D}} \hat{\sigma}^2. \quad (10)$$

Pearson correlations The **predictive Pearson score** (PPS), evaluates the predictive accuracy of the system – it is the Pearson correlation $r(q^*, \hat{q})$ between the ground truth quality scores q^* and the system predictions $\hat{q}$ in the dataset $\mathcal{D}$. The **uncertainty Pearson score** (UPS) $r(|q^* - \hat{q}|, \hat{\sigma})$, measures the alignment between the prediction errors $|q^* - \hat{q}|$ and the uncertainty estimates $\hat{\sigma}$.

E Uncertainty on OOD examples

We provide the comparison of the sharpness value, representing the quantified uncertainty for in-domain (ID) data (WMT21 news data with MQM annotations) and out-of-domain (OOD) data (WMT21 TEDTalks data with MQM annotations) in Figure 5. Sharpness as explained in App. D, is an indicator of the overall estimated confidence of a model over a given dataset. Thus we want to examine whether the estimated confidence intervals for the OOD data are representative of the expected increase in epistemic uncertainty.

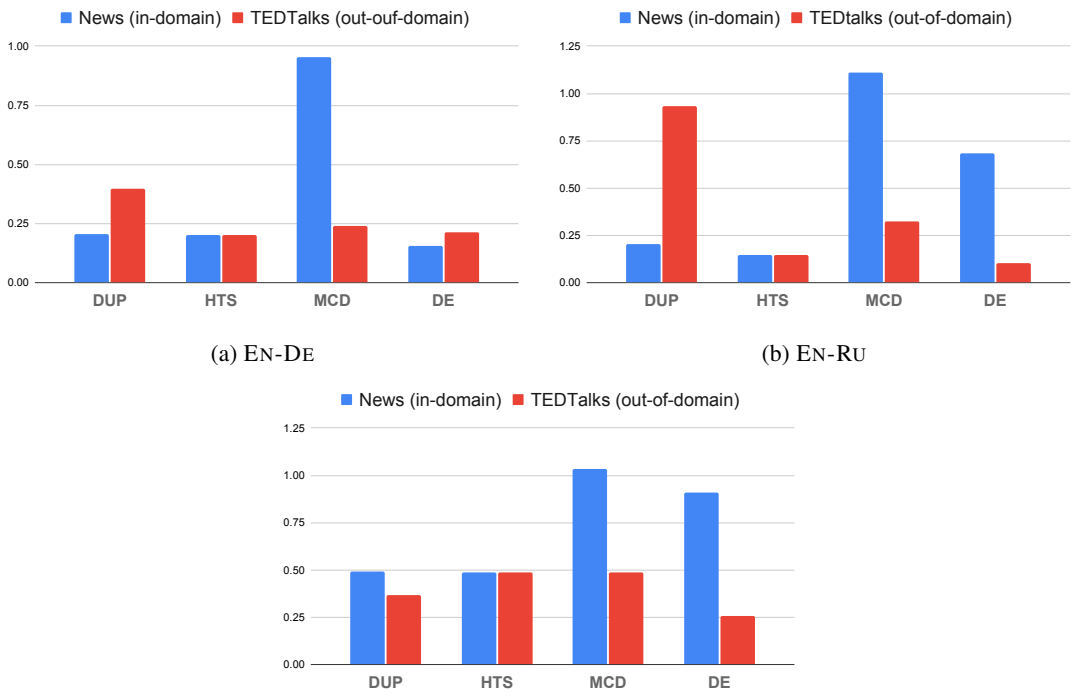
Looking at the sharpness variation per language pair, we can see that for EN-DE and EN-RU, where the aleatoric uncertainty is relatively low as indicated by the low HTS values, the sharpness increases significantly for the DUP model. This behaviour however does not hold for cases where aleatoric uncertainty is higher (ZH-EN). We speculate that this could be attributed to the fact that DUP is trained to capture total uncertainty, instead of only epistemic, and thus it is sensitive to increased noise in the data. Further experiments would be needed to verify this hypothesis.

Across language pairs, the values for HTS remain the same for ID and OOD, while for MCD we have the opposite effect than what was expected: sharpness drops significantly for OOD data in all language pairs. This further supports our claim that uncertainty predictors relying on model variance are not optimal to represent epistemic uncertainty.

For completeness we also provide the results for the rest of performance indicators on the TedTalk dats in Table 7. Note that for the OOD experiments we sampled half the dataset for testing and reserved the rest for calibration (resulting in approx. 4K segments per language pair for each split).

F Ablation tests for DUP

We present different ablation tests on the DUP architecture to compare the impact of different modelling choices on the training of the model. Our ablation tests are focusing on the second step model, $\mathcal{M}_{\text{E}}$, since it is the one that accounts for the uncertainty predictions.



(a) EN-DE

(b) EN-RU

(c) ZH-EN

Figure 5: Sharpness for in-domain (blue) News WMT21 MQM data and out-of-domain (red) TEDTalks WMT21 MQM data. We show changes in sharpness values on each language pair separately, for the **DUP**, **HTS**, **MCD** and **DE** models finetuned on News WMT20 MQM data.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-DE	σ^2-fixed	–	0.072	0.146	2.957	0.526
	MCD	0.178	0.052	0.147	2.500	0.540
	DE	0.371	0.062	0.314	1.977	0.571
	HTS	0.290	0.070	0.251	2.239	0.425
	HTS+MCD	0.401	0.073	0.227	1.756	0.545
	DE	0.346	0.058	0.346	2.219	0.526
EN-RU	σ^2-fixed	–	0.057	0.229	2.095	0.436
	MCD	0.086	0.065	0.238	1.846	0.425
	DE	0.271	0.057	0.346	1.679	0.441
	HTS	0.267	0.084	0.151	2.506	0.372
	HTS+MCD	0.293	0.068	0.402	1.473	0.387
	DUP	0.282	0.047	0.300	1.781	0.436
ZH-EN	σ^2-fixed	–	0.033	0.397	2.203	0.434
	MCD	0.063	0.023	0.283	2.348	0.447
	DE	0.23	0.036	0.586	1.865	0.456
	HTS	0.378	0.067	0.135	2.685	0.544
	HTS+MCD	0.288	0.073	0.223	2.276	0.425
	DUP	0.271	0.030	0.825	1.718	0.434

Table 7: Results for segment-level MQM predictions on TEDTalk data. Underlined numbers indicate the best result for each evaluation metric in each language pair.

F.1 Comparison of loss functions

We explore three different loss functions for the $\mathcal{M}_{\text{E}}$ model of **DUP**, described in Eqs. 11–13.

$$\mathcal{L}^{\text{E}}_{\text{ABS}}(\hat{\epsilon};\epsilon^*) = (\epsilon^* - \hat{\epsilon})^2 \quad (11)$$

$$\mathcal{L}^{\text{E}}_{\text{SQ}}(\hat{\epsilon};\epsilon^*) = ((\epsilon^*)^2 - \hat{\epsilon}^2)^2 \quad (12)$$

$$\mathcal{L}^{\text{E}}_{\text{HTS}}(\hat{\epsilon};\epsilon^*) = \frac{(\epsilon^*)^2}{2\hat{\epsilon}^2} + \frac{1}{2}\log(\hat{\epsilon})^2. \quad (13)$$

Losses $\mathcal{L}^{\text{E}}_{\text{ABS}}$ and $\mathcal{L}^{\text{E}}_{\text{SQ}}$ are variations of the mean squared error loss, using as argument either the absolute error $\hat{\epsilon}$ or the squared error $\hat{\epsilon}^2$.

We compare the performance of DUP models trained using the different losses on the segment-level DA data. According to the results in Table 8, all three losses perform similarly, with a slight advantage to $\mathcal{L}^{\text{E}}_{\text{HTS}}$. This motivated our choice to run the experiments discussed in the main paper using this loss as a representative of **DUP**.

F.2 Comparison of parameter sharing settings

For this paper the models used for $\mathcal{M}_{\text{Q}}$ and $\mathcal{M}_{\text{E}}$ use very similar architectures, except for the bottleneck layer, as depicted in Figure 6. We thus compare the impact of three different settings:

1. **NS**: Not sharing any parameters and training $\mathcal{M}_{\text{E}}$ from scratch.

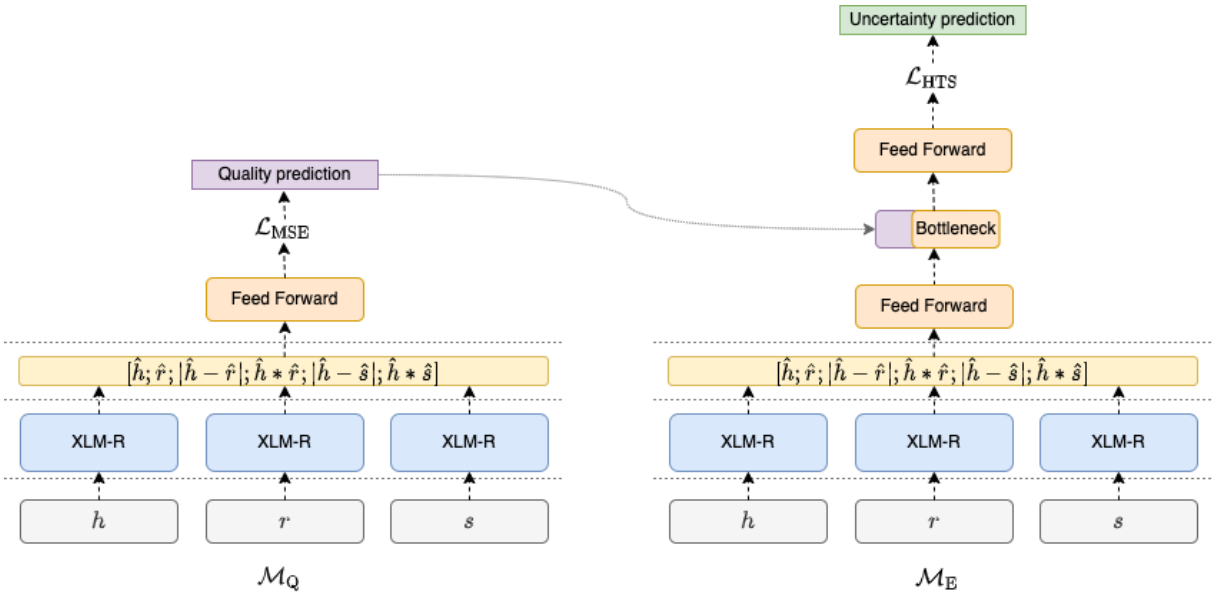


Figure 6: Architecture and dependencies of DUP $\mathcal{M}_{\text{Q}}$ and $\mathcal{M}_{\text{E}}$ models

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-XX	DUP $\mathcal{L}^{\text{E}}_{\text{ABS}}$	0.134	0.013	0.295	1.019	0.633
	DUP $\mathcal{L}^{\text{E}}_{\text{SQ}}$	0.140	0.012	0.315	1.022	0.633
	DUP $\mathcal{L}^{\text{E}}_{\text{HTS}}$	0.146	0.014	0.293	1.021	0.633
XX-EN	DUP $\mathcal{L}^{\text{E}}_{\text{ABS}}$	0.081	0.017	0.527	1.471	0.287
	DUP $\mathcal{L}^{\text{E}}_{\text{SQ}}$	0.084	0.017	0.534	1.470	0.287
	DUP $\mathcal{L}^{\text{E}}_{\text{HTS}}$	0.086	0.017	0.524	1.473	0.287
AVG	DUP $\mathcal{L}^{\text{E}}_{\text{ABS}}$	0.104	1.265	0.015	0.414	0.446
	DUP $\mathcal{L}^{\text{E}}_{\text{SQ}}$	0.108	1.262	0.014	0.427	
	DUP $\mathcal{L}^{\text{E}}_{\text{HTS}}$	0.112	1.266	0.015	0.411	0.446

Table 8: Comparison of different losses for the DUP method in segment-level DA prediction.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-XX	NS	0.272	0.008	0.344	0.919	0.566
	S	0.285	0.007	0.320	0.910	0.566
	SF	0.276	0.008	0.341	0.917	0.566
XX-EN	NS	0.090	0.021	0.546	1.455	0.334
	S	0.089	0.021	0.542	1.442	0.334
	SF	0.093	0.020	0.550	1.462	0.334
AVG	NS	0.177	0.015	0.450	1.201	0.444
	S	0.182	0.014	0.437	1.190	0.444
	SF	0.180	0.015	0.451	1.204	0.444

Table 9: Comparison of different parameter share configurations for DUP.

2. **S**: Sharing all (common) parameters between $\mathcal{M}_{\text{Q}}$ and $\mathcal{M}_{\text{E}}$; then keep fine-tuning $\mathcal{M}_{\text{E}}$ on the new uncertainty (error) prediction task.

3. **SF**: Sharing all (common) parameters between $\mathcal{M}_{\text{Q}}$ and $\mathcal{M}_{\text{E}}$ and freeze the XLM-R encoder weights and embeddings; then keep fine-tuning the rest of the $\mathcal{M}_{\text{E}}$ parameters on the new uncertainty (error) prediction task.

The results are presented in Table 9. We see that sharing parameters (**S**, **SF** settings) consistently results in a small boost for all uncertainty indicators. Since we do not see a significant further improvement by keeping the encoder frozen (**SF**), we perform the rest of the experiments presented in this work by simply sharing the parameters between $\mathcal{M}_{\text{Q}}$ and $\mathcal{M}_{\text{E}}$ (**S** setting).

G Results on other metrics

In this section we present results on the WMT 20 DA dataset using trainable metrics that differ to the COMET architecture, as an additional comparison. We select BLEURT and UniTE for this comparison. BLEURT (Sellam et al., 2020b), is a multilingual metric with high performance, which unlike COMET jointly encodes only the translation and reference inputs in order to predict the quality score of a segment. UniTE is a newly proposed architecture (Wan et al., 2022) which is taking into account three different input combinations with the translation segment, namely reference-only, source-only and source-reference-combined. Note that we do not optimise the hyper-parameters of these metrics since we are only interested in comparing the overall behaviour. Hence, improved results could be expected upon optimisation.

We present results on BLEURT in Tables 10, 11 and 12. We used a BLEURT implementation with RemBERT encoder (Chung et al., 2021), trained on the same DA setup described in §5 of the main paper. We notice that we observe similar behaviour of the proposed uncertainty predictors to the one identified for COMET, with the exception of the heteroscedastic predictors (**HTS**, **HTS+MCD**). It seems that without access to the source segments it is harder for the heteroscedastic approach to learn to predict meaningful variance intervals. In other words, it seems to be harder for HTS approaches to identify noisy inputs relying only on the reference segments. This finding highlights the importance of including source segments towards identification of noisy inputs and prediction of segment level quality with higher confidence. In comparison, we can see that the **DUP** approach significantly improves the uncertainty correlation (UPS).

For UniTE, we show results in Tables 13, 14 and 15. We implemented UniTE with an InfoXLM encoder (Chi et al., 2021), trained on the same DA setup described in §5 of the main paper. We noticed that on average the correlations achieved for the UPS performance indicator are lower to the ones obtained with COMET and BLEURT, especially for **MCD**. However, they follow similar pattern to the one identified in the main paper: we obtain significantly better correlations both for the HTS and the DUP predictors.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-XX	σ^2-fixed	–	0.107	4.698	1.984	0.317
	MCD	0.302	0.295	1.035	2.291	0.316
	DE	0.204	0.205	1.886	2.034	0.337
	HTS	0.328	0.104	4.383	1.925	0.319
	HTS+MCD	0.322	0.103	4.364	1.924	0.314
	DUP	0.435	0.105	5.917	2.033	0.317
XX-EN	σ^2-fixed	–	0.067	3.200	1.831	0.080
	MCD	0.274	0.204	1.030	1.965	0.079
	DE	0.032	0.107	1.392	1.845	0.111
	HTS	0.246	0.069	2.755	1.739	0.071
	HTS+MCD	0.240	0.069	2.753	1.739	0.068
	DUP	0.320	0.064	3.892	1.888	0.080
AVG	σ^2-fixed	–	0.086	3.910	1.903	0.192
	MCD	0.287	0.247	1.032	2.120	0.191
	DE	0.172	0.150	1.608	1.926	0.210
	HTS	0.285	0.086	3.527	1.827	0.189
	HTS+MCD	0.279	0.085	3.517	1.827	0.185
	DUP	0.374	0.083	4.853	1.957	0.192

Table 10: Results for segment-level DA predictions by BLEURT [**Average across language pairs**]. Underlined numbers indicate the best result for each evaluation metric in each language pair.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-CS	σ^2-fixed	–	0.083	3.462	1.860	0.400
	MCD	0.226	0.235	1.031	2.000	0.399
	DE	0.326	0.161	1.656	1.846	0.446
	HTS	0.291	0.071	3.727	1.848	0.386
	HTS+MCD	0.289	0.071	3.725	1.852	0.381
	DUP	0.410	0.079	5.029	1.986	0.400
EN-DE	σ^2-fixed	–	0.112	5.779	2.099	0.257
	MCD	0.432	0.336	1.040	2.603	0.256
	DE	0.198	0.217	2.508	2.207	0.267
	HTS	0.366	0.129	5.959	2.093	0.254
	HTS+MCD	0.358	0.128	5.846	2.083	0.248
	DUP	0.513	0.123	7.362	2.115	0.257
EN-JA	σ^2-fixed	–	0.134	6.040	2.105	0.277
	MCD	0.326	0.358	1.041	2.592	0.275
	DE	0.121	0.258	1.937	2.185	0.289
	HTS	0.278	0.126	5.105	2.003	0.293
	HTS+MCD	0.274	0.125	5.070	2.000	0.287
	DUP	0.420	0.124	7.295	2.122	0.277
EN-PL	σ^2-fixed	–	0.081	3.739	1.904	0.349
	MCD	0.217	0.247	1.032	2.105	0.347
	DE	0.247	0.154	1.728	1.876	0.374
	HTS	0.262	0.070	3.779	1.870	0.341
	HTS+MCD	0.260	0.069	3.739	1.870	0.335
	DUP	0.362	0.076	5.092	1.992	0.349
EN-RU	σ^2-fixed	–	0.100	4.321	1.959	0.356
	MCD	0.218	0.282	1.034	2.212	0.356
	DE	0.158	0.184	1.838	1.951	0.361
	HTS	0.280	0.097	4.019	1.896	0.350
	HTS+MCD	0.275	0.096	4.010	1.895	0.345
	DUP	0.383	0.103	5.177	1.967	0.356
EN-TA	σ^2-fixed	–	0.065	3.530	1.877	0.240
	MCD	0.131	0.218	1.029	2.047	0.240
	DE	0.162	0.135	1.822	1.864	0.278
	HTS	0.320	0.062	3.499	1.826	0.248
	HTS+MCD	0.317	0.062	3.534	1.828	0.249
	DUP	0.279	0.060	4.501	1.966	0.240
EN-ZH	σ^2-fixed	–	0.154	6.063	2.097	0.280
	MCD	0.506	0.375	1.041	2.540	0.280
	DE	0.171	0.297	1.940	2.297	0.282
	HTS	0.465	0.160	4.919	1.975	0.301
	HTS+MCD	0.449	0.160	4.920	1.974	0.297
	DUP	0.601	0.154	7.044	2.091	0.280

Table 11: Results for segment-level DA predictions by BLEURT for En-Xx LPs. Underlined numbers indicate the best result for each evaluation metric in each language pair.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
CS-EN	σ^2-fixed	–	0.068	3.085	1.816	0.061
	MCD	0.303	0.202	1.030	1.924	0.059
	DE	0.020	0.156	1.271	1.806	0.071
	HTS	0.291	0.073	2.775	1.735	0.052
	HTS+MCD	0.287	0.072	2.773	1.734	0.050
	DUP	0.360	0.068	3.781	1.838	0.061
DE-EN	σ^2-fixed	–	0.080	3.053	1.808	0.018
	MCD	0.364	0.210	1.028	1.908	0.016
	DE	0.087	0.159	1.320	1.767	0.044
	HTS	0.326	0.091	3.015	1.748	-0.002
	HTS+MCD	0.323	0.091	3.000	1.746	-0.003
	DUP	0.399	0.081	3.981	1.837	0.018
JA-EN	σ^2-fixed	–	0.072	3.701	1.896	0.095
	MCD	0.228	0.229	1.033	2.082	0.094
	DE	0.011	0.143	1.524	1.845	0.102
	HTS	0.192	0.069	2.973	1.779	0.084
	HTS+MCD	0.184	0.068	2.948	1.777	0.082
	DUP	0.262	0.066	4.328	1.959	0.095
KM-EN	σ^2-fixed	–	0.032	2.613	1.753	0.208
	MCD	0.084	0.145	1.029	1.827	0.205
	DE	0.020	0.002	1.207	2.003	0.278
	HTS	0.100	0.026	2.168	1.642	0.209
	HTS+MCD	0.099	0.026	2.152	1.640	0.199
	DUP	0.149	0.021	3.222	1.856	0.208
PL-EN	σ^2-fixed	–	0.068	3.324	1.853	0.053
	MCD	0.313	0.213	1.031	2.002	0.053
	DE	0.051	0.128	1.536	1.816	0.065
	HTS	0.253	0.069	2.842	1.766	0.057
	HTS+MCD	0.248	0.069	2.862	1.767	0.051
	DUP	0.358	0.067	3.922	1.919	0.053
PS-EN	σ^2-fixed	–	0.034	2.633	1.768	0.090
	MCD	0.176	0.151	1.029	1.861	0.088
	DE	-0.029	0.002	1.267	2.134	0.101
	HTS	0.165	0.033	2.058	1.647	0.075
	HTS+MCD	0.159	0.033	2.044	1.647	0.070
	DUP	0.218	0.028	3.267	1.859	0.090
RU-EN	σ^2-fixed	–	0.085	3.490	1.857	0.076
	MCD	0.261	0.228	1.030	1.986	0.076
	DE	0.027	0.153	1.482	1.815	0.084
	HTS	0.219	0.087	3.135	1.786	0.068
	HTS+MCD	0.212	0.087	3.148	1.785	0.067
	DUP	0.319	0.081	4.021	1.897	0.076
TA-EN	σ^2-fixed	–	0.041	2.069	1.658	0.108
	MCD	0.238	0.121	1.020	1.688	0.106
	DE	0.023	0.064	1.401	1.601	0.132
	HTS	0.245	0.046	1.704	1.561	0.098
	HTS+MCD	0.238	0.046	1.703	1.562	0.096
	DUP	0.273	0.034	2.959	1.767	0.108
ZH-EN	σ^2-fixed	–	0.076	3.542	1.886	0.080
	MCD	0.309	0.231	1.033	2.075	0.079
	DE	0.034	0.138	1.527	1.817	0.097
	HTS	0.278	0.078	2.984	1.790	0.071
	HTS+MCD	0.270	0.078	2.984	1.790	0.068
	DUP	0.353	0.075	4.252	1.932	0.080

Table 12: Results for segment-level DA predictions by BLEURT for Xx-En LPs. Underlined numbers indicate the best result for each evaluation metric in each language pair.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-XX	σ^2-fixed	–	0.018	0.224	0.913	0.650
	MCD	0.022	0.011	0.260	0.853	0.603
	DE	0.129	0.015	0.264	0.878	0.647
	HTS	0.183	0.013	0.319	0.892	0.630
	HTS+MCD	0.175	0.004	0.272	0.792	0.587
	DUP	0.139	0.015	0.250	0.924	0.650
XX-EN	σ^2-fixed	–	0.027	0.498	1.512	0.289
	MCD	0.059	0.016	0.523	1.352	0.271
	DE	0.049	0.028	0.503	1.488	0.293
	HTS	0.090	0.026	0.525	1.506	0.288
	HTS+MCD	0.079	0.010	0.572	1.351	0.267
	DUP	0.065	0.026	0.518	1.546	0.289
AVG	σ^2-fixed	–	0.023	0.368	1.228	0.460
	MCD	0.041	0.014	0.398	1.115	0.428
	DE	0.087	0.022	0.390	1.198	0.461
	HTS	0.134	0.020	0.427	1.215	0.450
	HTS+MCD	0.124	0.007	0.429	1.085	0.419
	DUP	0.100	0.021	0.391	1.251	0.460

Table 13: Results for segment-level DA predictions by UniTE [**Average across language pairs**]. Underlined numbers indicate the best result for each evaluation metric in each language pair.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
EN-CS	σ^2-fixed	–	0.013	0.284	0.990	0.735
	MCD	-0.007	0.010	0.242	0.803	0.672
	DE	0.108	0.011	0.348	0.946	0.732
	HTS	0.151	0.009	0.427	0.961	0.718
	HTS+MCD	0.132	0.003	0.249	0.743	0.672
	DUP	0.108	0.011	0.300	0.988	0.735
EN-DE	σ^2-fixed	–	0.038	0.145	1.178	0.623
	MCD	0.036	0.017	0.227	0.890	0.579
	DE	0.172	0.032	0.191	1.096	0.623
	HTS	0.262	0.030	0.281	1.064	0.603
	HTS+MCD	0.283	0.008	0.236	0.798	0.578
	DUP	0.241	0.031	0.207	1.213	0.623
EN-JA	σ^2-fixed	–	0.011	0.160	0.718	0.688
	MCD	-0.015	0.008	0.241	0.775	0.650
	DE	0.112	0.008	0.181	0.659	0.698
	HTS	0.117	0.008	0.196	0.705	0.687
	HTS+MCD	0.148	0.004	0.229	0.684	0.646
	DUP	0.089	0.008	0.178	0.728	0.688
EN-PL	σ^2-fixed	–	0.014	0.344	1.124	0.647
	MCD	-0.091	0.011	0.299	0.935	0.605
	DE	0.121	0.011	0.426	1.074	0.652
	HTS	0.127	0.014	0.481	1.086	0.626
	HTS+MCD	0.141	0.004	0.345	0.907	0.591
	DUP	0.060	0.013	0.364	1.133	0.647
EN-RU	σ^2-fixed	–	0.026	0.229	1.001	0.610
	MCD	-0.059	0.015	0.248	0.865	0.587
	DE	0.112	0.020	0.284	1.017	0.611
	HTS	0.129	0.013	0.357	1.056	0.600
	HTS+MCD	0.148	0.009	0.273	0.890	0.576
	DUP	0.085	0.022	0.256	1.040	0.610
EN-TA	σ^2-fixed	–	0.011	0.334	1.125	0.685
	MCD	0.079	0.005	0.396	1.052	0.647
	DE	0.120	0.018	0.335	1.089	0.688
	HTS	0.228	0.015	0.390	1.096	0.678
	HTS+MCD	0.207	0.003	0.410	1.032	0.634
	DUP	0.164	0.009	0.371	1.110	0.685
EN-ZH	σ^2-fixed	–	0.019	0.101	0.505	0.571
	MCD	0.177	0.011	0.214	0.760	0.504
	DE	0.162	0.012	0.109	0.486	0.546
	HTS	0.272	0.009	0.134	0.494	0.522
	HTS+MCD	0.204	0.002	0.213	0.622	0.453
	DUP	0.234	0.014	0.125	0.508	0.571

Table 14: Results for segment-level DA predictions by UniTE for En-Xx LPs. Underlined numbers indicate the best result for each evaluation metric in each language pair.

		UPS ↑	ECE ↓	Sha. ↓	NLL ↓	PPS ↑
CS-EN	σ^2-fixed	–	0.027	0.507	1.485	0.176
	MCD	0.021	0.008	0.531	1.296	0.157
	DE	0.016	0.029	0.491	1.454	0.179
	HTS	0.049	0.029	0.474	1.462	0.171
	HTS+MCD	0.037	0.005	0.563	1.273	0.150
	DUP	0.032	0.027	0.510	1.490	0.176
DE-EN	σ^2-fixed	–	0.037	0.294	1.492	0.551
	MCD	0.109	0.018	0.337	1.242	0.532
	DE	0.089	0.033	0.320	1.437	0.558
	HTS	0.116	0.028	0.390	1.443	0.554
	HTS+MCD	0.117	0.008	0.434	1.247	0.526
	DUP	0.093	0.036	0.303	1.532	0.551
JA-EN	σ^2-fixed	–	0.018	0.518	1.360	0.315
	MCD	0.111	0.005	0.647	1.289	0.288
	DE	0.073	0.022	0.498	1.366	0.319
	HTS	0.128	0.018	0.547	1.425	0.310
	HTS+MCD	0.116	0.008	0.669	1.281	0.280
	DUP	0.100	0.014	0.547	1.420	0.315
KM-EN	σ^2-fixed	–	0.006	0.689	1.251	0.428
	MCD	0.087	0.015	0.773	1.272	0.407
	DE	0.040	0.003	0.650	1.253	0.425
	HTS	0.125	0.003	0.659	1.257	0.416
	HTS+MCD	0.124	0.016	0.944	1.311	0.396
	DUP	0.096	0.005	0.816	1.282	0.428
PL-EN	σ^2-fixed	–	0.033	0.523	1.587	0.196
	MCD	0.011	0.022	0.484	1.408	0.178
	DE	0.048	0.032	0.551	1.546	0.202
	HTS	0.061	0.034	0.545	1.552	0.199
	HTS+MCD	0.053	0.012	0.523	1.409	0.181
	DUP	0.039	0.032	0.545	1.612	0.196
PS-EN	σ^2-fixed	–	0.003	0.792	1.350	0.260
	MCD	0.064	0.011	0.861	1.333	0.245
	DE	0.022	0.004	0.768	1.342	0.264
	HTS	0.080	0.005	0.772	1.353	0.259
	HTS+MCD	0.062	0.012	0.970	1.364	0.243
	DUP	0.068	0.003	0.803	1.349	0.260
RU-EN	σ^2-fixed	–	0.041	0.368	1.664	0.223
	MCD	0.048	0.023	0.411	1.392	0.210
	DE	0.064	0.041	0.385	1.658	0.226
	HTS	0.102	0.040	0.404	1.649	0.225
	HTS+MCD	0.088	0.011	0.455	1.351	0.209
	DUP	0.056	0.040	0.395	1.700	0.223
TA-EN	σ^2-fixed	–	0.031	0.607	1.481	0.322
	MCD	0.038	0.022	0.609	1.351	0.302
	DE	0.035	0.033	0.605	1.470	0.327
	HTS	0.073	0.026	0.648	1.568	0.320
	HTS+MCD	0.065	0.010	0.627	1.357	0.298
	DUP	0.061	0.030	0.599	1.592	0.322
ZH-EN	σ^2-fixed	–	0.026	0.468	1.587	0.258
	MCD	0.062	0.017	0.455	1.417	0.242
	DE	0.042	0.027	0.484	1.552	0.262
	HTS	0.089	0.027	0.504	1.556	0.257
	HTS+MCD	0.071	0.011	0.480	1.420	0.238
	DUP	0.064	0.025	0.480	1.603	0.258

Table 15: Results for segment-level DA predictions by UniTE for Xx-En LPs. Underlined numbers indicate the best result for each evaluation metric in each language pair.