COMETKIWI: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task

Ricardo Rei[*1,2,4], **Marcos Treviso**[*3,4], **Nuno M. Guerreiro**[*3,4], **Chrysoula Zerva**[*3,4], **Ana C. Farinha**[1], **Christine Maroti**[1], **José G. C. de Souza**[1], **Taisiya Glushkova**[3,4], **Duarte M. Alves**[1,4], **Alon Lavie**[1], **Luisa Coheur**[2,4], **André F. T. Martins**[1,3,4]

[1]Unbabel, Lisbon, Portugal, [2]INESC-ID, Lisbon, Portugal
[3]Instituto de Telecomunicações, Lisbon, Portugal
[4]Instituto Superior Técnico, University of Lisbon, Portugal

Abstract

We present the joint contribution of IST and Unbabel to the WMT 2022 Shared Task on Quality Estimation (QE). Our team participated on all three subtasks: (i) Sentence and Word-level Quality Prediction; (ii) Explainable QE; and (iii) Critical Error Detection. For all tasks we build on top of the COMET framework, connecting it with the predictor-estimator architecture of OPENKIWI, and equipping it with a word-level sequence tagger and an explanation extractor. Our results suggest that incorporating references during pretraining improves performance across several language pairs on downstream tasks, and that jointly training with sentence and word-level objectives yields a further boost. Furthermore, combining attention and gradient information proved to be the top strategy for extracting good explanations of sentence-level QE models. Overall, our submissions achieved the best results for all three tasks for almost all language pairs by a considerable margin.[1]

1 Introduction

Quality Estimation (QE) is the task of automatically assigning a quality score to a machine translation output without depending on reference translations (Specia et al., 2018). In this paper, we describe the joint contribution of Instituto Superior Técnico (IST) and Unbabel to the WMT22 Quality Estimation shared task (Zerva et al., 2022), where systems were submitted to three tasks: (i) Sentence and Word-level Quality Prediction; (ii) Explainable QE; and (iii) Critical Error Detection.

This year, we leverage the similarity between the tasks of MT evaluation and QE and bring together the strengths of two frameworks, COMET (Rei et al., 2020), which has been originally developed for reference-based MT evaluation, and OPENKIWI (Kepler et al., 2019), which has been developed for word-level and sentence-level QE. Namely, we implement some of the features of the latter, as well as other new features, into the COMET framework. The result is COMETKIWI, which links the predictor-estimator architecture with COMET training-style, and incorporates word-level sequence tagging.

Given that some language pairs (LPs) in the test set were not present in the training data, we aimed at developing QE systems that achieve good multilingual generalization and that are flexible enough to account for unseen languages through few-shot training. To do so, we start by pretraining our QE models on Direct Assessments (DAs) annotations from the previous year's Metrics shared task as it was shown to be beneficial in our previous submission (Zerva et al., 2021). Then we fine-tune our models with the data made available by the shared task.[2] We experimented with different pretrained multilingual transformers as the backbones of our models, and we developed new explainability methods to interpret them. We describe our systems and their training strategies in Section 3. Overall, our main contributions are:

- We combine the strengths of COMET and OPENKIWI, leading to COMETKIWI, a model that adopts COMET training features useful for multilingual generalization along with the predictor-estimator architecture of OPENKIWI.
- Following our previous work (Zerva et al., 2021), we show the importance of pretraining QE models on annotations from the Metrics shared task.
- We show that we can improve results for new LPs with only 500 examples without harming correlations for other LPs.
- We propose a new interpretability method that uses attention and gradient information along

[*]Equal contribution. ✉ ricardo.rei@unbabel.com
[1]https://github.com/Unbabel/COMET

[2]For zero-shot LPs we use 500 training examples which means we turn it into a few-shot setting. The only exception is English→Yoruba which was kept zero-shot.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 634 - 645
December 7-8, 2022

with a head-level scalar mix module that further refines the relevance of attention heads.

Our submitted systems achieve the best multilingual results on all tracks by a considerable margin: for sentence-level DA our system achieved a 0.572 Spearman correlation (+7% than the second best system); for word-level our system achieved a 0.341 MCC score (+2.4% than the second best system); and for Explainable QE our system achieved 0.486 R@K score (+10% than the second best system). The official results for all LPs are presented in Table 6 in the appendix.

2 Background

Quality Estimation. QE systems are usually designed according to the granularity in which predictions are made, such as sentence and word-level. In sentence-level QE, the goal is to predict a single quality score $\hat{y} \in \mathbb{R}$ given the whole source and its translation as input. Word-level QE works in a lower granularity level, with the goal of predicting binary quality labels $\hat{y}_i \in \{\text{OK}, \text{BAD}\}$ for all $1 \leq i \leq n$ *machine-translated words*, indicating whether that word is a translation error or not.

Transformers. The multi-head attention mechanism is the key component in transformers, being responsible for contextualizing the information within and across input sentences (Vaswani et al., 2017). Concretely, given as input a matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$ containing d-dimensional representations for n queries, and matrices $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{m \times d}$ for m keys and values, the *scaled dot-product attention* at a single head is computed as:

$$\mathsf{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \pi \underbrace{\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)}_{\boldsymbol{Z} \in \mathbb{R}^{n \times m}} \boldsymbol{V} \in \mathbb{R}^{n \times d}. \quad (1)$$

The π transformation maps rows to distributions, with $\mathsf{softmax}$ being the most common choice, $\pi(\boldsymbol{Z})_{ij} = \mathsf{softmax}(\boldsymbol{z}_i)_j$. Multi-head attention is computed by evoking Eq. 1 in parallel for each head h:

$$\mathsf{head}_h(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathsf{att}(\boldsymbol{Q}\boldsymbol{W}_h^Q, \boldsymbol{K}\boldsymbol{W}_h^K, \boldsymbol{V}\boldsymbol{W}_h^V),$$

where $\boldsymbol{W}_h^Q, \boldsymbol{W}_h^K, \boldsymbol{W}_h^V$ are learnable linear transformations. Finally, the output of the multi-head attention module at the ℓ-th layer is a set of hidden states $\boldsymbol{H}_\ell \in \mathbb{R}^{n \times d}$ formed via the concatenation of all $\boldsymbol{h}_{\ell,1}, ..., \boldsymbol{h}_{\ell,H}$ heads in that layer followed by a learnable linear transformation $\boldsymbol{W}^O$:

$$\boldsymbol{H}_\ell = \mathsf{concat}(\boldsymbol{h}_{\ell,1}, ..., \boldsymbol{h}_{\ell,H})\boldsymbol{W}^O.$$

The hidden states are further refined through position-wise feed-forward blocks and residual connections to obtain a final representation: $\boldsymbol{H}_\ell = \mathsf{FFN}(\boldsymbol{H}_\ell) + \boldsymbol{H}_\ell$. Transformers with only encoder-blocks, such as BERT (Devlin et al., 2019) and XLM (Conneau et al., 2020), have only the encoder self-attention, and thus $m = n$.

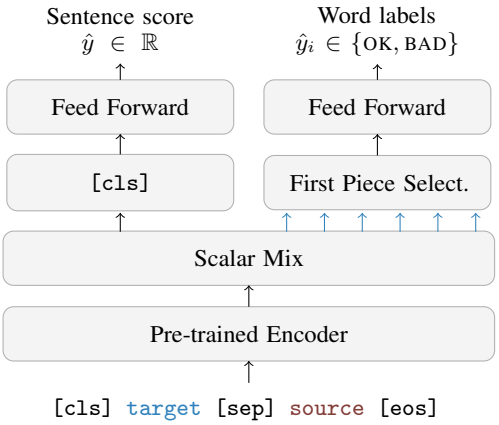

Figure 1: General architecture of CometKiwi for sentence-level (left part) and word-level QE (right part).

3 Implemented Systems

The overall architecture of our models is shown in Figure 1. The machine translated sentence $\boldsymbol{t} = \langle t_1, ..., t_n \rangle$ and its source sentence counterpart $\boldsymbol{s} = \langle s_1, ..., s_m \rangle$ are concatenated and passed as input to the encoder, which produces d-dimensional hidden state vectors $\boldsymbol{H}_0, ..., \boldsymbol{H}_L$ for each layer $0 \leq \ell \leq L$, where $\boldsymbol{H}_i \in \mathbb{R}^{(n+m) \times d}$, where $\ell = 0$ corresponds to the embedding layer. Next, all hidden states are fed to a scalar mix module (Peters et al., 2018) that learns a weighted sum of the hidden states of each layer of the encoder, producing a new sequence of aggregated hidden states $\boldsymbol{H}_{\text{mix}}$ as follows:

$$\boldsymbol{H}_{\text{mix}} = \lambda \sum_{\ell=0}^{L} \beta_\ell \boldsymbol{H}_\ell, \quad (2)$$

where λ is a scalar trainable parameter, $\boldsymbol{\beta} \in \triangle^L$, is given by $\boldsymbol{\beta} = \mathsf{sparsemax}(\boldsymbol{\phi})$ using a sparse transformation (Martins and Astudillo, 2016), with $\boldsymbol{\phi} \in \mathbb{R}^L$ as learnable parameters and $\triangle^L := \{\boldsymbol{\beta} \in \mathbb{R}^L : \mathbf{1}^\top \boldsymbol{\beta} = 1, \boldsymbol{\beta} \geq 0\}$[3].

[3] As it has been shown in (Rei et al., 2022) not all layers are relevant and thus, using $\mathsf{sparsemax}$ we learn to ignore layers

For sentence-level models, the hidden state of the first token (<cls>) is used as sentence representation $\boldsymbol{H}_{\text{mix},0} \in \mathbb{R}^d$, which, in turn, is passed to a 2-layered feed-forward module in order to get a sentence score prediction $\hat{y} \in \mathbb{R}$. For word-level models, we first retrieve the hidden state vectors associated with the first word piece of each machine translated token, and then pass them to a linear projection to get word-level predictions $\hat{y}_i \in \{\text{OK}, \text{BAD}\}$, $\forall_{1 \leq i \leq n}$. Moreover, attention matrices $\boldsymbol{A}_{1,1}, ..., \boldsymbol{A}_{L,H}$ for all layers and heads are also recovered as a by-product of the forward propagation.

Pretraining on Metrics Data. Every year, the WMT News Translation shared task organizers collect human judgments in the form of DAs. The collective corpora of 2017, 2018, and 2019 contain 24 LPs and a total of 657k samples with source, target, reference, and DA score. We follow our experiments from last year (Zerva et al., 2021) and start by pretraining our QE models on this data using the learning objective proposed by UniTE (Wan et al., 2022), which incorporates reference translations into training and thus acts as data augmentation.

Setting pretrained transformers as encoders. We follow the recent trend (Kepler et al., 2019; Ranasinghe et al., 2020) and experiment with three different pretrained multilingual transformers as the encoder layer of our models: XLM-R Large (Conneau et al., 2020),[4] InfoXLM Large (Chi et al., 2021),[5] and RemBERT (Chung et al., 2021).[6] XLM-R and InfoXLM consist of 24 encoder blocks with 16 attention heads each, whereas RemBERT has 32 encoder blocks with 18 attention heads each.

3.1 Task 1: Quality prediction

After the pretraining phase, we adapt our models to the released QE data using source and translation (i.e., in this phase we do not include references) to the different type of quality assessments provided, namely, DA and HTER[7] from the MLQE-PE corpus (Fomicheva et al., 2022) and Multidimensional Quality Metrics (MQM) annotations from WMT 2020 and 2021 (Freitag et al., 2021a,b).

3.1.1 Sentence-level quality prediction

For the sentence-level QE task we consider a multi-task setting (using sentence scores alongside supervision from OK/BAD tags) and the sentence-level only setting, with supervision only from the sentence-level quality assessment y. We found that adding the word-level supervision was beneficial for models built on top of InfoXLM. For the sentence-level supervision we used both DA and MQM scores. In this multi-task setting we use a combined loss as described in Eq. 5:

$$\mathcal{L}_{\text{sent}}(\theta) = \frac{1}{2}(y - \hat{y}(\theta))^2 \quad (3)$$

$$\mathcal{L}_{\text{word}}(\theta) = -\frac{1}{n}\sum_{i=1}^{n} w_{y_i} \log p_\theta(y_i) \quad (4)$$

$$\mathcal{L}(\theta) = \lambda_s \mathcal{L}_{\text{sent}}(\theta) + \lambda_w \mathcal{L}_{\text{word}}(\theta), \quad (5)$$

where $w \in \mathbb{R}^2$ represents the class weights given for OK and BAD tags, and λ_s, λ_w are used to weigh the combination of the sentence and word-level losses, respectively. Note that $\lambda_s = 1$ and $\lambda_w = 0$ yields a fully sentence-level model.

Few-shot language adaptation. Since in this shared task submissions are tested on 5 LPs for which there is no official training data (*km-en*, *ps-en*, *en-ja*, *en-cs*, *en-yo*), we experimented with few-shot adaptation using half of the data released in the official development set. The official development set has 1K examples for each language pair (except *en-yo* for which there is no available data). To perform few-shot language adaptation we split the data into two halves: one for fine-tuning and another for validation.

Ensembling models. For our final submission for Direct Assessments we combine six multilingual systems using different hyperparameters by computing an weighted average of their outputs, where the weights for each language pair were tuned with Optuna (Akiba et al., 2019). The major difference between the ensembled models comes from the underlying encoder and whether or not they used word-level supervision. Three models of our final ensemble use word-level supervision while the other three use only sentence-level supervision. Regarding the encoder, three models use InfoXLM, two models use RemBERT and a single model uses XLM-R.

Our final submission for MQM predictions was an ensemble of eleven multilingual systems, which

that do not help in the task at hands

[4] https://huggingface.co/xlm-roberta-large

[5] https://huggingface.co/microsoft/infoxlm-large

[6] https://huggingface.co/google/rembert

[7] HTERs are available only for word-level subtasks.

combined the six systems used in the DA ensemble as well as five additional systems. For these additional systems, we made two major adjustments to the fine-tuning process. First, we filtered the DA data to the languages that were included in the MQM LPs, namely *ru-en*, *en-zh*, and *en-de*. Second, we incorporated the MQM data into the fine-tuning process, either as an additional fine-tuning step after fine-tuning on the language-filtered DA data, or by concatenating the DA and MQM data together. All additional systems used word-level supervision in addition to sentence-level and used InfoXLM as encoder.

3.1.2 Word-level quality prediction

Similarly, for the word-level QE tasks we experimented with both the multi-task setting and word-labels only ($\lambda_s = 0$ and $\lambda_w = 1$). Overall, we found that adding the sentence-level supervision was beneficial, especially for the languages pairs included in the test-set. Nonetheless, for some LPs, ignoring sentence-level supervision showed superior performance. Due to the mix of high-, mid- and low-resource languages in the data, the distribution of OK and BAD tags differs substantially between LPs leading to inconsistent performance in terms of MCC (see Table 5 in the appendix). To mitigate this, for the word-level subtask, we prepend a language prefix token to the beginning of the source and target segments during training and testing.

Pretraining on post-edit corpora. Extending the pretraining on Metrics data, we pretrain the word-level models on two corpora that include both word-level labels and sentence (HTER) scores, namely QT21 (Specia et al., 2017) and APEQuest (Ive et al., 2020). We compute the sentence-level score, using translation edit rate (TER) (Snover et al., 2006) between the target and the corresponding post-edited sentence.

Ensembling models. For word-level we followed a similar ensembling technique used for sentence-level, namely we combine multiple systems trained with different hyperparameters, encoders and pre-training setups. In the case of word-level predictions however, we need to resolve how to aggregate multiple predictions into OK/BAD tags. We use Optuna (Akiba et al., 2019) to choose how to weight and combine the models based on performance for each language pair on our internal test-set and we compare three different approaches:

1. A naive "best-only" approach: we identify the best model for each LP and use its predictions.

2. We ensemble the logits of each model: for each input segment we compute an ensembles of logits as $\sum_{i \in \mathcal{M}} w_i v_i$, where $\mathcal{M}$ is the set of models, w_i is the weight of each model and v_i the model logit vector. We use Optuna to find the optimal weight w_i for each model in each LP.

3. We ensemble the predicted tags of each model: for each input segment we compute an ensembles of tags as $\alpha \sum_{i \in \mathcal{M}} w_i c_i$, where c_i is the predicted class and α is the weight given for the BAD class. We use Optuna to find the optimal weights w_i for each model and the optimal BAD weight α for each LP.

In the final submission we combine five models for the post-edit originated LPs: a RemBERT based model, an InfoXLM based model pretrained on APEQuest and QT21, and three checkpoints that are based on InfoXLM but use different parameters for the BAD/OK weights and learning rate that were found via Optuna. For MQM we also combine five models, but this time instead of choosing three checkpoints based on optimising weights and learning rate, we use three different checkpoints with different training data mix on the relevant DA LPs, as this seemed to impact the performance on MQM word-level more than the weight ratios. Refer to §4 and Table 3 for more details.

3.2 Task 2: Explainable QE

The goal of the Explainable QE task is to identify machine translation errors without relying on word-level label information. In other words, it can be cast as an unsupervised word-level quality estimation problem, where explanations can be seen as highlights, representing the relevance of input words w.r.t. the model's prediction via continuous scores, aiming at identifying tokens that were not properly translated.

Several explainability methods can be used to extract highlights from a sentence-level model, such as post-hoc (Ribeiro et al., 2016; Arras et al., 2016) or inherently interpretable methods (Lei et al., 2016; Guerreiro and Martins, 2021). In our submission, we opted to use attention-based methods as they achieved the best results in the previous constrained track of the Explainable QE shared task (Fomicheva et al., 2021). Concretely, we take inspiration in the method developed by Treviso et al.

	Direct Assessment												
Encoder	**km-en**	**ps-en**	**en-ja**	**en-cs**	**en-mr**	**ru-en**	**ro-en**	**en-zh**	**en-de**	**et-en**	**si-en**	**ne-en**	**avg.**
Baseline (Zerva et al., 2021)													
XLM-R	0.615	0.601	0.295	0.535	0.419	0.703	0.828	0.513	0.500	0.806	0.565	0.793	0.598
Pretrained models													
InfoXLM	0.619	0.603	0.328	0.510	0.462	0.731	0.829	0.554	0.516	0.803	0.561	0.777	0.608
RemBERT	0.600	0.621	0.338	0.525	0.447	0.680	0.818	0.487	0.491	0.810	0.525	0.747	0.591
XLM-R	0.610	0.579	0.325	0.503	0.405	0.715	0.832	0.541	0.514	0.782	0.540	0.740	0.591
Sentence-level only													
XLM-R	0.628	0.591	0.350	0.531	0.551	0.761	0.859	0.577	0.568	0.800	0.565	0.796	0.631
InfoXLM	0.629	0.623	0.348	0.515	0.574	0.747	0.858	0.586	0.551	0.828	0.568	0.790	0.635
RemBERT	0.634	0.631	0.346	0.570	0.564	0.754	0.862	0.534	0.531	0.822	0.550	0.782	0.632
Few-shot Language Adaptation													
XLM-R	0.650	0.619	0.352	0.551	0.546	0.753	0.852	0.571	0.554	0.813	0.562	0.798	0.635
InfoXLM	0.641	0.650	0.367	0.549	0.549	0.751	0.855	0.591	0.565	0.824	0.563	0.803	0.642
RemBERT	0.625	0.641	0.367	0.568	0.563	0.756	0.857	0.540	0.527	0.824	0.568	0.796	0.636
Sentence + word-level training													
InfoXLM	0.617	0.586	0.344	0.532	0.572	0.761	0.865	0.586	0.579	0.829	0.576	0.804	0.637
RemBERT	0.634	0.628	0.356	0.564	0.571	0.762	0.860	0.541	0.553	0.826	0.564	0.799	0.638
Few-shot Language Adaptation													
InfoXLM	0.643	0.632	0.335	0.557	0.560	0.766	0.860	0.575	0.582	0.833	0.578	0.809	0.644
RemBERT	0.644	0.645	0.356	0.567	0.568	0.759	0.856	0.545	0.552	0.835	0.561	0.804	0.641
Final Ensemble													
Ensemble 6x	**0.664**	**0.669**	**0.380**	**0.591**	**0.593**	**0.782**	**0.871**	**0.597**	**0.593**	**0.845**	**0.588**	**0.820**	**0.666**

Table 1: Results for sentence-level QE in terms of Spearman correlation for DA.

(2021), which consists of scaling attention weights by the ℓ_2-norm of value vectors (Kobayashi et al., 2020) and finding the attention heads with the best performance on the dev set, and propose two new modifications:

- **Attention × GradNorm:** Following the findings of Chrysostomou and Aletras (2022), we decided to extract explanations that consider both attention and gradient information. More precisely, we scale the attention weights by the ℓ_2-norm of the gradient of value vectors:

$$\boldsymbol{A}_{\ell,h} \left\| \nabla \boldsymbol{V}_{\ell,h} \right\|_2 . \tag{6}$$

- **Head Mix:** We reformulate the scalar mix module (Eq. 2) to consider different weights for representations coming from different attention heads as follows:

$$\boldsymbol{H}_{\text{mix}} = \lambda \sum_{\ell=0}^{L} \beta_\ell \sum_{h=1}^{H} \gamma_{\ell,h} \boldsymbol{h}_{\ell,h}, \tag{7}$$

where the *layer* mix coefficients $\boldsymbol{\beta} \in \triangle^L$ are given by $\boldsymbol{\beta} = \pi(\boldsymbol{\phi})$, and the *head* mix coefficients $\boldsymbol{\gamma}_\ell \in \triangle^H$ are given by $\boldsymbol{\gamma}_\ell = \pi(\boldsymbol{\theta}_\ell)$. $\lambda \in \mathbb{R}$, $\boldsymbol{\phi} \in \mathbb{R}^L$ and $\boldsymbol{\theta} \in \mathbb{R}^{L \times H}$ are learnable parameters. We experimented both with dense (π as softmax) and sparse (π as sparsemax, Martins and Astudillo 2016) transformations. After training, the Head Mix coefficients can help to find attention heads with high validation performance, which is helpful for explaining zero-shot LPs.

Furthermore, since all of our sentence-level models use subword tokenization, to get explanations for an entire word we follow Treviso et al. (2021) and sum the scores of its word pieces.

Ensembling explanations. In our final submissions we average the explanation scores of different attention heads and layers to create a final explainer. We decided which heads and layers to aggregate together by looking at their performance on the dev set, selecting the top-5 with the highest explainability score.

3.3 Task 3: Critical Error Detection

Critical translations are defined as translations with strongly semantic deviations from the original source sentence, with the potential to lead to negative impacts in critical applications. The goal of this task is to predict sentence-level scores indicating whether a translation contains a critical error. Since the evaluation metrics automatically account for different binarization thresholds to separate good translations from bad ones, for this task we employed a single sentence-level InfoXLM

model from Task 1 that was trained on DA data. Moreover, we participated only in the *constrained setting*, meaning that we did not trained our systems specifically for this task. Therefore, our goal for this task was to validate whether our QE system from Task 1 was able to detect and differentiate translations with critical errors.

4 Experimental Results

As we have seen in Section 3, for our experiments we split the provided development sets into two equal size halves creating a new internal devset and an internal testset. The resulting sets contain ≈ 500 segments per language pair for both DA and MQM, word and sentence-level. As for baselines we used our submitted systems from previous shared tasks: for Task 1 we used the M1M-ADAPT (Zerva et al., 2021), and for Task 2 we used the Attn × Norm explainer (Treviso et al., 2021). The official results for Task 1 and Task 2 are shown in Table 6.

4.1 Quality Estimation

Sentence-level submissions were evaluated using the Spearman's rank correlation. Pearson's correlation, MAE, and RMSE were also used as secondary metrics, but here we report only Spearman correlation since it was the primary metric used to rank systems. Word-level submission were evaluated using MCC, F_1-OK, and F_1-BAD, but we report only MCC as it was considered the main metric. The submitted systems were independently evaluated on in-domain and zero-shot LPs for direct assessments and MQM.

Direct Assessments. Results for sentence-level DAs can be seen in Table 1. The results show that the training strategies employed in COMETKIWI, namely (i) pretraining models using Metrics data and (ii) incorporating references into training, lead to a correlation close to our best system from last year while disregarding the data from the MLQE-PE corpus. When fine-tuning on MLQE-PE data, we get overall improvements of $\sim$ 4%, and further fine-tuning on new LPs gives $\sim$ 1% overall improvement. Still, for the unseen LPs (*km-en, ps-en, en-ja, en-cs*), we got improvements between 2-3% with just 500 samples. Among the three backbone transformers, we noticed that InfoXLM is the one that leads to a higher Spearman correlation (+1.7% than XLM-R and RemBERT). Furthermore, including word-level supervision always maintains or improves the results, especially for InfoXLM. In contrast, RemBERT does not seem to benefit from this signal. We suspect that, for this task, the benefit of word-level supervision is not higher because the word-level information is coming from post-editions, which are conceptually different from DA annotations.

	MQM			
System (fine-tuned on)	**en-de**	**en-ru**	**zh-en**	**avg.**
Sentence-level only				
DA	0.529	0.534	0.215	0.426
DA + MQM	0.531	0.552	0.250	0.444
DA (3 LPs) + MQM	0.538	0.550	0.262	0.450
Sentence + word-level training				
DA	0.525	0.557	0.217	0.433
DA (3 LPs)	0.560	0.561	0.222	0.448
DA + MQM	0.540	0.568	0.262	0.457
DA (3 LPs) + MQM	0.553	**0.569**	0.268	0.463
DA (3 LPs) concat. MQM	**0.578**	0.547	**0.278**	**0.468**
Final Ensemble				
Ensemble 11x	0.568	0.556	0.223	0.449

Table 2: Results for sentence-level QE in terms of Spearman correlation for MQM.

MQM. Results for sentence-level MQM systems are shown in Table 2. The results show that the two main techniques used for adapting to MQM data, filtering DA data to the three MQM LPs and using MQM data for fine-tuning, improved Spearman correlations for all LPs over the pure DA baseline, for both sentence-level and multi-task systems. However, these techniques improved certain LPs more than others, so combining them together improved multilingual scores even further. Overall, we noticed that our results for MQM data have a high variance. To mitigate this, we concatenated the DA and MQM datasets together for a single fine-tuning, resulting in our best individual system on our internal test set. Due to these peculiarities in the MQM LPs, we decided to ensemble systems tuned on both DA and MQM data. Our final ensemble did not have as strong results as the individual systems on our internal test set, yet, it showed superior performance upon submission to codalab leader-board.

Word-level. For the word-level task we tuned models separately for the LPs that consisted of post-edit-derived word tags and the ones consisting of MQM-derived word tags; we report the Matthew's correlation coefficient (MCC) in Table 3. We experimented with multi-tasking by adding sentence-level supervision to the word-level task and found

Method	Post-edit						MQM			
	en-cs	en-ja	en-mr	km-en	ps-en	avg.	en-de	en-ru	zh-en	avg.
Baseline (Zerva et al., 2021)	0.272	0.154	0.326	0.427	0.348	0.305	0.176	0.177	0.065	0.139
InfoXLM as encoder										
Word-level	0.351	0.183	0.337	0.443	0.372	0.337	-	-	-	-
+ Sentence-level	0.410	0.230	0.368	0.436	0.369	0.363	0.294	0.256	0.399	0.316
+ LP prefix	0.371	0.202	0.391	0.512	0.411	0.377	0.259	0.440	0.211	0.303
+ APEQuest & QT21	0.414	0.245	0.372	0.494	0.389	0.383	0.246	0.382	0.209	0.279
+ tuned class-weights	0.389	0.218	0.421	0.499	0.391	0.384	0.285	0.404	0.172	0.287
DA (3LPs) + MQM	-	-	-	-	-	-	0.265	0.367	0.360	0.331
RemBERT as encoder										
Word + sentence-level	0.353	0.163	0.303	0.443	0.369	0.326	0.262	0.309	0.147	0.240
+ LP prefix	0.384	0.257	0.375	0.460	0.370	0.369	0.288	0.356	0.297	0.313
Ensemble "best-only"	0.414	0.245	0.421	0.512	0.411	0.401	0.300	0.382	0.360	0.347
Ensemble logits	**0.438**	**0.257**	**0.445**	**0.547**	**0.430**	**0.423**	**0.325**	0.443	0.296	0.355
Ensemble tags	0.432	0.253	0.429	0.537	0.423	0.415	0.313	**0.446**	**0.408**	**0.389**

Table 3: Results for word-level QE in terms of MCC for the post-edit and MQM LPs. Note that in each row, we use models trained separately on the MQM and non-MQM LPs.

that it boosts performance especially for the out-of-English translations. For the non-MQM LPs we used the HTER scores as sentence level targets as we found they lead to significantly higher correlations. We can also see that using the sentence-mix and the language prefix boosted the performance for all LPs, both in the MQM and post-edit originated LPs. Overall, the results show further improvements when we use the HTER scores of APEQuest and QT21 as additional pretraining data, but only for specific LPs. These findings merit further investigation, since the directionality of the LPs seems to have impacted our experiments. Finally, ensembling led to better results across all languages. Ensembling the logits led to better results for the post-edit originated LPs, while word-level ensembling helped more the MQM-originated LPs. Yet, in the submitted versions we found that the difference in performance between the three ensembling methods yielded similar results, with only 1-2% difference, while in the averaged multilingual versions these differences were even smaller, varying less than 0.1%.

4.2 Explainable QE

Since the explanations are given as continuous scores, they are evaluated against the ground-truth word-level labels in terms of the Area Under the Curve (AUC), Average Precision (AP), and Recall at Top-K (R@K) metrics only on the subset of translations that contain errors. Although R@K was considered the main metric for this task, we optimized internally for the average of all three metrics. The results are shown in Table 4.

Discussion. The results highlight several contrasts between explanations for DA and MQM data: (i) while RemBERT is useful as an encoder for DA data (outperforms InfoXLM in 3 out of 5 LPs), it is outperformed by InfoXLM for all MQM LPs; (ii) the Head Mix component improves performance for DA, but it does not impact significantly the scores for MQM; and (iii) the Sparse Head Mix generally outperforms the Soft Head Mix for DA, but the trend flips for MQM. On what comes to the explainability methods, the baseline method (Attn $\times$ Norm – scaling the attention weights by the ℓ_2-norm of value vectors), which obtained the best results in last year's Explainable QE shared task, is outperformed by our new method (Attn $\times$ GradNorm) for both DA and MQM data. Moreover, ensembling explanations from different heads brings further consistent improvements across the board for all LPs. For the zero-shot setting (*en-yo*), we build an ensemble of explanations by using the heads that were more common among the ensembles for all other LPs. This approach might be worth researching further, since it is possible to study the Head Mix coefficients to select good-performing attention heads.

5 Official Results

We present the official results of our submissions alongside the results from other competitors in Section B for all three tasks. For sentence-level, our submissions achieved the best results for 6/9 LPs. For word-level, we obtained the best results for 5/9 LPs. For the explainable QE track, we obtained the

Method	Direct Assessment						MQM			
	en-cs	en-ja	en-mr	km-en	ps-en	avg.	en-de	en-ru	zh-en	avg.
Baseline (Treviso et al., 2021)†	0.602	0.510	0.428	0.636	0.633	0.562	0.529	0.552	0.450	0.510
InfoXLM as encoder										
Attn × GradNorm	0.602	0.495	0.417	0.653	0.648	0.563	0.539	0.559	0.474	0.524
+ Soft Head Mix	0.600	0.495	0.426	0.656	0.653	0.566	0.532	0.563	0.467	0.521
+ Sparse Head Mix	0.604	0.503	0.421	0.658	0.660	0.569	0.541	0.551	0.454	0.515
Ensemble	0.641	0.521	0.440	0.669	0.667	0.588	**0.580**	**0.603**	**0.505**	**0.563**
+ Soft Head Mix	0.621	0.501	0.432	0.681	0.661	0.579	0.567	0.588	0.504	0.553
+ Sparse Head Mix	**0.645**	0.519	**0.450**	0.688	0.675	0.595	0.574	0.582	0.484	0.547
RemBERT as encoder										
Attn × GradNorm	0.596	0.511	0.427	0.675	0.676	0.577	0.474	0.532	0.448	0.485
+ Soft Head Mix	0.588	0.538	0.430	0.658	0.654	0.574	0.473	0.529	0.455	0.486
+ Sparse Head Mix	0.588	0.534	0.428	0.658	0.652	0.572	0.470	0.530	0.443	0.481
Ensemble	0.609	0.551	0.443	**0.702**	0.685	0.598	0.516	0.554	0.506	0.525
+ Soft Head Mix	0.613	**0.561**	0.448	0.699	0.692	0.603	0.521	0.558	0.498	0.526
+ Sparse Head Mix	0.620	0.557	0.447	**0.702**	**0.691**	**0.604**	0.511	0.551	0.503	0.522

Table 4: Explainable QE task results in terms of the average of AUC, AP and R@K. †We used InfoXLM to compute the results for the baseline.

best results for all but two LPs (*km-en* and ps-en). Although the critical error detection task had no other competitor for the *constrained setting*, our submission vastly surpassed the organizers' baseline. We also obtained the best results for the multilingual settings (including and excluding *en-yo*) for all tasks. Finally, when averaging the results for all LPs, our submissions place on top for all tasks.

6 Conclusions and Future Work

We presented the joint contribution of IST and Unbabel to the WMT 2022 QE shared task. We found that incorporating references during pretraining improves performance across several LPs on downstream tasks, and that jointly training with sentence and word-level objectives yields a further boost. For Task 1, our final submissions were ensembles of models finetuned with different pretrained language models as encoders, boosting the results when compared to the previous year submission. For Task 2, we take inspiration on the literature of explainability and propose to use gradient information in tandem with attention weights, and to further refine the impact of attention heads towards the prediction via the Head Mix component. Besides leading to better explainability performance for some LPs, this strategy is potentially useful to identify good attention heads at inference time for zero-shot LPs, and deserves more investigation. Overall, our submissions achieved the best results for all tasks (including Task 3) for almost all LPs by a considerable margin.

One of the challenges of leveraging big ensembles is the burdensome weight of parameters and inference time. For future work we will extend our recent work, COMETINHO (Rei et al., 2022) and explore how to effectively distill large ensembles into small and more practical QE systems.

Acknowledgements

This work was supported by the P2020 programs (MAIA, contract 045909), by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the European Research Council (ERC StG DeepSPIN, 758969), and by the Fundação para a Ciência e Tecnologia (contract UIDB/50008/2020).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan 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, Online. Association for Computational Linguistics.

George Chrysostomou and Nikolaos Aletras. 2022. An empirical study on explanations in out-of-domain settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6920–6938, Dublin, Ireland. Association for Computational Linguistics.

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*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. 2021. The Eval4NLP shared task on explainable quality estimation: Overview and results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 165–178, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2022. MLQE-PE: A Multilingual Quality Estimation and Post-Editing Dataset. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4963–4974, Marseille, France. European Language Resources Association.

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

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, 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.

Nuno M. Guerreiro and André F. T. Martins. 2021. SPECTRA: Sparse structured text rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6534–6550, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Julia Ive, Lucia Specia, Sara Szoc, Tom Vanallemeersch, Joachim Van den Bogaert, Eduardo Farah, Christine Maroti, Artur Ventura, and Maxim Khalilov. 2020. A post-editing dataset in the legal domain: Do we underestimate neural machine translation quality? In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3692–3697, Marseille, France. European Language Resources Association.

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.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium. European Association for Machine Translation.

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

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proc. ACM SIGKDD*, pages 1135–1144. ACM.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadin, Matteo Negri, and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 55–71, Nagoya Japan.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Marcos Treviso, Nuno M. Guerreiro, Ricardo Rei, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the explainable quality estimation shared task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 133–145, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5998–6008. Curran Associates, Inc.

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

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 Shared Task on Quality Estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

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

A Data Information

The data used for finetuning our QE systems is shown in Table 5. For DA data, we split the original development set to generate a new dev/test split, therefore the reported numbers in the table correspond to this "internal" dev split.

LP	Samples	Source Tokens	Target Tokens	Target OK / BAD
TRAIN				
en-de	9000	147870	153656	0.84 / 0.16
en-mr	26000	690516	561371	0.90 / 0.10
en-zh	9000	148657	163308	0.65 / 0.35
et-en	9000	126877	185491	0.75 / 0.25
ne-en	9000	135205	181707	0.41 / 0.59
ro-en	9000	154538	167471	0.71 / 0.29
ru-en	9000	104423	132006	0.85 / 0.15
si-en	9000	141283	166914	0.42 / 0.58
en-de†	54681	1571090	1926444	0.90 / 0.10
en-ru†	15628	312185	354871	0.95 / 0.05
zh-en†	75327	134165	2789907	0.87 / 0.13
DEV				
en-de	500	8262	8555	0.84 / 0.16
en-mr	500	13803	11216	0.91 / 0.09
en-zh	500	8422	9302	0.75 / 0.25
et-en	500	7081	10257	0.73 / 0.27
ne-en	500	7542	10247	0.38 / 0.62
ro-en	500	8550	9202	0.78 / 0.22
ru-en	500	5984	7511	0.84 / 0.16
si-en	500	7866	9415	0.41 / 0.59
en-cs	500	10302	9302	0.75 / 0.25
en-ja	500	10354	13287	0.73 / 0.27
km-en	495	9015	8843	0.45 / 0.55
ps-en	500	13463	12160	0.51 / 0.49
en-de†	503	10535	12454	0.96 / 0.04
en-ru†	503	10767	11911	0.91 / 0.09
zh-en†	509	980	19192	0.98 / 0.02

Table 5: DA and MQM (†) data for all LPs.

B Official Results

Critical Error Detection. Submissions for this task were evaluated in terms of ranking using R@K

and MCC as metrics. In Table 7, we report only MCC scores as it was the main metric for this task.

QE and Explainable QE. Table 6 shows the official results for sentence-level QE (top), word-level QE (middle), and explainable QE (bottom).

	Direct Assessment								MQM		
Team	**en-cs**	**en-ja**	**en-mr**	**en-yo**	**km-en**	**ps-en**	**all**	**all/yo**	**en-ru**	**en-de**	**zh-en**
Sentence-level QE											
Baseline	0.560	0.272	0.436	0.002	0.579	0.641	0.415	0.497	0.333	0.455	0.164
Alibaba	-	-	-	-	-	-	-	-	0.505	0.550	0.347
NJUQE	-	-	0.585	-	-	-	-	-	0.474	**0.635**	0.296
Welocalize	0.563	0.276	0.444	-	0.623	-	0.448	0.506	-	-	-
joanne.wjy	0.635	0.348	0.597	-	0.657	0.697	-	0.587	-	-	-
HW-TSC	0.626	0.341	0.567	-	0.509	0.661	-	-	0.433	0.494	**0.369**
Papago	0.636	0.327	**0.604**	0.121	0.653	0.671	0.502	0.571	0.496	0.582	0.325
IST-Unbabel	**0.655**	**0.385**	0.592	**0.409**	**0.669**	**0.722**	**0.572**	**0.605**	**0.519**	0.561	0.348
Word-level QE											
Baseline	0.325	0.175	0.306	0.000	0.402	0.359	0.235	0.257	0.203	0.182	0.104
NJUQE	-	-	0.412	-	0.421	-	-	-	0.390	**0.352**	0.308
HW-TSC	0.424	**0.258**	0.351	-	0.353	0.358	-	0.218	0.343	0.274	0.246
Papago	0.396	0.257	**0.418**	0.028	**0.429**	0.374	0.317	0.343	0.421	0.319	0.351
IST-Unbabel	**0.436**	0.238	0.392	**0.131**	0.425	**0.424**	**0.341**	**0.361**	**0.427**	0.303	**0.360**
Explainable QE											
Baseline	0.417	0.367	0.194	0.111	0.580	0.615	0.381	0.435	0.148	0.074	0.048
f.azadi	-	-	-	-	0.622	0.668	-	-	-	-	-
HW-TSC	0.536	0.462	0.280	-	**0.686**	**0.715**	-	0.535	0.313	0.252	0.220
IST-Unbabel	**0.561**	**0.466**	**0.317**	**0.234**	0.665	0.672	**0.486**	**0.536**	**0.390**	**0.365**	**0.379**

Table 6: Official results for sentence-level QE (top) in terms of Spearman's correlation, word-level QE (middle) in terms of MCC, and explainable QE (bottom) in terms of R@K. We estimated the numbers of *en-yo* for teams that did not submit to *en-yo* directly but still submitted to all other LPs and to the *multilingual* (all) category.

Method	**en-de**	**pt-en**
Baseline	0.0738	-0.0013
InfoXLM finetuned on DAs	0.5641	0.7209

Table 7: Official results for the Critical Error Detection task in terms of MCC.