IST-Unbabel 2021 Submission for the Explainable Quality Estimation Shared Task

Marcos V. Treviso[1,2], **Nuno M. Guerreiro**[1,2], **Ricardo Rei**[2,3,4], **André F. T. Martins**[1,2,3]
[1]Instituto de Telecomunicações, Lisbon, Portugal
[2]Instituto Superior Técnico, Lisbon, Portugal
[3]Unbabel, Lisbon, Portugal
[4]INESC-ID, Lisbon, Portugal

Abstract

We present the joint contribution of Instituto Superior Técnico (IST) and Unbabel to the Explainable Quality Estimation (QE) shared task, where systems were submitted to two tracks: constrained (without word-level supervision) and unconstrained (with word-level supervision). For the constrained track, we experimented with several explainability methods to extract the relevance of input tokens from sentence-level QE models built on top of multilingual pre-trained transformers. Among the different tested methods, composing explanations in the form of attention weights scaled by the norm of value vectors yielded the best results. When word-level labels are used during training, our best results were obtained by using word-level predicted probabilities. We further improve the performance of our methods on the two tracks by ensembling explanation scores extracted from models trained with different pre-trained transformers, achieving strong results for in-domain and zero-shot language pairs.

1 Introduction

Quality estimation (QE) aims at assessing the quality of a translation system without relying on reference translations (Blatz et al., 2004; Specia et al., 2018). This paper describes the joint contribution of Instituto Superior Técnico (IST) and Unbabel to the Explainable Quality Estimation shared task (Fomicheva et al., 2021a). The goal of the shared task is to identify translation errors without direct word-level supervision (constrained track) or with access to word-level labels (unconstrained track).

Recent advances in QE have led to consistent improvements at predicting quality assessments such as *Direct Assessments* (DAs, Graham et al. 2013). Traditional QE systems had to predict *Human Translation Error Rate* (HTER, Snover et al. 2006), yet with the advent of neural machine translation, we observed a shift from fluency into adequacy errors (Martindale and Carpuat, 2018). For that reason, DAs started getting used as the ground-truth score for assessing the quality of translations (Specia et al., 2020). However, with DAs we lose the ability to generate word-level supervision, impacting the interpretability of sentence-level predictions in terms of lower granularity elements such as word-level translation errors.

At the same time, state-of-the-art QE systems such as OpenKiwi (Kepler et al., 2019b) and TransQuest (Ranasinghe et al., 2020b) build on top of multilingual pre-trained models such as BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), which are largely responsible for the performance boost we have observed in the last two editions of the WMT QE shared task (Fonseca et al., 2019; Specia et al., 2020). Due to the usage of such overparametrized black-box models, this performance boost also comes at the cost of efficiency and interpretability.

Research in explainable NLP uncovered several strategies to interpret models' decisions, either in a post-hoc manner by querying a trained model for extracting perturbation or gradient measures (Ribeiro et al., 2016; Arras et al., 2016), or by building models that are inherently interpretable (Lei et al., 2016; Chang et al., 2020). Recent works have also put transformers under the lens of explainability, aiming at unraveling interpretable patterns that clarify how decisions emerge from attention heads and across hidden states at each layer (De Cao et al., 2020; Abnar and Zuidema, 2020; Voita et al., 2021).

In this shared task, we experiment with several of these methods to extract the relevance of input tokens from sentence-level QE models built on top of multilingual pre-trained transformers[1]. For the constrained track, where models are unaware of word-level supervision, our best results were de-

[1]Our code can be found at: https://github.com/deep-spin/explainable_qe_shared_task/.

rived from attention-based explanations. When we used word-level labels during training, the best results were obtained by using word-level predicted probabilities. Furthermore, we were able to push the performance further by ensembling explanations for both tracks.

2 Background

Quality Estimation. QE systems are usually designed according to the granularity in which predictions are made: word, sentence, or document-level. The goal of word-level QE is to assign quality labels (OK or BAD) to each *machine-translated word*, indicating whether that word is a translation error or not. Additionally, current systems also classify *source words* to denote words in the original sentence that have been mistranslated or omitted in the target. On the other hand, sentence-level QE aims at predicting the quality of the whole translated sentence, either in terms of how many edit operations are required to fix it (HTER) or in terms of human judgments (DA). Similarly, document-level QE systems predict a single outcome (a real score or a ranking index) for an entire document.

Transformers. The multi-head attention mechanism is the bedrock on which transformers are built. They are responsible for contextualizing the information within and across input sentences dynamically (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:

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

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

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

where $\boldsymbol{W}_h^Q$, $\boldsymbol{W}_h^K$, $\boldsymbol{W}_h^V$ are learned linear transformations. The output of the multi-head attention module is the concatenation of all k heads followed by a learnable linear transformation $\boldsymbol{W}^O$:

$$\text{mh-att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{concat}(\text{head}_1, ..., \text{head}_k)\boldsymbol{W}^O.$$

This way, heads have the capability of learning specialized phenomena. Transformers with only encoder-blocks, such as BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), have only the encoder self-attention, and thus $m = n$.

Explainability in NLP. There is a large body of work on the analysis and interpretation of models in NLP. Some of these models are built on top of attention mechanisms, which automatically learn a weighted representation of input features. Attention weights provide plausible, but not always faithful, explanations (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). In contrast, rationalizers with hard attention are arguably more faithful but require stochastic networks (Lei et al., 2016; Bastings et al., 2019), with recent works avoiding stochasticity via sparse deterministic selections (Guerreiro and Martins, 2021). Other approaches seek local explanations by considering gradient measures (Arras et al., 2016; Bastings and Filippova, 2020), or by perturbing the input and querying the classifier in a post-hoc manner (Ribeiro et al., 2016; Kim et al., 2020). Since transformers are composed of several layers and attention heads, many works analyze and improve the multi-head attention mechanism directly to produce better explanations (Kobayashi et al., 2020; Hao et al., 2021). More elaborated methods consider the entire flow of information coming from attention weights, hidden states, or gradients to interpret the model's decision (De Cao et al., 2020; Abnar and Zuidema, 2020; Voita et al., 2021).

3 Constrained Track

The goal of the constrained track is to identify machine translation errors without explicit word-level annotation. More precisely, it aims at performing word-level quality estimation by casting the task as a prediction explainability problem. In the context of QE, explanations can be seen as highlights, representing the relevance of input words w.r.t. the model's prediction via continuous scores. We next describe the datasets, models, and explainability methods that we used for this track.

3.1 Datasets

Seeking to improve the performance of our models on the zero-shot language pairs (LPs), we used all language pairs from the MLQE-PE dataset (Fomicheva et al., 2020) to train our models

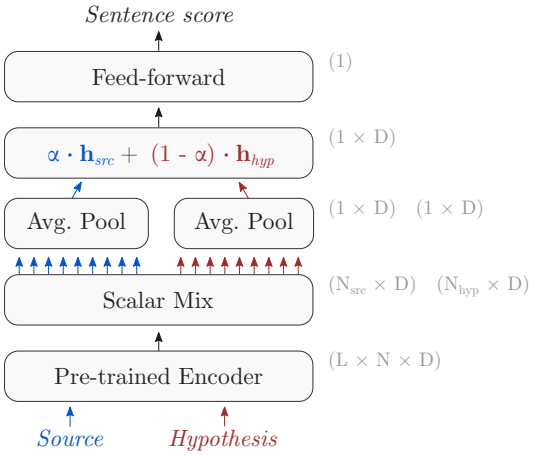

Figure 1: General architecture of our models for the constrained track. L represents the number of layers. N_{src} and N_{hyp} represent the number of words in the source and hypothesis sentences, respectively. $N = N_{\text{src}} + N_{\text{hyp}}$ is the number of words after concatenating the two sentences. D is the size of hidden vectors.

for both tracks. For RO-EN and ET-EN, we evaluated our models on the validation set of these LPs. For the two zero-shot LPs, DE-ZH and RU-DE, we used the 20 sentences made available by the shared task and the validation sets of EN-ZH and EN-DE to improve the robustness of the evaluation of explanations w.r.t. the target language. We used word-level labels to train word-level models for the unconstrained track only. For sentence-level models, we supervise our models using DA scores.

3.2 Sentence-level Models

Since QE is a fundamental tool in many MT pipelines, we focus our efforts on designing and explaining QE systems with high sentence-level performance. Therefore, we opted to follow the recent trend in this area (Kepler et al., 2019b; Ranasinghe et al., 2020a) and employed two pre-trained multilingual language models as the feature extractors for our models: XLM-RoBERTa and RemBERT.

The overall architecture of our models is shown in Figure 1. The tokenized source $\boldsymbol{s} = \langle s_1, ..., s_N \rangle$ and hypothesis $\boldsymbol{t} = \langle t_1, ..., t_M \rangle$ sentences are concatenated and passed as input to the encoder, which produces hidden state vectors $\boldsymbol{H}_0, ..., \boldsymbol{H}_L$ for each layer $0 \leq \ell \leq L$, where $\boldsymbol{H}_i \in \mathbb{R}^{(N+M)\times d}$. 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 $\boldsymbol{H}_{L+1}$. We split $\boldsymbol{H}_{L+1}$ into source $\boldsymbol{H}_{src} \in \mathbb{R}^{N\times d}$ and hypothesis hidden states $\boldsymbol{H}_{hyp} \in \mathbb{R}^{M\times d}$, which are independently passed to an average pooling layer to get their sentence representations $\boldsymbol{h}_{src}$ and $\boldsymbol{h}_{hyp}$. We merge both representations via a convex combination with $\alpha = 0.5$ to encourage the model to use both source and hypothesis contexts. Finally, we pass the combined vector to a 2-layered feed-forward module in order to get a sentence score prediction $\hat{y} \in \mathbb{R}$. Moreover, attention matrices $\boldsymbol{A}_1, ..., \boldsymbol{A}_L$ are also recovered as a by-product of the forward propagation, where $\boldsymbol{A}_i \in \mathbb{R}^{(N+M)\times(N+M)}$. The hyperparameters used for training can be found in §B.

XLM-RoBERTa as encoder. We set a XLM-RoBERTa Large (XLM-R, Conneau et al. 2020) as the encoder layer.[2] XLM-R is a cross-lingual transformer pre-trained on massive amounts of multilingual data. It consists of 24 encoder blocks with 16 attention heads each. Following (Zerva et al., 2021) we train our complete model on DAs by using adapters for the XLM-R encoder (Houlsby et al., 2019; Pfeiffer et al., 2020) to adapt it to the domain specific data of the QE task with minimal training effort.

XLM-RoBERTa for zero-shot LPs. To improve the robustness of XLM-R on out-of-domain data, we used an XLM-RoBERTa Large model that was trained with DA's from the metrics shared task.[3] Next, we set it as the encoder layer, and adapted it for predicting DAs from the MLQE corpus as in (Zerva et al., 2021). Altogether, the data from the Metrics shared task encompasses 30 language pairs from the news domain—yet, the zero-shot LPs are not included in this set. The hyperparameters and the training regime of this model are the same as the previously described XLM-R. We denote this model as XLM-R-M from here on.

RemBERT as encoder. We replace the XLM-R by a RemBERT model as the encoder layer (Chung et al., 2021).[4] Multilingual BERT (Devlin et al., 2019) has been shown to provide complementary performance to XLM-based models for sentence-level and word-level QE (Kepler et al., 2019a). We opted to use RemBERT since it can be seen as a larger multilingual BERT with decoupled input and output embeddings, which helps to accelerate

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

[3] https://huggingface.co/Unbabel/xlm-roberta-wmt-metrics-da

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

ENCODER	RO-EN	ET-EN	DE-ZH	RU-DE
OpenKiwi	0.820	0.757	0.395	0.176
XLM-R	0.878	0.756	0.521	**0.563**
XLM-R-M	0.877	**0.780**	**0.797**	0.352
RemBERT	**0.883**	0.762	-0.002	0.505

Table 1: Pearson correlation of our sentence-level QE systems by varying the model used as the encoder layer.

training. It consists of 32 encoder blocks with 18 attention heads each. Rather than aggregating layers with the scalar mix layer, we perform average pooling over the hidden states of the last layer of RemBERT. For training, we simply finetune the whole model with small learning rates.

Results. Table 1 summarizes the performance of our sentence-level models on the validation set in terms of Pearson correlation for each language pair evaluated in the shared task. For completeness, we show results for the 20 sentences made available by the shared task for DE-ZH and RU-DE. We also include OpenKiwi with a XLM-R Large as the encoder for comparison. We note that results for DE-ZH and RU-DE are noisy due to the small amount of validation data available for these LPs.

3.3 Explainability Methods

Several explainability methods can be used to extract highlights from a trained model in a post-hoc fashion. It is also possible to design a model that is explainable by construction, such as rationalizers (Lei et al., 2016; Bastings et al., 2019). We investigate rationalizers, attention, gradient, and perturbation-based methods for this shared task.

Attention-based methods. Since the backbone of our models consists of pre-trained multilingual transformers, we studied their main component—the multi-head attention mechanism—expecting to find interpretability patterns that assign higher scores to words associated with translation errors. We extracted the following explanations from the multi-head attention mechanism:

- **Attention weights:** average the attention matrix $\mathbf{A}$ row-wise for all heads in all layers, amounting to a total of $24 \times 16 = 384$ and $32 \times 18 = 576$ explanation vectors $\boldsymbol{a} \in \mathbb{R}^{N+M}$ for XLM-R and RemBERT-based models, respectively.

- **Cross-attention weights:** by manual inspection of attention weights, we noticed that some attention heads learn plausible connections from source-to-hypothesis and hypothesis-to-source. Therefore, instead of computing a row-wise average of the entire attention matrix, we average only cross-alignment rows.[5]

- **Attention $\times$ Norm:** following the findings of Kobayashi et al. (2020), we scale attention weights by the norm of value vectors $\left\|\boldsymbol{V}\boldsymbol{W}_h^V\right\|_2$.

Gradient-based methods. Explanations extracted by storing gradients computed during the backward propagation is a standard tool used to interpret NLP models. For this shared task, we investigate the following gradient-based methods:[6]

- **Gradient $\times$ Hidden States:** we compute gradients w.r.t. the hidden states of each layer, and multiply the resultant vectors by the hidden state vectors themselves: $\nabla_{\boldsymbol{H}_i} \times \boldsymbol{H}_i \in \mathbb{R}^{N+M}$, for $0 \leq i \leq L+1$.

- **Gradient $\times$ Attention:** the same as before, but we use the output of the multi-head attention module instead of the hidden states.

- **Integrated Gradients**: we extract integrated gradient explanations w.r.t. the hidden states of each layer. We use a zero-vector as the baseline. We map gradients to explainability scores by normalizing them by their L2 norm and summing the hidden dimensions: $\mathbf{1}^\top \nabla_{\boldsymbol{H}_i} / \left\|\nabla_{\boldsymbol{H}_i}\right\|_2$.

Perturbation-based methods. As baselines, we also extracted explanations using **LIME** (Ribeiro et al., 2016) and a **leave-one-out** strategy, where we replace the "erased" token by the `<mask>` token, which is used for the masked-language model training of XLM-R and RemBERT.

Rationalizers. We append a differentiable binary mask layer (Bastings et al., 2019) on top of the XLM-R model in order to select which tokens are passed on for an estimator for the prediction of a sentence-level score. For each instance, we take the model representations from the scalar-mix layer and pass it to an encoder module, in which we sample a binary mask $\boldsymbol{z} \in [0,1]^{N+M}$ from a relaxed Bernoulli distribution (Maddison et al., 2017; Jang et al., 2017), and pass $\boldsymbol{z} \odot [\boldsymbol{s}; \boldsymbol{t}]$ to an estimator module, which re-embeds the masked input and

[5] Note that we can get cross-attentions from XLM-R and RemBERT by selecting only the words of the source that attend to the hypothesis and vice-versa.

[6] Our implementation is based on Captum: https://captum.ai/

	RO-EN		ET-EN	
ENCODER	Source	Target	Source	Target
OpenKiwi	0.581	0.620	0.488	0.554
XLM-R	0.610	0.644	**0.503**	**0.559**
XLM-R-M	**0.636**	**0.667**	0.464	0.530
RemBERT	0.624	0.659	0.474	0.555
	DE-ZH		RU-DE	
ENCODER	Source	Target	Source	Target
OpenKiwi	0.**271**	0.184	0.243	0.029
XLM-R	0.230	0.312	0.273	0.061
XLM-R-M	0.262	**0.336**	**0.343**	0.179
RemBERT	0.173	0.211	0.247	**0.201**

Table 2: Source and target MCC results of our word-level QE systems by varying the model used as the encoder layer. The values of λ for each model are: $10^3, 10^4, 10^4, 10^4$.

pass it to a linear output layer. Therefore, good explanations $\boldsymbol{z}$ will aid the estimator in producing good sentence-level scores. In training time, the parameters of the encoder and the estimator are jointly trained. In test time, we do not sample the binary masks. Instead, we use the relaxed Bernoulli distribution probabilities as explanations.

4 Unconstrained Track

In this track, we opted to use word-level annotation by incorporating a word-level loss to our previous models. To do this, we apply a map from word pieces to tokens after the scalar mix layer and pass the hidden vectors of each token through a feed-forward layer with a sigmoid activation to predict scores $\hat{y}_i \in [0, 1]$. We weight the word-level loss by λ and sum it with the sentence-level loss. As baseline, we train a XLM-R Large model using OpenKiwi with the default hyperparameters. For all word-level models, we train with $\lambda \in \{10^3, 10^4, 10^5\}$ and save the checkpoint with the best performance on the validation set.

Results. Table 2 shows the results of our word-level models on the validation set in terms of Matthews correlation coefficient (MCC) for each LP evaluated in the shared task. For completeness, we include the results for the 20 available sentences for DE-ZH and RU-DE.

5 Experimental Results

Although we can regard the extracted explanations as errors in the translation output, an analogous evaluation of word-level QE is not straightforward since the standard metrics require binary labels rather than continuous scores. Therefore, the explanations 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.

Furthermore, since all of our models use subword tokenization, to get explanations for an entire word, we tried aggregating the scores of its word pieces by taking the sum, mean, or max, and we found that taking the sum performs better overall.

5.1 Constrained Track

Attention heads are better alone. We found that some attention heads (mostly at upper layers) learned to focus on words associated with BAD tags, achieving great performance in terms of AUC and AP on the validation set. We show in Figure 2 the target AUC of different attention heads per layer as a heatmap for RO-EN, with darker colors indicating higher results.[7] We can see that attention heads in layers 18 and 19 perform better than other layers in general, and that some attention heads solely outperform the average of all attention heads for all respective layers. For example, the attention head 3 at layer 18 achieves an AUC score of 0.79, while the average of all attention heads from layer 18 gets an AUC score of 0.74 (5 points difference). The findings are similar for source AUC, with the exception that attention heads at lower layers also seem to achieve comparable, yet not better, results. This behavior was also noted by Fomicheva et al. (2021b), with the difference that we analyzed attention heads independently rather than averaging them at each layer. Kobayashi et al. (2020) also arrive at similar findings but in terms of alignment error rate in a neural machine translation context.

Attention $\times$ Norm outperforms other explainers. By scaling attention probabilities by the L2 norm of value vectors, we improved the performance further. All of our best results consist of attention-based explainers, with the majority being the explanations that consider the norm of value vectors. We show the results of our best explainers on the validation set of RO-EN in Table 3 using XLM-R as encoder.[8] When using XLM-R-M or RemBERT as encoder the results are similar, except that the best explainer comes from different attention heads at different upper layers.

[7] We got similar findings for ET-EN.

[8] Results for ET-EN follow the same trend (see §C).

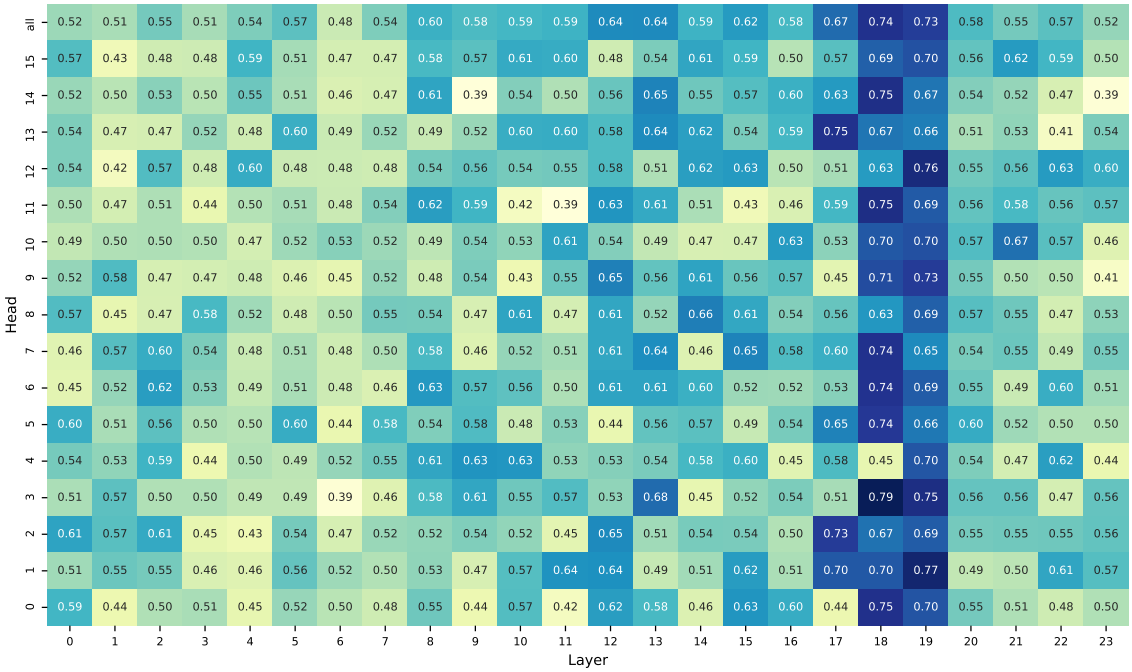

Figure 2: Target AUC of different attention heads at each layer of our XLM-R model for RO-EN. The last tick on the y-axis represents the average of all attention heads.

	Source			Target		
EXPLAINER	AUC	AP	R@K	AUC	AP	R@K
Attention	0.7445	0.6353	0.5164	0.7894	0.7189	0.6054
Cross-attention	0.7514	0.6345	0.5170	0.8066	0.7378	0.6293
Attention × Norm	**0.7851**	**0.6875**	**0.5701**	**0.8136**	**0.7432**	**0.6342**
Gradient × Hidden States	0.6949	0.5629	0.4399	0.6780	0.5388	0.4044
Gradient × Attention	0.7104	0.5942	0.4913	0.7618	0.6747	0.5628
Integrated Gradients	0.6539	0.5251	0.4059	0.6560	0.5148	0.3853
LIME	0.6470	0.5160	0.3922	0.5892	0.4576	0.3300
Leave-one-out	0.6970	0.5673	0.4409	0.5921	0.4752	0.3567
Relaxed-Bernoulli Rationalizer	0.4803	0.3638	0.2483	0.5434	0.4043	0.2914

Table 3: Constrained track results for different explainability methods on the validation set of RO-EN using XLM-R as encoder.

Overall, we observed that attention methods outperform gradient and perturbation methods by a considerable margin, and gradients w.r.t. attention outputs yield better results than gradients w.r.t. hidden states, indicating that the information stored in attention heads is valuable. In Figure 3 we show the attention map of two attention heads that perform well in terms of source AUC and target AUC on the validation set of RO-EN. We noted qualitatively that attention-heads that perform well on source AUC usually focus on cross-sentence tokens,[9] whereas attention-heads that have good results in terms of target AUC usually focus on hypothesis tokens. Lastly, our strategy of appending a bottleneck layer acting as rationalizer did not work well, achieving worse results than perturbation-based methods.

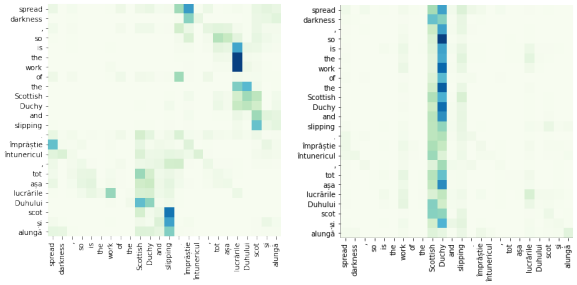

Figure 3: Example of two attention maps from particular heads that perform well on source AUC (left) and target AUC (right) for RO-EN.

[9]Cross-sentence tokens are hypoehsis tokens attended by source tokens and also source tokens attended by hypoehsis tokens.

LP	Encoder	Source (constrained)			Target (constrained)			Source (unconstrained)			Target (unconstrained)		
		AUC	AP	R@K	AUC	AP	R@K	AUC	AP	R@K	AUC	AP	R@K
RO-EN	OpenKiwi	-	-	-	-	-	-	0.907	0.811	0.704	0.921	0.826	0.718
	XLM-R	0.785	0.687	0.570	0.814	0.743	0.634	0.914	0.825	0.722	0.928	0.851	0.764
	XLM-R-M	0.753	0.661	0.548	0.769	0.693	0.593	0.913	0.826	0.724	0.926	0.851	0.761
	RemBERT	0.784	0.699	0.590	0.790	0.686	0.572	0.918	0.831	0.731	0.934	0.862	0.769
	Ensemble	**0.807**	**0.720**	**0.607**	**0.842**	**0.772**	**0.662**	**0.927**	**0.844**	**0.744**	**0.942**	**0.874**	**0.786**
ET-EN	OpenKiwi	-	-	-	-	-	-	0.848	0.749	0.635	0.873	0.798	0.692
	XLM-R	0.733	0.618	0.486	0.740	0.648	0.530	0.858	0.768	0.656	0.881	0.814	0.711
	XLM-R-M	0.623	0.504	0.367	0.712	0.625	0.513	0.854	0.751	0.630	0.875	0.804	0.704
	RemBERT	**0.750**	**0.638**	**0.523**	0.708	0.595	0.476	0.851	0.747	0.631	0.881	0.806	0.703
	Ensemble	0.744	0.637	0.509	**0.764**	**0.680**	**0.569**	**0.870**	**0.778**	**0.668**	**0.896**	**0.832**	**0.735**
DE-ZH	OpenKiwi	-	-	-	-	-	-	0.721	0.616	0.545	0.648	0.483	0.356
	XLM-R	0.720	0.465	0.288	0.683	0.542	0.406	0.674	0.486	0.298	0.650	0.511	0.352
	XLM-R-M	0.773	**0.609**	**0.454**	0.697	0.545	0.427	0.711	0.574	0.463	**0.712**	**0.595**	**0.468**
	RemBERT	0.762	0.579	0.405	0.692	0.470	0.358	0.619	0.443	0.341	0.585	0.445	0.354
	Ensemble	**0.792**	0.581	0.440	**0.711**	**0.575**	**0.477**	**0.745**	**0.635**	**0.548**	0.705	0.575	0.418
RU-DE	OpenKiwi	-	-	-	-	-	-	0.727	0.620	0.559	0.620	0.409	0.359
	XLM-R	0.719	0.400	0.316	0.822	0.500	0.335	0.729	0.604	0.485	0.623	0.369	0.282
	XLM-R-M	0.743	0.529	0.425	0.838	0.532	0.369	0.740	0.645	0.545	0.640	0.470	**0.447**
	RemBERT	0.776	**0.646**	**0.550**	0.826	0.537	0.418	**0.802**	0.712	0.607	**0.721**	0.504	0.393
	Ensemble	**0.804**	0.604	0.459	**0.855**	**0.628**	**0.514**	0.799	**0.716**	**0.616**	0.719	**0.521**	0.439

Table 4: Constrained (left) and unconstrained (right) track results on the validation set for all LPs using the Attention × Norm explainer.

Results for all LPs. We show the results on the validation set for all LPs in Table 4 (left) with the best Attention × Norm explanations for each tested encoder. We also report results of ensembled explanations, which are obtained by simply averaging selected Attention × Norm explanations from models with different encoders. When comparing single encoders for in-domain LPs, we see that explanations from our XLM-R-based model achieved the best results for source and target metrics on RO-EN, with competitive results on ET-EN, for which explanations from a RemBERT-based model ranked first for source metrics. Despite being a simple strategy, we usually got ∼2 more points of AUC, AP, and R@K by averaging attention explanations. We note that explanations from XLM-R-M and RemBERT perform well on the 20 sentences made available by the shared task for zero-shot LPs. Between XLM-R and XLM-R-M, explanations from the latter lead to better results for both DE-ZH and RU-DE, suggesting that the additional data from the Metrics shared task might help to improve the robustness for zero-shot LPs. Ensembling explanations also leads to higher performance for zero-shot LPs. However, we note that results for DE-ZH and RU-DE are noisy due to the small amount of validation data.

5.2 Unconstrained Track

In this track, we used the predicted probabilities of BAD tags from supervised word-level QE models as explanation scores. The results are shown in Table 4 (right). As found in the constrained track, XLM-R and RemBERT-based models perform better for in-domain LPs, while XLM-R-M and RemBERT lead to better results for zero-shot LPs. Consistent with our findings in the constrained track, ensembling explanations also reflects in improvements in this track.

LP	Source			Target		
	AUC	AP	R@K	AUC	AP	R@K
RO-EN	0.856	0.727	0.621	0.881	0.783	0.678
ET-EN	0.863	0.757	0.640	0.824	0.740	0.630
DE-ZH	0.731	0.495	0.356	0.707	0.475	0.336
RU-DE	0.770	0.626	0.518	0.755	0.581	0.468
RO-EN	0.934	0.813	0.708	0.940	0.844	0.745
ET-EN	0.935	0.854	0.768	0.922	0.851	0.762
DE-ZH	0.668	0.468	0.322	0.673	0.500	0.369
RU-DE	0.848	0.709	0.593	0.806	0.633	0.515

Table 5: Official test set results for constrained (top) and unconstrained (bottom) tracks.

6 Official results

The official results of the shared task are shown in Table 5 for all LPs. Our final submissions consist of ensembled explanations since they proved to perform better for all LPs in both tracks. More specifically, we ensembled Attention × Norm explainers from the models shown in Table 4 (left) for the constrained track; and we ensembled the pre-

dicted probabilities of BAD tags from the models shown in Table 4 (right) for the unconstrained track. Overall, results for the unconstrained track are superior to those obtained in the constrained track. However, the opposite is true for DE-ZH, suggesting that extracting rationales from a sentence-level QE model is a promising weak-supervised strategy to identify translation errors.

7 Conclusion

Final remarks. We have shown that the multi-head mechanism—the bedrock on which transformers are built—is able to learn the importance of tokens associated with BAD tags. Furthermore, composing explanations in the form of attention probabilities scaled by the norm of value vectors leads to further improvements (Kobayashi et al., 2020). Ensembling these explanations yields the best results overall for all tested metrics on all LPs, including zero-shot ones.

Future work. Transformers are composed of many parameters across a vast amount of heads and layers. Strategies that explore how explanations are formed as we move to upper layers are promising, such as computing attention flows and differentiable binary masks per layer (Abnar and Zuidema, 2020; De Cao et al., 2020). Moreover, as shown in Figure 4, we noticed that our best explainers suffer on sentences with higher quality, likely due to the low number of translation errors for those sentences. A simple way to circumvent this problem is to force the explainer to "focus" on words associated with lower scores (or to the BAD class in a classification setting). Thus, strategies such as framing the prediction of DA scores as a classification problem or inducing class-wise rationalizers (Chang et al., 2019) can be helpful.

This shared task focused only on the intersection between explainability and Quality Estimation, yet for future work we plan to apply explainability methods to recent MT metrics such as COMET (Rei et al., 2020a,b; Glushkova et al., 2021) and BLEURT (Sellam et al., 2020a,b).

Acknowledgements

This work was supported by the P2020 programs MAIA (contract 045909) and Unbabel4EU (contract 042671), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

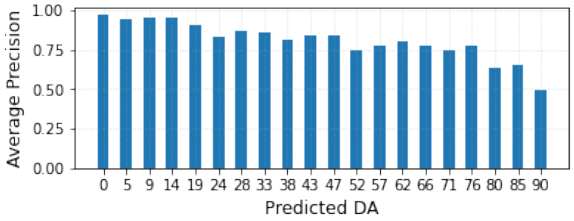

Figure 4: Predicted DAs vs. the average precision of our best explainer on the validation set of RO-EN.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

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.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

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

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2019. A Game Theoretic Approach to Class-wise Selective Rationalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Lan-

guage 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.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, 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. 2021a. The Eval4NLP Shared Task on Explainable Quality Estimation: Overview and Results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*.

Marina Fomicheva, Lucia Specia, and Nikolaos Aletras. 2021b. Translation Error Detection as Rationale Extraction. *arXiv preprint arXiv:2108.12197*.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020. MLQE-PE: A Multilingual Quality Estimation and Post-Editing Dataset. *arXiv preprint arXiv:2010.04480*.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

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*, Online. 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.

Nuno Miguel 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 (EMNLP)*, Online. Association for Computational Linguistics.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-Attention Attribution: Interpreting Information Interactions Inside Transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12963–12971.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019a. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84, Florence, Italy. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019b. 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.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167, Online. 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.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables.

Marianna Martindale and Marine Carpuat. 2018. Fluency over adequacy: A pilot study in measuring user trust in imperfect MT. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 13–25, Boston, MA. Association for Machine Translation in the Americas.

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.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

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

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. 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.

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.

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.

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, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

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

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.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André Martins. 2021. IST-Unbabel 2021 Submission for the Quality Estimation Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

A Computing infrastructure

Our infrastructure consists of 5 machines with the specifications shown in Table 6. The machines were used interchangeably, and all experiments were executed in a single GPU. Despite having machines with different specifications, we did not observe large differences in the execution time of our models across distinct machines.

#	GPU	CPU
1	4 × Titan Xp - 12GB	16 × AMD Ryzen 1950X @ 3.40GHz - 128GB
2	4 × GTX 1080 Ti - 12GB	8 × Intel i7-9800X @ 3.80GHz - 128GB
3	3 × RTX 2080 Ti - 12GB	12 × AMD Ryzen 2920X @ 3.50GHz - 128GB
4	3 × RTX 2080 Ti - 12GB	12 × AMD Ryzen 2920X @ 3.50GHz - 128GB
5.1	4 × Quadro RTX 6000 - 24GB	12 × Intel Xeon Silver 4214 @ 2.20GHz - 256GB
5.2	4 × RTX 2080 Ti - 12GB	12 × Intel Xeon Silver 4214 @ 2.20GHz - 256GB

Table 6: Computing infrastructure.

B Training hyperparameters

The hyperparameters used for training are shown in Table 7.

HYPERPARAM.	XLM-R	XLM-R-M	REMBERT
Feed-forward size	1024	1024	1024
Batch size	2	2	1
Optimizer	Adam	Adam	Adam
Number of epochs	10	10	10
Early stopping patience	3	3	3
Encoder learning rate	1×10^{-4}	1×10^{-4}	3×10^{-5}
Feed-forward learning rate	1×10^{-4}	1×10^{-4}	1×10^{-5}
Gradient accumulation	4	4	8
Dropout	0.05	0.05	0.05

Table 7: Hyperparmeters used for training sentence (constrained) and word-level (unconstrained) QE systems.

C Full results for the constrained track

Following the analysis described in §5.1, we report the best results for each explainability method for XLM-R-based models in Table 8 on the validation set of RO-EN and Table 9 on the validation set of ET-EN. We also report the best explainers based on Attention × Norm for XLM-R-M and RemBERT-based models. For explainability methods based on attention weights, we show two attention heads: one with the best performance on source AUC and another with the best performance on target AUC. Besides submitting ensembled explanations, we also made submissions with Attention × Norm heads that achieve the top performance on the validation set of RO-EN and ET-EN.

#	Encoder	Explainer	Source			Target		
			AUC	AP	R@K	AUC	AP	R@K
1	XLM-R	Attention - Layer 18 - Head 3	0.6555	0.4569	0.3509	0.7894	0.7189	0.6054
2	XLM-R	Attention - Layer 18 - Head 0	0.7445	0.6353	0.5164	0.7462	0.6488	0.5197
3	XLM-R	Cross-attention - Layer 18 - Head 3	0.7092	0.5461	0.4139	0.8066	0.7378	0.6293
4	XLM-R	Cross-attention - Layer 18 - Head 0	0.7514	0.6345	0.5170	0.7374	0.6254	0.4883
5	XLM-R	Attention × Norm - Layer 18 - Head 3	0.7178	0.5686	0.4372	**0.8136**	**0.7432**	**0.6342**
6	XLM-R	Attention × Norm - Layer 19 - Head 2	**0.7851**	**0.6875**	**0.5701**	0.8099	0.7301	0.6153
7	XLM-R	Gradient × Hidden States - Layer 15	0.6949	0.5629	0.4399	0.6780	0.5388	0.4044
8	XLM-R	Gradient × Attention - Layer 17	0.7104	0.5942	0.4913	0.7618	0.6747	0.5628
9	XLM-R	Integrated Gradients - Layer 15	0.6539	0.5251	0.4059	0.6560	0.5148	0.3853
10	XLM-R	LIME	0.6470	0.5160	0.3922	0.5892	0.4576	0.3300
11	XLM-R	Leave-one-out	0.6970	0.5673	0.4409	0.5921	0.4752	0.3567
12	XLM-R	Relaxed-Bernoulli Rationalizer	0.4803	0.3638	0.2483	0.5434	0.4043	0.2914
13	XLM-R-M	Attention × Norm - Layer 23 - Head 3	0.6993	0.5824	0.4571	0.7686	0.6932	0.5932
14	XLM-R-M	Attention × Norm - Layer 23 - Head 1	0.7530	0.6612	0.5479	0.7612	0.6841	0.5802
15	RemBERT	Attention × Norm - Layer 23	0.7824	0.6987	0.5901	0.7904	0.6865	0.5723
16	RemBERT	Attention × Norm - Layer 22 - Head 5	0.7842	0.6822	0.5752	0.7167	0.5549	0.4278
1	Ensemble	(5) + (6) + (15)	0.8043	0.7137	0.5970	0.8398	0.7695	0.6606
2	Ensemble	(5) + (6) + (14) + (15)	**0.8074**	**0.7203**	**0.6071**	**0.8421**	**0.7725**	**0.6624**

Table 8: Full constrained track results on the validation set of RO-EN.

#	Encoder	Explainer	Source			Target		
			AUC	AP	R@K	AUC	AP	R@K
1	XLM-R	Attention - Layer 18 - Head 3	0.6406	0.5205	0.3811	0.7094	0.6210	0.5037
2	XLM-R	Attention - Layer 18 - Head 0	0.6656	0.5619	0.4438	0.7055	0.6011	0.4779
3	XLM-R	Cross-attention - Layer 18 - Head 3	0.6587	0.5335	0.3947	0.7270	0.6396	0.5226
4	XLM-R	Cross-attention - Layer 17 - Head 13	0.7090	0.5927	0.4673	0.6788	0.5760	0.4599
5	XLM-R	Attention × Norm - Layer 18 - Head 3	0.6697	0.5540	0.4228	0.7257	0.6373	0.5200
6	XLM-R	Attention × Norm - Layer 19 - Head 2	0.7335	0.6181	0.4857	**0.7404**	**0.6477**	**0.5303**
7	XLM-R	Gradient × Hidden States - Layer 14	0.6567	0.5403	0.4156	0.6041	0.4837	0.3619
8	XLM-R	Gradient × Attention - Layer 17	0.6613	0.5597	0.4322	0.6891	0.5983	0.4798
9	XLM-R	Integrated Gradients - Layer 15	0.6194	0.4995	0.3699	0.5705	0.4649	0.3489
10	XLM-R	LIME	0.6221	0.4968	0.3606	0.5405	0.4297	0.3222
11	XLM-R	Leave-one-out	0.6584	0.5375	0.4082	0.5493	0.4494	0.3412
12	XLM-R	Relaxed-Bernoulli Rationalizer	0.4933	0.3794	0.2481	0.5406	0.4277	0.3211
13	XLM-R-M	Attention × Norm - Layer 21 - Head 8	0.6235	0.5041	0.3670	0.7122	0.6254	0.5133
14	XLM-R-M	Attention × Norm - Layer 21 - Head 9	0.5510	0.4106	0.2738	0.7068	0.6175	0.5059
15	RemBERT	Attention × Norm - Layer 23	0.7465	**0.6382**	**0.5229**	0.7085	0.5954	0.4756
16	RemBERT	Attention × Norm - Layer 23 - Head 8	**0.7501**	0.6203	0.4912	0.6758	0.5486	0.4418
1	Ensemble	(5) + (6) + (15)	**0.7467**	**0.6368**	**0.5113**	0.7545	0.6662	0.5512
2	Ensemble	(5) + (6) + (14) + (15)	0.7441	0.6366	0.5089	**0.7639**	**0.6805**	**0.5688**

Table 9: Full constrained track results on the validation set of ET-EN.