The UMD Submission to the Explainable MT Quality Estimation Shared Task: Combining Explanation Models with Sequence Labeling

Tasnim Kabir **Marine Carpuat**
Department of Computer Science
University of Maryland, College Park
{tkabir1,marine}@umd.edu

Abstract

This paper describes the UMD submission to the Explainable Quality Estimation Shared Task at the Eval4NLP 2021 Workshop on "Evaluation & Comparison of NLP Systems". We participated in the word-level and sentence-level MT Quality Estimation (QE) constrained tasks for all language pairs: Estonian-English, Romanian-English, German-Chinese, and Russian-German. Our approach combines the predictions of a word-level explainer model on top of a sentence-level QE model and a sequence labeler trained on synthetic data. These models are based on pre-trained multilingual language models and do not require any word-level annotations for training, making them well suited to zero-shot settings. Our best performing system improves over the best baseline across all metrics and language pairs, with an average gain of 0.1 in AUC, Average Precision, and Recall at Top-K score.

1 Introduction

Quality estimation (QE) is the task of predicting the quality of the machine translation (MT) output without reference translation. Predictions can be done at different levels of granularity, such as sentences or words. The explainable QE shared task (Fomicheva et al., 2021a) proposes to frame the identification of translation errors as an explainable QE task, where sentence-level quality judgments are explained by highlighting the words responsible for errors in the MT hypothesis. Given a source sentence and an MT hypothesis, systems are thus asked to provide word-level judgments of translation quality in addition to sentence-level judgments.

Our submission builds on state-of-the-art sentence-level QE models, MonoTransQuest (Ranasinghe et al., 2020a,b). As suggested by the organizers, we rely on the LIME explanation model (Ribeiro et al., 2016) to obtain word-level prediction from the MonoTransQuest model's sentence-level score. We hypothesize that synthetic examples of translation errors can help improve word-level predictions. As a result, we combine the predictions of MonoTransQuest-LIME with those of the Divergent mBERT model which addresses the related task of detecting semantic divergences in bitext (Briakou and Carpuat, 2020). Divergent mBERT model can detect fine-grained differences in bitext by learning to rank synthetic divergence examples of varying granularity. As a result, our approach does not require any word-level labels at training time. Both models are based on multilingual language models and are therefore amenable to zero-shot transfer.

Our submitted system improves over its components and over the official baseline on all tracks and on all language pairs, based on all evaluation metrics (AUC, AP, Recall at Top-K, and Pearson's correlation). Compared to the best baseline system for target languages, it improves AUC by 0.119 for Estonian-English (Et-En), 0.068 for Romanian-English (Ro-En), 0.085 for German-Chinese (De-Zh), and 0.128 for Russian-German (Ru-De). Similarly, for AP score, it has achieved an improvement of 0.095 for Et-En, 0.074 for Ro-En, 0.064 for De-Zh, and 0.13 for Ru-De. For Recall at Top-K score, it has achieved an improvement of 0.103 for Et-En, 0.071 for Ro-En, 0.045 for De-Zh, and 0.13 for Ru-De. For source language word-level scores, it achieves an average gain of 0.18 for AUC, 0.071 for AP and 0.12 for Recall at Top-K score over the average of all languages' baseline scores. Finally, for sentence-level scores, it has achieved an improvement of 0.36 for Et-En, 0.359 for Ro-En, 0.271 for De-Zh, and 0.06 for Ru-De compared to the average of all baseline models for Pearson's correlation.

2 Approach

We first describe the two components of our ensemble and then explain how they are combined.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 230–237

https://doi.org/10.26615/978-954-452-056-4_022

2.1 MonoTransQuest-LIME Model

The first ensemble component is based on one of the baselines provided by the organizers. It uses MonoTransQuest, the state-of-the-art model in the WMT 2020 QE shared task (Ranasinghe et al., 2020a,b), including for mid-resource and high-resource language pairs. This model uses a single XLM-Roberta transformer model (Ranasinghe et al., 2020a,b) trained with data released in WMT quality estimation tasks in recent years. The input of the model is the concatenation of the original sentence x_{source} and its translation x_{target}, separated by the $[SEP]$ token. Therefore, $x = x_{source}, [SEP], x_{target}$ and the model used the embedding of the $[CLS]$ token as the input of a softmax layer, and this layer F predicts the sentence-level score $F(x)$ of the translation at the sentence-level. Mean-squared-error loss is used as the objective function.

For generating word-level scores from the sentence-level scores, the toolkit LIME is suggested by the organizers. LIME explains the predictions of a black-box model by providing a local linear approximation of the model's behavior. For generating an explanation for a prediction, LIME generates neighborhood data by randomly hiding features from the instance and then learns locally weighted linear models on this neighborhood data to explain each of the classes in an interpretable way[1]. Here, LIME treats words in the input sequence as features and thus lets us generate word-level QE scores from the MonoTransQuest sentence-level QE predictions.

We use existing pre-trained MonoTransQuest models. Ranasinghe et al. (2020a,b) note that the QE task can be challenging in the practical environment where the systems have to work in a multilingual setting, so selecting appropriate models for each language pair is key. As summarized in Table 1, for the development languages (et-en, ro-en), we select existing MonoTransQuest models trained on the language pair tested. For the zero-shot test languages (de-zh, ru-de), we select existing MonoTransQuest models trained on language pairs that involve one of the two languages and English (en-zh and en-de, respectively).

Task	Model and Training Data
Et-En	TransQuest/monotransquest-da-et_en-wiki
Ro-En	TransQuest/monotransquest-da-ro_en-wiki
De-Zh	TransQuest/monotransquest-da-en_zh-wiki
Ru-De	TransQuest/monotransquest-da-en_de-wiki

Table 1: MonoTransQuest models used for each task.

2.2 Divergent mBERT

Briakou and Carpuat (2020) introduced the Divergent mBERT model which is a BERT-based model that can detect cross-lingual semantic divergences by ranking synthetic divergences of varying granularity without supervision. Cross-lingual semantic divergence refers to the difference in meaning between sentences written in different languages (Vyas et al., 2018) and therefore might correspond to some adequacy errors observed in MT output.

The Divergent mBERT model is designed to make both sentence-level and word-level predictions. The input of this model is a sequence x generated by concatenating an English sentence x_e and a French sentence x_f with helper delimiter tokens. Therefore, $x = ([CLS], x_e, [SEP], x_f, [SEP])$. Here, the $[CLS]$ token serves as the representative for the sentence-pair x which is passed through a feed-forward network F to get the score $F(x)$ which is converted into the probability that x is equivalent.

For word-level prediction, the final hidden state h_t is passed through a feed-forward layer and a softmax layer for each token y_t in encoded sentence pair x. This produces the probability that the token y_t belongs to the equivalent class. For sentence-level prediction, the model uses margin-loss and for token-level prediction, it uses cross-entropy loss of all tokens. The word-level evaluation on this model found that it outperforms Random Baseline across all metrics. Therefore, this model proves that we can benefit from training even with noisy word-level labels. We can map this task to identifying the error in the word-level QE by marking all divergences as errors.

We made a small change to the original Divergent mBERT model by fine-tuning XLM-Roberta (Conneau et al., 2020) rather than mBERT, and keeping the rest of the model architecture, loss definition, and training data unchanged. As a result, this model is trained on French-English sentence pairs, where positive examples of equivalence are drawn from bitext with a filtering step to ensure that they are not noisy, and negative samples are automatically generated by corrupting the positive samples to introduce meaning mismatches (e.g.,

[1] https://github.com/marcotcr/lime

by deleting dependency subtrees in one language, substituting words with near-synonyms, or phrases with other phrases that have the same syntactic structure). As a result, this model is used in zero-shot settings for all the test languages of the shared task and does not use any manual QE annotation.

2.3 Ensembling Method

We adopt the approach of Kepler et al. (2019) for building ensemble models for word-level quality estimation, which simply averages the predictions of the ensemble components. While their ensemble had five models, we average the predictions of the two models above, either at the sentence or word level. Given a source sentence (src) and the machine translation hypothesis (mt), Divergent mBERT and MonoTransQuest-LIME produce word-level scores for each word in the MT hypothesis. These are averaged to produce the final word-level score. The same process is used to combine sentence-level predictions. The overall system architecture is shown in Figure 1 for word-level predictions.

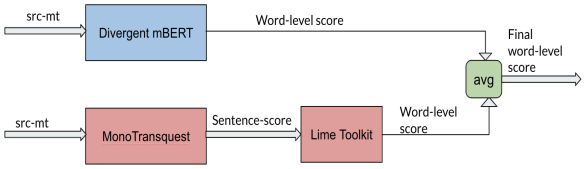

Figure 1: System architecture of the ensemble method. The input src-mt represents the language pair for which the sentence-level and the word-level score are being generated.

3 Datasets

In this section, we describe the data used for training, development, and evaluation.

3.1 Training Data

We use ensemble components that have been pre-trained on different datasets.

MonoTransQuest We use the original models that have been publicly released. They were trained on publicly available datasets from recent WMT sentence-level quality estimation tasks (Specia et al., 2018; Fonseca et al., 2019; Specia et al., 2020). These datasets were collected from Wikipedia and Reddit. In this setup, the Et-En and Ro-En are considered as medium resource language and En-Zh and En-De are considered as high resource language pairs. In Table 1, we can see the lists the training data used to train the original pre-trained MonoTransQuest models. We can note that as there were no De-Zh and Ru-De language pairs used, thus, this model supports prediction in a zero-shot setting.

Divergent mBERT We used the same training data as the original model by Briakou and Carpuat (2020). The training data was the English and French text from WikiMatrix which was normalized with Moses toolkit and tokenized. In our model, we have used "XLMRobertaTokenizer" where the original model used "BERTTokenizer". Similar to the original model, the alignment of English and French bitext was done using Berkeley word aligner. After filtering the noisy samples, the top 5500 samples, ranked by LASER similarity score, were picked, and then the synthetic divergent examples were generated. The synthetic data was generated similar to the original model's synthetic data generation process which is: subtree deletion by deleting a randomly selected subtree in the dependency parse of the English sentence, or French words aligned to English words in that subtree, Phrase Replacement by substituting random source or target sequences by another sequence of words with matching POS tags and lexical substitution by substituting English words with hypernyms or hyponyms from WordNet.

3.2 Development Data

We used the official shared task development data, which is drawn from the Multilingual Quality Estimation and Post-Editing (MLQE-PE) dataset (Fomicheva et al., 2020). There are two language pairs in the development set, with 1000 sentences each: Estonian-English (Et-En) and Romanian-English (Ro-En).

3.3 Test Data

The test data included four language pairs: Estonian-English (Et-En), Romanian-English (Ro-En), German-Chinese (De-Zh), and Russian-German (Ru-De). The first two language pairs are the same as the development set language pairs. German-English (De-Zh) and Russian-German (Ru-De) language pairs are zero-shot languages since they were not available in the development phase. Test set statistics are given in Table 2. Models are evaluated against human annotations by submitting to the official leaderboard.

Language Pair	# of sentences
Et-En	1000
Ro-En	1000
De-Zh	1410
Ru-De	1180

Table 2: Test data statistics.

4 System Configuration

We used the same set of system configurations for all the language pairs in our experiment to ensure consistency among all language pairs.

MonoTransQuest We have used pre-trained MonoTransQuestmodel on the HuggingFace Transformers library (Wolf et al., 2019). We used those pre-trained MonoTranquest models [2]. We have not changed any hyperparameter from those models.

Divergent mBERT For training Divergent mBERT we have used a batch size of 16, Adam optimizer with learning rate $2e^{-5}$ and a linear rate warmup. The model was trained with only training data. The model was trained for five epochs. We have varied the hyperparameter settings: epoch was varied from 3 to 15 epochs, the margin was varied from 5 to 10, the alpha value was varied from 0.2 to 1. Table 3 is the list of the final hyper-parameter settings we used for training in our experiments.

Batch Size	16
Loss Function	Sentence-level: Margin Loss Token-level: Cross Entropy
Optimizer	AdamW
Learning Rate	$2e^{-5}$
Scheduler	Linear Schedule with Warmup
Epoch	5
Margin	5
Alpha	1

Table 3: Divergent mBERT hyperparameters.

5 Evaluation Metrics

This section describes the official evaluation metrics for the shared task. For sentence-level scores, Pearson's correlation is used and for word-level scores, AUC, AP, and Recall at Top-K are used:

- **Pearson's Correlation:** This measures the strength and direction of a linear relationship between the model predicted sentence-level score and human-annotated sentence-level score. Values always range between -1 (strong negative relationship) and +1 (strong positive relationship).

- **AUC Score:** In this shared task, the AUC score between the model predicted output and gold explanation MT score is computed using sklearn[3]. Given a test set of N sentences:

$$AUC = \frac{1}{N}\sum_{n} AUC_n(w_n, a_n^{x^T}) \quad (1)$$

Here, w_n is a vector representing binary gold word-level labels for each sentence n in the test set and $a_n^{x^T}$ is the vector for the model predicted word-level score for the target words x^T in each target sentence in test set with length T. Equation 1 computes the AUC score to compare the model predicted word-level scores a against binary gold labels (Fomicheva et al., 2021b). Here, AUC_n is the area under the curve generated by plotting the true positive against false positive of the word-level scores of the n^{th} sentence at different thresholds. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

- **AP Score:** AP (Average Precision) evaluates word-level predictions and complements AUC scores which can be overly optimistic for imbalanced data (Fomicheva et al., 2021b). Average precision[4] is defined as:

$$AP = \sum_{n}(R_n - R_{n-1})P_n \quad (2)$$

where P_n and R_n are the precision and recall at the n^{th} threshold, where words are assigned to the positive class if the model predicts a score for this word that is higher than the n^{th} threshold.

- **Recall at Top-K:** This metric checks whether the highest predicted values have been assigned to the words corresponding to actual errors. For example, if the gold standard output is $1, 1, 0, 0, 0, 0, 0$ then the recall value

[2] https://huggingface.co/MonoTransQuest

[3] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[4] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html

checks whether in the model predicted output, the highest values have been assigned to the first and second word. Specifically this metric computes the proportion of words with the highest attribution corresponding to errors against the total number of errors in the MT output (Fomicheva et al., 2021b):

$$Recall\ at\ Top\text{-}K = \frac{1}{k} \sum_{j \in e_{1:k}} w_j \quad (3)$$

Here, k is the number of errors in the sentence and $e = argsort(a^{x^T})$ is the sequence of highest to lowest sorted indexes of target words according to the attribution scores. The final score is the average over test instances, and ranges from 0 to 1.

6 Results

We describe the performance of our models on the development and test sets using the official shared task metrics.

6.1 Development Set Results

The results achieved by each of the organizers provided baseline systems, Divergent mBERT, and our ensemble method is described in Table 4. This table described the results on the Et-En and Ro-En language pairs of the validation set. Our ensemble model outperforms the baselines, as well as its components according to all metrics.

We can see a consistent improvement for all the language pairs in the development set. In comparison with the average of all baselines, for Et-En language pair, on target word-level scores, our ensemble method achieves an improvement of 0.15 in AUC score, 0.17 in AP score, and 0.17 in Recall at Top-K score. Similarly, for the source language, it achieves an improvement of 0.19 in the AUC score, 0.123 in AP score, and 0.2 in Recall at Top-K score over the average of all baselines. Similarly, For the Ro-En language pair, on target word-level scores, our ensemble method achieves an improvement of 0.15 in AUC score, 0.21 in AP score, and 0.231 in Recall at Top-K score over the average of all baselines. Similarly, for the source language, it achieves an improvement of 0.185 in the AUC score, 0.156 in AP score, and 0.23 in Recall at Top-K score over the average of all baselines.

The Divergent mBERT model has a smaller but consistent advantage over all the baseline models. If we take the average on baselines on Et-En, this model achieves an improvement of 0.126 on AUC, 0.043 on AP, and 0.126 on Recall at Top-K score. Similarly, on the average of all baselines for Ro-En, this model achieves an improvement of 0.152 on AUC, 0.1 on AP, and 0.187 on Recall at Top-K score. Overall, these results suggest that Divergent mBERT and MonoTransQuest have complementary strengths, which benefit the ensemble.

We illustrate the complementarity of the ensemble components with randomly selected examples from the Ro-En development set in Table 5. The Divergent mBERT model predicts better error labels for short sentences than the MonoTransQuest-model. However, MonoTransQuest is more accurate on longer or more complex sentences.

6.2 Word-level Test Results

Table 6 summarizes model performance for all the language pairs in the test set. Consistent with the development set results, the ensemble improves over all the baselines, and outperforms each of its components.

Et-En : For Et-En language pair, on target word-level scores, our ensemble method achieves an improvement of 0.164 in AUC score, 0.186 in AP score, and 0.181 in Recall at Top-K score over the average of all baselines. Similarly, for the source language, it achieves an improvement of 0.236 in the AUC score, 0.13 in AP score, and 0.208 in Recall at Top-K score over the average of all baselines.

Ro-En For Ro-En language pair, on target word-level scores, our ensemble method achieves an improvement of 0.129 in AUC score, 0.173 in AP score, and 0.135 in Recall at Top-K score over the average of all baselines. Similarly, for the source language, it achieves an improvement of 0.22 in the AUC score, 0.102 in AP score, and 0.193 in Recall at Top-K score over the average of all baselines.

De-Zh For De-Zh language pair, on target word-level scores, our ensemble method achieves an improvement of 0.13 in AUC score, 0.1 in AP score, and 0.09 in Recall at Top-K score over the average of all baselines. Similarly, for the source language, it achieves an improvement of 0.123 in the AUC score, 0.01 in AP score, and 0.08 in Recall at Top-K score over the average of all baselines.

Ru-De For Ru-De language pair, on target word-level scores, our ensemble method achieves an im-

Pair	System	Word-level Score						Sentence-level Score
		Target			Source			
		AUC	AP	Recall	AUC	AP	Recall	Pearson's
Et-En	*Random (Baseline 1)*	0.505	0.387	0.284	0.496	0.380	0.249	-0.048
	XMover-SHAP (Baseline 2)	0.583	0.456	0.352	0.513	0.394	0.262	0.415
	TransQuest-LIME (Baseline 3)	0.592	0.510	0.402	-1.00	-1.00	-1.00	0.722
	Divergent mBERT	0.686	0.494	0.472	0.608	0.403	0.357	0.572
	Ensemble	**0.710**	**0.621**	**0.515**	**0.695**	**0.510**	**0.459**	**0.772**
Ro-En	*Random (Baseline 1)*	0.488	0.359	0.239	0.505	0.374	0.254	-0.021
	XMover-SHAP (Baseline 2)	0.638	0.464	0.339	0.541	0.384	0.265	0.638
	TransQuest-LIME (Baseline 3)	0.619	0.552	0.439	-1.00	-1.00	-1.00	0.882
	Divergent mBERT	0.734	0.557	0.526	0.618	0.412	0.372	0.742
	Ensemble	**0.728**	**0.664**	**0.570**	**0.708**	**0.535**	**0.486**	**0.890**

Table 4: Word-level and sentence-level scores on development data. The baseline scores are taken from the leader-board. Best results for each language by any method are marked in bold.

Source Sentence 1	Dobridorul nu este o excepție în ceea ce privește depozitele de
Target Sentence 1	The acquirer is not a deposit exception
Gold word-level label	1 1 0 0 1 1 1
MonoTransQuest label	-0.034 -0.090 -0.043 -0.023 0.006 -0.039 -0.096
Divergent mBERT label	1 1 0 0 0 0 0
Source Sentence 2	Dacă IA este programată pentru „ ” obiectivele pot fi induse implicit prin recompensarea unor tipuri de comportament sau prin pedepsirea altora.
Target Sentence 2	This is because if IA is scheduled for another the objectives can be induced implicitly by rewarding some types of behaviour or by punishing others.
Gold word-level label	1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MonoTransQuest label	0.002 0.013 0.040 -0.005 -0.022 -0.017 -0.007 -0.044 -0.023 0.012 -0.007 -0.007 -0.052 -0.027 -0.004 -0.034 -0.02 -0.003 -0.025 -0.006 -0.009 -0.002 -0.020 -0.018 0.001 -0.019 -0.005 0.001
Divergent mBERT label	0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 5: Examples of word-level labels for different models.

Pair	System	Training Data	Target			Source		
			AUC	AP	Recall	AUC	AP	Recall
Et-En	*Random (Baseline 1)*	-	0.497	0.358	0.274	0.487	0.339	0.194
	XMover-SHAP (Baseline 2)	Et-En	0.616	0.441	0.338	0.535	0.371	0.231
	TransQuest-LIME (Baseline 3)	Et-En	0.624	0.536	0.424	0.544	0.440	0.309
	Divergent mBERT	En-Fr	0.725	0.536	0.493	0.544	0.440	0.309
	Ensemble Method	Et-En, En-Fr	**0.743**	**0.631**	**0.527**	**0.758**	**0.514**	**0.453**
Ro-En	*Random (Baseline 1)*	-	0.516	0.311	0.187	0.500	0.280	0.150
	XMover-SHAP (Baseline 2)	Ro-En	0.666	0.438	0.295	0.534	0.292	0.148
	TransQuest-LIME (Baseline 3)	Ro-En	0.634	0.523	0.415	0.478	0.351	0.243
	Divergent mBERT	En-Fr	0.717	0.462	0.452	0.478	0.351	0.243
	Ensemble Method	Ro-En, En-Fr	**0.734**	**0.597**	**0.486**	**0.724**	**0.410**	**0.373**
De-Zh	*Random (Baseline 1)*	-	0.496	0.294	0.174	0.500	0.300	0.174
	XMover-SHAP (Baseline 2)	WMT's all language pairs	0.545	0.334	0.220	0.474	0.287	0.159
	TransQuest-LIME (Baseline 3)	WMT's all language pairs	0.460	0.271	0.145	0.486	0.317	0.196
	Divergent mBERT	En-Fr	0.556	0.303	0.238	0.478	**0.351**	0.243
	Ensemble Method	En-Zh, En-Fr	**0.630**	**0.400**	**0.265**	**0.610**	0.311	**0.252**
Ru-De	*Random (Baseline 1)*	-	0.492	0.308	0.216	0.506	0.341	0.237
	XMover-SHAP (Baseline 2)	WMT's all language pairs	0.522	0.328	0.224	0.522	0.356	0.259
	TransQuest-LIME (Baseline 3)	WMT's all language pairs	0.404	0.262	0.164	0.534	**0.427**	0.320
	Divergent mBERT	En-Fr	0.579	0.418	0.321	0.478	0.351	0.243
	Ensemble Method	En-De, En-Fr	**0.650**	**0.458**	**0.354**	**0.658**	0.413	**0.373**

Table 6: Word level results for all language pairs on the test set in terms of AUC, AP and Recall at Top-K. The baseline scores are taken from the leader-board. Best results for each language by any method are marked in bold.

provement of 0.18 in AUC score, 0.16 in AP score, and 0.153 in Recall at Top-K score over the average of all baselines. Similarly, for the source language, it achieves an improvement of 0.14 in the AUC score, 0.04 in AP score, and 0.101 in Recall at Top-K score over the average of all baselines.

Discussion Taken together, these results show that the ensemble method performs similarly for those language pairs that are used in the training phase (Et-En, Ro-En language pairs) and for the

zero-shot language pairs (De-Zh, Ru-De). It has an average improvement of 0.15 in AUC, 0.18 in AP, and 0.16 in Recall at Top-K score for those language pairs which is in its training data and an average improvement of 0.16 in AUC, 0.13 in AP, and 0.12 in Recall at Top-K score for those language pairs which it is not trained on. Therefore, we can see we can have a consistent gain for all language pairs with this method without including that particular language pair in the training data. The Divergent mBERT model is effective on all language pairs, even though it is trained on synthetic data generated from English-French bitext: this suggests that the word-level weak supervision provided by the synthetic samples is robust, although it would be interesting to investigate the impact of the choice of training languages further in future work. Secondly, a clear takeaway is that the ensembling of different systems can give large gains, even if some of the subsystems are weak individually and even in zero-shot settings.

6.3 Sentence-level Scores on Test Set

Pair	System	Pearson's
Et-En	*Random (Baseline 1)*	-0.029
	XMover-SHAP (Baseline 2)	0.494
	MonoTransQuest (Baseline 3)	0.772
	Divergent mBERT	0.021
	Ensemble	**0.772**
Ro-En	*Random (Baseline 1)*	0.017
	XMover-SHAP (Baseline 2)	0.695
	MonoTransQuest (Baseline 3)	0.899
	Divergent mBERT	0.661
	Ensemble	**0.899**
De-Zh	*Random (Baseline 1)*	0.000
	XMover-SHAP (Baseline 2)	0.336
	MonoTransQuest (Baseline 3)	0.335
	Divergent mBERT	0.096
	Ensemble	**0.495**
Ru-De	*Random (Baseline 1)*	-0.017
	XMover-SHAP (Baseline 2)	0.252
	MonoTransQuest (Baseline 3)	**0.498**
	Divergent mBERT	0.449
	Ensemble	0.303

Table 7: Results of sentence-level submissions and their performance on the test set. The baseline scores are taken from the leaderboard. Best results for each language by any method are marked in bold.

Table 7 contains the results for the sentence-level submission on the test set. Evaluated on Pearson's correlation, our ensemble method has a consistent improvement of 0.36 for Et-En, 0.359 for Ro-En, 0.271 for De-Zh, and 0.06 for Ru-De compared to the average of all baseline models. However, in a zero-shot setting, Pearson's correlation varies significantly between different language pairs.

We observe that the MonoTransQuest baseline achieves better performance on test language pairs, which is not surprising since it was trained on all the language pairs of WMT. This impacts results on the zero-shot languages: for De-Zh MonoTransQuest outperforms Divergent mBERT by a large margin, but the ensemble still benefits from Divergent mBERT. For Ru-De, the MonoTransQuest achieve the strongest level correlation and Divergent mBERT does not improve over it when added to the ensemble, unlike for word-level predictions.

The Divergent mBERT model has unequal performance across languages. It achieves the highest Pearson Correlation for Ro-En, which is the closest language pair to the one it is trained on (English-French) but performs poorly for Estonian-English and German-Chinese.

7 Conclusion

We described the University of Maryland's contribution to the Eval4NLP 2021 Shared Task on Quality Estimation. Our submission was based on ensembling existing models: (1) the state-of-the-art framework MonoTransQuest model followed by the LIME explanation model, and (2) an mBERT model trained to detect cross-lingual semantic divergences. We show that averaging the prediction of these models outperforms all the baselines and their individual predictions, even though none of the ensemble components are trained with word-level supervision.

Overall, our approach shows the benefits of leveraging pre-trained multilingual LMs to port to multiple language pairs, including in zero-shot settings: the Divergent mBERT component is even trained on a language pair that is not used for any of the test tasks. In the future, training Divergent mBERT with other language pairs can lead to more promising results. This work also shows the complementarity of explanation models and of sequence labelers trained on synthetic data for word-level predictions. In future work, controlled comparison of these approaches on the same languages and data conditions can lead to further insights on their respective strengths and weaknesses.

References

Eleftheria Briakou and Marine Carpuat. 2020. Detecting fine-grained cross-lingual semantic divergences without supervision by learning to rank. In *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 1563–1580, Online. Association for Computational Linguistics.

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 *ACL*.

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 FT 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.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019. Unbabel's participation in the wmt19 translation quality estimation shared task. *WMT 2019*, page 80.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*.

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*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

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, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André FT Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. 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 1503–1515.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.