Explaining Errors in Machine Translation with Absolute Gradient Ensembles

Melda Eksi **Erik Gelbing** **Jonathan Stieber** **Chi Viet Vu**
Department of Computer Science
Technical University of Darmstadt
Germany
{melda.eksi, erik.gelbing, jonathan.stieber, chiviet.vu}
@stud.tu-darmstadt.de

Abstract

Current research on quality estimation of machine translation focuses on the sentence-level quality of the translations. By using explainability methods, we can use these quality estimations for word-level error identification. In this work, we compare different explainability techniques and investigate gradient-based and perturbation-based methods by measuring their performance and required computational efforts[1]. Throughout our experiments, we observed that using absolute word scores boosts the performance of gradient-based explainers significantly. Further, we combine explainability methods to ensembles to exploit the strengths of individual explainers to get better explanations. We propose the usage of absolute gradient-based methods. These work comparably well to popular perturbation-based ones while being more time-efficient.

1 Introduction

Building trustworthy and reliable Machine Translation (MT) systems has been a broad topic in Natural Language Processing (NLP) for the past decade. Advances in research have led to dominant architectures like the BERT transformer models (Devlin et al., 2019; Wolf et al., 2020), which got widely adapted by the NLP community for a variety of tasks, including automated MT. Pretraining models (McCann et al., 2017; Devlin et al., 2019; Liu et al., 2019) on generic corpora has simplified the deployment of performant, yet easily adaptable models, and therefore became a central part of new translation systems which show great progress in terms of their translation quality while maintaining reasonable computational costs (Liu et al., 2020).

Traditionally, estimating the quality of a given translation requires human supervisors who are able to identify weaknesses in the translation, attribute wrongly translated parts and correct them manually (Comparin and Mendes, 2017). Reference-free (i.e. without access to a reference translation) Quality Estimation (QE) tries to solve this costly and time-consuming process by providing models that are able to assign quality-scores automatically (Fomicheva et al., 2020b).

TransQuest (TQ), presented by Ranasinghe et al. (2020), is a quality estimation framework which won the sentence-level direct assessment shared task in WMT 2020, thus we will utilize TQ primarily in this work as our target model to be explained.

This year's 'Explainable Quality Estimation' shared task (Fomicheva et al., 2021) focuses on the evaluation of current QE by using different explainability methods. As the organizers propose, the identification of translation errors therefore should be seen as an explainability problem. Explanations are expected to provide insights into the connection between a given input-output-pair so that humans are able to easily understand the explanation and, if necessary, take action. Further, analyzing a set of predictions of a system with explainability methods helps with comparing the system's decision with human reasoning, ultimately assessing trust in the system (Ribeiro et al., 2016).

In this work, we will compare five explainability techniques on this year's task dataset (Section 3). We experiment with perturbation-based methods as well as gradient-based methods (Section 4) and propose a simple, yet effective ensembling technique to improve the overall classification performance. Our goal is to provide an overview of each approach's capabilities in terms of classification performance, but also in the context of the computational overhead required for each approach.

2 Related work

Explainability is already a key research topic for Computer Vision (CV), but there are many rising efforts to make NLP models explainable as well

[1] Code for this paper is available at https://github.com/SinisterThaumaturge/MetaScience-Explainable-Metrics

(Pröllochs et al., 2019; Rajani et al., 2019). It can be helpful for different tasks, e.g. automated fact-checking for public health (Kotonya and Toni, 2020). Frameworks for interpretability were already developed for CV, e.g. iNNvestigate or InterpretML (Alber et al., 2019; Nori et al., 2019) but specific ones for NLP are also on the rise, like AllenNLP Interpret (Wallace et al., 2019).

Essential for ensuring trust in models are evaluation metrics that are more aligned with human perception of 'goodness'. For this reason, there are rising efforts in research for the creation of such metrics for various NLP tasks.

Existing popular evaluation metrics for text generation tasks like ROUGE (Lin, 2004) for text summarization and BLEU (Papineni et al., 2002) for MT base their measures on statistical similarity rather than evaluating semantic similarity. Several alternative metrics aim to tackle these shortcomings: unsupervised metrics MoverScore (Zhao et al., 2019) and its reference-free extension XMover-Score (XMS) (Zhao et al., 2020) that generates better aligned cross-lingual embeddings on basis of which translation quality can be measured more similarly to how humans do it.

It should be possible to derive valuable information from sentence-level QE scores for word-level translation error identification. Until now there are no existing approaches for explaining errors in MT based on QE scores that we know of. However, in consideration of the 'Explainable Quality Estimation' shared task that is part of EMNLP 2021, we expect new approaches that address this problem.

3 Data

We conduct our experiments on the Multilingual Quality Estimation and Automatic Post-editing Dataset (MLQE-PE) (Fomicheva et al., 2020a). The training and development data for this shared task consists of Romanian-English (Ro-En) and Estonian-English (Et-En) language pairs where word-level gold labels and sentence-level gold scores are provided in addition to source sentences and MT outputs. Data is shown exemplary in Figure 1.

Our approach is to estimate word-level scores in an unsupervised fashion to show specific errors in translation. Therefore, we use explanations on sentence-level scores to get word-level scores, which are then evaluated with the given gold standard word-level scores in the development set.

The training and development data consists of 7000 and 1000 tokenized sentences respectively for both Estonian-English and Romanian-English. Sentence-level scores range in [0, 100] where higher scores indicate better translations. We predict these with the QE model. Tokens for the sentence quality score (word-level labels) are rated binary, 1 being relevant and thus responsible for low-quality scores and 0 being correct tokens.

Source: Turnul a fost distrus de cutremur , trebuind să fie recunstruit în anii următori .
MT: The earthquake destroyed the pole , having to be reunified in the years to come .
Gold Explanations Source: 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0
Gold Explanations MT: 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0
Gold Sentence Level Score: 49.667

Figure 1: Example Romanian-English sentence data from the MLQE-PE training set (Fomicheva et al., 2020a). Highlights show wrongly translated tokens.

Test data consists of further annotated Et-En and Ro-En sentences. Zero-shot test sets for German-Chinese (De-Zh) and Russian-German (Ru-De) language pairs are also provided in addition, which neither contain word- nor sentence-level annotations. We did not use the training set at all as the models were all pre-trained and evaluated the explanations on the development and test set.

4 Methods

4.1 Perturbation-based methods

Methods described in the following section follow a perturbation-based approach. As the name suggests, perturbation-based explainers perturb the inputs randomly, with the goal of observing a changing behavior on the output of the model to be explained, or even on its individual neurons (Shrikumar et al., 2016).

4.1.1 LIME

Ribeiro et al. (2016) propose the usage of Local Interpretable Model-agnostic Explanations (LIME). As a *model-agnostic* explainability technique, LIME quickly gained popularity and acceptance not just in the NLP community. The main goal of LIME is to explain any complex model $f : \mathbb{R}^d \rightarrow \mathbb{R}$ by creating a simple interpretable model $g \in G$

(e.g. a sparse linear model), that is *locally trustworthy*. This means that instead of trying to globally explain the predictions of the model, specific instances $x \in \mathbb{R}^d$ are selected and explained on a local level. While many modern NLP models such as Bidirectional Encoder Representations from Transformers (BERT), Robustly Optimized BERT Pretraining Approach (RoBERTa) or XLM-RoBERTa (XLM-R) (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020) rely on word embeddings as their input representation, these representations contradict the expectations we have for interpretable explanations. As explanations should be easily understandable, Local Interpretable Model-agnostic Explanations (LIME) is using a word-level representation that enables humans to understand the influence of each word on the decision of the underlying classifier or regressor. For each instance to be explained, the original input is transformed into an *interpretable representation* $x' \in \{0,1\}^{d'}, d' : number\ of\ words\ in\ a\ sentence$, which holds a binary vector denoting whether a certain component is present or not. To explain a specific instance, the input x' is perturbed randomly (e.g. masking for text classification). The resulting perturbations are weighted by a proximity function π_x (e.g. cosine distance for text classification). The proximity function is a distance measure where an input that has been heavily modified should have a high distance to the original, which means that the explanation will probably also differ more strongly. The weighting process ensures that the resulting explanations are locally faithful, as more similar perturbations have a higher impact on the loss of the objective function and therefore on the overall explanation. Additionally, the resulting perturbations are used to minimize an error function $\xi(x)$, e.g. linear least squares in our case.

Figure 2 shows the LIME explanations for the pre-trained MonoTransQuest (MTQ) estimator on an example translation.

4.1.2 SHAP

Lundberg and Lee (2017) propose SHapley Additive exPlanations (SHAP) which builds upon various existing explainability techniques unified in a class of *additive feature attribution methods*. To explain certain instances, SHAP values are utilized to measure the influence of features towards a certain prediction.

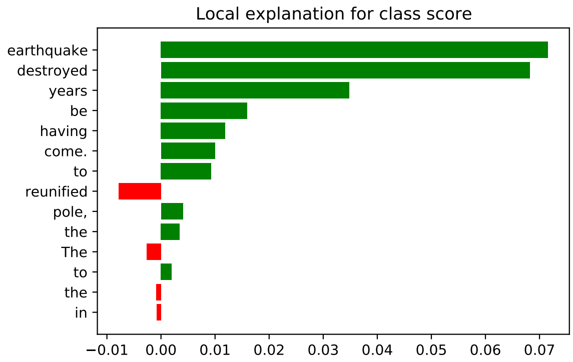

Figure 2: Explanations provided by the LIME Text Explainer (Ribeiro et al., 2016) on our example sentence. As the source sentence is fixed, we are only interested in the explanation of the target sentence. Each word can be seen as an interpretable feature holding a score that denotes its influence on the decision of the estimator.

Additive Feature Attribution Methods are a collection of methodologies that all utilize a linear function as their explanation model:

$$g(z') = \phi + \sum_{i=1}^{M} \phi_i z'_i, \tag{1}$$

The summation of the attributed effect ϕ_i of each feature from z' approaches the original model $f(x)$. Lundberg and Lee (2017) show that various current explainers like LIME or DeepLIFT (Shrikumar et al., 2017; Bach et al., 2015) match the definition in (1) and can therefore be transformed into an additive feature attribution method.

SHAP Values are based on Shapley regression values, Shapley sampling values and Quantitative Input Influence (Lipovetsky and Conklin, 2001; Bach et al., 2015; Strumbelj and Kononenko, 2014). Shapley values provide a measure for the importance of individual features and are utilized in combination with the additive feature attribution methods to derive SHAP values.

Based on these values, the SHAP explainer is able to attribute the impact of each feature to the overall prediction. SHAP conditions on one feature at a time and incrementally adds up the other features to determine ϕ_i. As the calculated effect is often dependant on the order of the features presented to the equation, ϕ_i is computed recursively for all possible orderings of features and then averaged (Shrikumar et al., 2016; Lundberg and Lee, 2017).

4.1.3 Occlusion

Occlusion is a generalization of the sliding window approach presented by Zeiler and Fergus (2014). The algorithm replaces (contiguous) patches of the input with a baseline (in our case: zero-scalar, denoting the presence or absence of a feature). Importance scores are calculated by measuring the effect of the perturbation in form of the difference in the predictions of the output layer. Therefore, we test each feature independently by comparing the output of the model if the feature is enabled (it takes its original value) versus disabling it (replacing the feature with the baseline). The resulting heatmap represents the attributions of each feature. Importance scores of the output are used to propagate their influence back to the inputs.

4.2 Gradient-based methods

Gradient-based explainers (i.e. backpropagation-based explainers) are based on the idea of propagating an importance score measured at an individual output backwards to the network (Shrikumar et al., 2016). We use these methods targeting the embedding layers of the QE model. The scores which we can propagate towards this layer serve as the explanations for individual embeddings and therefore for the overall model. Usually, these methods are more lightweight and thus require less computational overhead.

4.2.1 Layer Gradient X Activation

Layer Gradient X Activation is based on gradient*input which is a very simple explanation technique (Baehrens et al., 2010). Since the gradients represent how, and how strongly the model will behave for each input dimension, they can be seen as an expression of importance. Unfortunately, the gradient is only an accurate representation of importance locally when considering small steps. Layer Gradient X Activation is based on Shrikumar et al. (2016), which is the preceding work to DeepLIFT (Section 4.2.2). We use this approach to compute the element-wise product of gradients and activation, but contrary to regular gradient*input we only apply the method on the hidden embedding layer of the quality estimator, in order to retrieve explanations.

4.2.2 DeepLIFT

DeepLIFT (Learning Important FeaTures) (Shrikumar et al., 2017) is used to explain instances by measuring the effect $C_{\Delta x_i \Delta t}$ of a feature x_i on the overall prediction if set to a predefined reference value compared to its true value. The summation of effects for each input:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t \tag{2}$$

is called the *summation-to-delta* property with t being the reference activation of the output and Δt representing the *difference-from-reference*. Equation (2) conducts that the difference-from-reference can be determined for each x_i in the context of the reference value.

As the authors propose, DeepLIFT uses a set of rules (i.e. Linear, Rescale and RevealCancel) for assigning importance scores which can be seen as approximations of Shapley values (Section 4.1.2). In fact, the linear and rescale rules were presented in an earlier version of their work whereas the reveal-cancel rule was published later as an improved version, avoiding specific pitfalls which they describe in their work in detail. These rules are used to map the contribution scores of neurons to their immediate inputs, and further, to any input for a given target output by utilizing backpropagation.

We used **Integrated Gradients** (IG) (Sundararajan et al., 2017) as an additional gradient-based method during our experiments (see Appendix A), but we will not go into this method further as it does not have any noteworthy performance compared to the other methods.

4.3 Absolute methods

Throughout our experiments, we often observed that wrong words got high positive gradient explanations for the MTQ model. To further investigate this, we propose a different way of using the output of the explainers by just taking the absolute values for each word score. Therefore, we assume that wrong words do not have large negative gradients but instead that their gradient magnitude is higher than those of correctly translated words, leading to the intuition that wrong words have higher deviations in negative, but also positive direction with respect to the gradient. We labeled absolute methods with the prefix *Abs.* in our experiments (Section 5).

4.4 Ensemble methods

To improve our performance of the task dataset, we consider simple, yet effective ensembling techniques of the previously described approaches. In short, we use an unsupervised ensemble method to combine the different explainers. Therefore, we tested combining the individual explanations using the minimum, maximum or mean explanation values of all members in the ensemble. With these simple voting strategies, we were able to find ensembles that outperformed their individual members by a significant margin while maintaining reasonable computational extra costs. Our most successful ensembles are visualized in Figure 4. We report on our results in Section 5.2.1 and 5.2.2 in more detail.

5 Experiments

5.1 Experimental setup

All of our gradient-based methods (Section 4.2) as well as the Occlusion approach (Section 4.1.3) were implemented using the open source *Captum Model Interpretability Library* (Kokhlikyan et al., 2019).

We base our results primarily on the outputs of the MonoTransQuest (MTQ) model but also tried experimenting with XMover-Score (XMS) whenever possible (mentioned in Section 2). Due to difficulties of making the XMover-Score (XMS) model work with Captum, we were only able to retrieve predictions of the quality estimator for LIME and SHAP, for which we used their original implementations published by the authors.

We were presented with two strong baselines by the shared task, XMS+SHAP and MTQ+LIME, as well as a weak baseline of random explanations. We extended the baselines with an all-zero baseline for both language pairs, which serves as another weak baseline where no explanations are used at all. A detailed overview of our results can be found in Table 1 and Table 4. We evaluated each explainer for both language pairs and also used simple unsupervised ensembles of specific explainers.

We conducted our experiments with MTQ using those pre-trained models for the respective language pair. We performed additional tests for the zero-shot language pairs using the any-to-any model, as no specialized pre-trained models were available for these language pairs.

All methods were used with the initial parameters, except for LIME where we explored the difference in using 1000 samples instead of the default 5000 samples.

5.2 Word piece explanations

To receive the gradient-based explanation we use the embedding layer of the MTQ model which uses XLM-RoBERTa word embeddings. This embedding layer raises the problem of WordPiece embeddings (Wu et al., 2016): specific words are not only represented by one word embedding but can consist of multiple WordPiece embeddings where each of them will result in an explanation. The number of explanations will therefore be higher than the actual word counts in most cases (see Figure 3).

Text: Turnul a fost distrus de cutremur , trebuind să fie recunstruit în anii următori . (15)
Embedded text: _Turn, ul, _a, _fost, _distr, us, _de, _cutremur, _, _trebui, nd, _să, _fie, _recu, n, stru, it, _în, _ani, i, _următor, i, _. (23)

Figure 3: WordPiece example. Pieces without a _-prefix show that the corresponding word consists of different word pieces. Numbers in the brackets show the number of words and the number of WordPieces of the text.

In order to overcome this problem, we assume that if one WordPiece is translated wrong, the whole word translation is wrong. For the non-absolute methods specifically (we do not use the absolute explanations here) we take the minimum of the different WordPiece explanations. The underlying assumption is that if one WordPiece has a low score it indicates that this WordPiece and therefore the whole word is translated wrong. For the absolute methods, which assume that wrong words have larger absolute values, we use the maximum of the absolute WordPiece explanations.

We conducted additional experiments with different methods, e.g. using mean or absolute minimum, but the results were not on a par with the abovementioned methods for all explainers.

Since LIME and SHAP are model-agnostic and do not use WordPieces for their explanations, this problem does not occur.

5.2.1 Individual explainers

Our results show that most gradient-based methods are hardly able to compete with the baseline of MTQ+LIME. While gradient-based explainers remain clearly below the baseline's values on both datasets, we could achieve at least comparable results using DeepLIFT or Layer Gradient X Activation (LGXA) with absolute word scores. As it can

	Method	Model	Ro-En			Et-En		
			AUC	AP	RC	AUC	AP	RC
	Random Baseline		*0.488*	*0.359*	*0.239*	*0.49*	*0.378*	*0.271*
Gradient	DeepLift	MTQ	0.544	0.458	0.346	0.545	0.460	0.353
	LayerGradientXActivation		0.567	0.463	0.349	0.534	0.441	0.336
	Max. Gradient-based Ensemble		0.599	0.487	0.375	0.574	0.465	0.357
Abs. Gr.	Abs. DeepLift	MTQ	0.675	0.543	0.428	0.621	0.511	**0.404**
	Abs. LayerGradientXActivation		0.645	0.517	0.405	0.589	0.481	0.373
	Max. Abs. Gradient-based Ensemble		**0.682**	**0.552**	**0.440**	**0.622**	**0.509**	0.401
Perturbation	*LIME 5000 Samples*	MTQ	*0.619*	*0.552*	*0.439*	*0.592*	*0.510*	*0.402*
	LIME 1000 Samples		0.618	0.547	0.435	0.587	0.501	0.395
	SHAP		0.589	0.496	0.385	0.571	0.487	0.376
	Occlusion		0.590	0.517	0.405	0.533	0.463	0.352
	LIME 1000 Samples	XMS	0.639	0.489	0.371	0.608	0.467	0.358
	SHAP		*0.638*	*0.464*	*0.339*	*0.583*	*0.456*	*0.352*
	Max. Perturbation-based Ensemble	Both	0.674	0.550	0.435	0.612	0.514	0.407

Table 1: Evaluation results for different explanation methods, models and both language pairs on the development set - MTQ: MonoTransQuest, XMS: XMover-Score, AP: average precision, RC: recall on top 5. This table only shows some of the methods that we implemented. The complete list can be found in appendix A. The underlined scores show the best scores for a method type, the **bold** values show the global maximum and the *italic* values show the baselines.

be seen in Table 1, we can achieve comparable, if not even slightly improved AUC scores for these methods while maintaining mostly equal precision and recall values. In general, our experiments show that the usage of absolute word scores fails across the board for perturbation-based approaches, leading to even worse results than both weak baselines. Considering our initial observation (Section 4.3), this is not a big surprise as it is only valid for gradient-based methods. For precisely those approaches, however, we can see an improvement in every case if we consider just taking the absolute word scores of the respective explainer instead of the signed values. Our best performing approach, Abs. DeepLIFT outperforms MTQ+LIME with a difference of 0.056 in AUC on the Ro-En dataset and 0.029 for Et-En.

The baseline of XMS+SHAP has improved AUC values, but lower precision and recall scores than MTQ+LIME. Comparing our methods to XMS+SHAP reflects these differences, ultimately leading to Abs. DeepLIFT and Abs. LGXA now consequently outperforming the baseline on all measured scores. The AUC values for Abs. DeepLIFT are better than XMS+SHAP by 0.037 for Ro-En and by 0.029 for Et-En.

Across just the perturbation-based methods we can see mostly similar performances close to the XMS+SHAP baseline. Because the runtime of LIME with the default 5000 samples was too high, we only used 1000 samples. This decrease in runtime only reduced the performance marginally. For the language pair Ro-En there was only a decrease in AUC of 0.001 and 0.005 for Et-En. As the word scores for LIME were not available, we needed to calculate the word scores of LIME for our ensembles. Experiments with 1000 samples were sufficient. Occlusion, our additional perturbation-based explainer, performed worse than LIME and SHAP with AUCs of 0.59 for Ro-En and 0.533 for Et-En.

We observe that the results of LIME and SHAP with XMS are better than the MTQ results. Experiments with XMS could thus further improve the explainability performance.

5.2.2 Ensembles

We can see that the simple ensemble methods, described in Section 4.4, are able to improve the explanation score for absolute and non-absolute gradient methods. For the non-absolute method, a maximum ensemble of IG, DeepLift and LGXA results in an AUC score of 0.599 which is an improvement of 0.032 compared to the best single explainer LGXA. We see an improvement of 0.007 in the group of absolute gradient methods with the

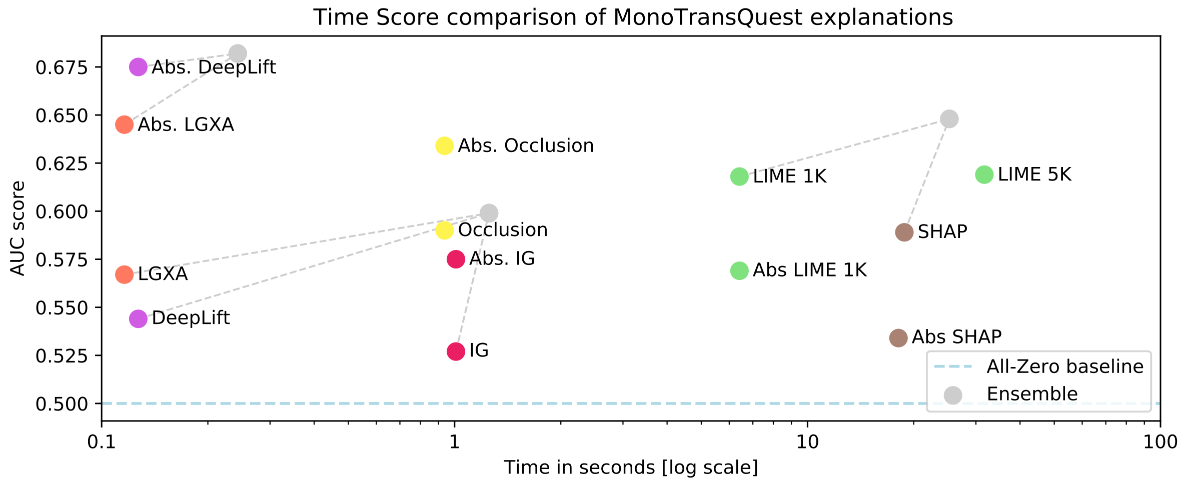

Figure 4: AUC scores on the development set for the MonoTransQuest explanations combined with explanation time of the Ro-En language pair. Notice that we use log scale which indicates that especially the perturbation-based methods LIME and SHAP take much longer compared to the gradient-based methods.

maximum ensemble of Abs. DeepLift and Abs. LGXA, reaching 0.682. There is also an improvement in the values if we use an ensemble of all perturbation-based methods. The maximum ensemble of them was better than the best performing perturbation-based method LIME+XMS by 0.035. Using all the methods and combining them into an ensemble did not increase the performance of the explainers. The all-method ensemble achieves the same AUC score as the ensemble with Abs. DeepLift and Abs. LGXA for the language pair Ro-En, but it was worse in all other metrics. Our best performing explainer is the ensemble consisting of Abs. DeepLift and Abs. LGXA.

5.3 Explanation duration

Figure 4 shows the AUC scores of the models compared to the duration in seconds for generating a single explanation. It can be seen that gradient-based methods have a massive advantage compared to perturbation-based methods in terms of the required computation time. Although we only measured execution times naively, the collected values should be sufficient to obtain a rough picture of how the individual methods compare to each other. Since perturbation-based methods do multiple forward passes while gradient-based methods use only one, we can see that gradient-based methods are much faster than perturbation-based ones overall. While LGXA only takes about 0.1 seconds, LIME with 5000 samples can take more than 30 seconds for a single sentence explanation.

Occlusion proves to achieve worse results than LIME and SHAP but with a faster execution time. However, Occlusion still requires significantly longer execution times than most gradient-based approaches, making it not competitive with the other methods.

Since our ensembles are simple operations, their duration is only the sum of the duration times of each ensemble member. As shown, our ensemble with gradient-based methods is not only significantly faster but also performs better than the baselines of LIME and SHAP.

5.4 Zero-shot explainers

We did further experiments with the provided language pairs De-Zh and Ru-De. The results can be seen in Table 2. We tried using our best-performing method, the maximum ensemble of Abs. DeepLift and Abs. LGXA. By using this setup we could easily include IG into the list of experiments. Surprisingly, absolute methods, except Abs. IG, perform worse than their non-absolute counterparts for De-Zh. The best performing method for De-Zh was the traditional ensemble consisting of all gradient-based methods. With an AUC of only 0.569 the explainer performance is insufficient. We can observe similar performances for the language pair Ru-De where the absolute methods are worse. The exception is Abs. DeepLift with the best performing AUC value of 0.621 and an AP value of 0.511 which is even better than the best score for Et-En. These experiments illustrate that the performances of the explainers vary significantly for each language pair.

	Method	Model	De-Zh			Ru-De		
			AUC	AP	RC	AUC	AP	RC
Gradient	Integrated Gradients [1]	MTQ	0.476	0.334	0.187	0.618	0.311	0.175
	DeepLift [2]		0.453	0.316	0.172	0.614	0.297	0.191
	LayerGradientXActivation [3]		0.500	0.344	0.195	0.500	0.344	0.195
	Max. Gradient-based [E]		**0.569**	**0.421**	**0.321**	0.592	0.332	0.226
Abs. Gr.	Abs. Integrated Gradients [1]	MTQ	0.547	0.386	0.250	0.520	0.405	0.302
	Abs. DeepLift [2]		0.453	0.306	0.214	**0.621**	**0.511**	**0.404**
	Abs. LayerGradientXActivation [3]		0.436	0.321	0.230	0.589	0.481	0.373
	Max. Abs. Gradient-based [E 2,3]		0.467	0.309	0.246	0.555	0.283	0.170

Table 2: Results for the zero-shot language pairs De-Zh and Ru-De on the development set (provided gold standard of 20 annotated sentence pairs). No baselines were provided. - MTQ: MonoTransQuest, [E]: Ensemble of all methods, [E 2,3] ensembles of methods 2 and 3

5.5 Shared task performance

For each language pair we chose methods that performed best on the development sets as our submission methods. Our results are shown in Table 3.

The any-to-any tokenizer had problems with tokenizing Russian and Chinese sentences, which led to the problem of getting less word level explanations than necessary. We solved this by padding the explanations with the default value of 1 as a simple fix.

For the Et-En and Ro-En language pairs, our ensembling approach outperformed the random and XMS+SHAP baselines, while performing on par with MTQ+LIME. On the Ru-De and De-Zh data, our approach hardly shows improvements over the given baselines which probably results due to the weak performance of the any-to-any model on these language pairs.

6 Discussion

The dependency on the performance of QE models is the main limitation of our approaches. Explanations are generated in a pipeline fashion where potential errors will be propagated into the explanation models. MTQ and XMS work quite well already, but there is still room for improvement, especially for language pairs that are not that similar. It is likely that we could achieve better explanations with better QEs.

Another limitation of absolute gradient-based methods is their model-awareness where explainers need access to a QE model's training procedure in order to calculate the gradients. In cases where explanations should be generated in a black-box manner, perturbation-based methods like LIME or SHAP are better suited.

We showed that perturbation-based methods work generally well for predicting errors in MT based on QE and that existing gradient-based methods perform quite poorly in comparison. Our proposed absolute gradient method is a simple extension of those existing methods, but with large performance improvements. However, absolute perturbations seem to worsen the performance of existing perturbation-based methods. Explainer ensembles outperformed single explainers in all cases, where maximum ensembles generally worked best. Absolute explanations also improved gradient-based ensembles.

Our experiments justify the popularity of perturbation-based explainers. Nonetheless, gradient-based methods should not be overlooked. They are not only faster in comparison, but with the extension of absolute explanation ensembles can also perform better for the given task and are hence worth to consider.

7 Conclusion

We showed that absolute gradient-based methods are worthy contenders to perturbation-based methods when it comes to generating plausible word-level explanations for MT. Explainer ensembles also exploit the strengths of their individual members and yield better explanations, be they perturbation-based or gradient-based ones. Gradient-based methods have the potential to be used in online applications given that they are more time-efficient than popular perturbation-based approaches, even as ensembles. Black-box models however are better explained with regular perturbation-based methods.

Language Pair	Method	AUC	AP	RC
Et-En	*MTQ+LIME*	*0.624*	*0.536*	*0.424*
Et-En	Max. Abs. Gradient-based Ensemble (Abs. DeepLift & Abs. LayerGradientXActivation)	0.658	0.516	0.404
Ro-En	*XMS+SHAP*	*0.666*	*0.438*	*0.295*
Ro-En	Max. Abs. Gradient-based Ensemble (Abs. DeepLift & Abs. LayerGradientXActivation)	0.677	0.505	0.381
De-Zh	*XMS+SHAP*	*0.545*	*0.334*	*0.220*
De-Zh	Max. Gradient-based Ensemble (DeepLift & LayerGradientXActivation)	0.513	0.311	0.185
Ru-De	*XMS+SHAP*	*0.522*	*0.328*	*0.227*
Ru-De	Abs. DeepLift	0.543	0.328	0.224

Table 3: Results on the test data of our models that performed best on the development set. All methods use MTQ as their QE model. We include the results of the best performing baseline in terms of AUC score for each language pair for comparison in *italic*.

Future work might explore training QE and explanation methods end-to-end, find better performing (multilingual) QE models, or train models on word-level information. One could also try to solve the given problem with recently proposed explanation methods that try to tackle problems of existing explainers.

Another way of improving the explanation scores might be using supervised ensemble methods on different explainers by using the training dataset to train e.g. a simple decision tree. The training dataset could be also used to finetune the QE models.

Acknowledgement

This work was created in the context of the Meta-Science Seminar at the Technical University of Darmstadt. We thank Steffen Eger for his supervision.

References

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2019. innvestigate neural networks! *J. Mach. Learn. Res.*, 20:93:1–93:8.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831.

Lucia Comparin and Sara Mendes. 2017. Using error annotation to evaluate machine translation and human post-editing in a business environment. *Proceedings of EAMT 2017, Prague, May 29*, 31.

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, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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*.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. MLQE-PE: A multilingual quality estimation and post-editing dataset. *arXiv preprint arXiv:2010.04480*.

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

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Jonathan Reynolds, Alexander Melnikov, Natalia Lunova, and Orion Reblitz-Richardson. 2019. Pytorch captum. `https://github.com/pytorch/captum`.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7740–7754. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stan Lipovetsky and Michael Conklin. 2001. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6294–6305.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Nicolas Pröllochs, Stefan Feuerriegel, and Dirk Neumann. 2019. Learning interpretable negation rules via weak supervision at document level: A reinforcement learning approach. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 407–413. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4932–4942. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5070–5081. International Committee 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*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713.

Erik Strumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. Allennlp interpret: A framework for explaining predictions of NLP models. 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 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, pages 7–12. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer.

Wei Zhao, Goran Glavas, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1656–1671. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 563–578. Association for Computational Linguistics.

A Appendix

	Method	Model	Ro-En			Et-En		
			AUC	AP	RC	AUC	AP	RC
	All-Zero Baseline		0.5	0.251	0.210	0.5	0.285	0.253
	Random Baseline		*0.488*	*0.359*	*0.239*	*0.49*	*0.378*	*0.271*
Gradient	Integrated Gradients [1]	MTQ	0.527	0.407	0.288	0.539	0.422	0.317
	DeepLift [2]		0.544	0.458	0.346	0.545	0.460	0.353
	LayerGradientXActivation [3]		0.567	0.463	0.349	0.534	0.441	0.336
	Mean Gradient-based [E]		0.561	0.465	0.349	0.554	0.461	0.360
	Max. Gradient-based [E]		0.599	0.487	0.375	0.574	0.465	0.357
	Min. Gradient-based [E]		0.505	0.405	0.295	0.518	0.429	0.329
	Mean Gradient-based [E 2,3]		0.559	0.461	0.342	0.544	0.451	0.347
	Max. Gradient-based [E 2,3]		0.577	0.479	0.367	0.560	0.463	0.360
	Min. Gradient-based [E 2,3]		0.538	0.448	0.333	0.523	0.440	0.336
Abs. Gradient	Abs. Integrated Gradients [1]	MTQ	0.575	0.433	0.317	0.520	0.405	0.302
	Abs. DeepLift [2]		0.675	0.543	0.428	0.621	0.511	0.404
	Abs. LayerGradientXActivation [3]		0.645	0.517	0.405	0.589	0.481	0.373
	Mean Abs. Gradient-based [E 2,3]		0.677	0.541	0.526	0.618	0.504	0.393
	Max. Abs. Gradient-based [E 2,3]		**0.682**	0.552	0.440	**0.622**	0.509	0.401
	Min. Abs. Gradient-based [E 2,3]		0.653	0.520	0.399	0.599	0.490	0.384
Perturbation	*LIME 5000 Samples*	MTQ	*0.619*	*0.552*	*0.439*	*0.592*	*0.510*	*0.402*
	LIME 1000 Samples		0.618	0.547	0.435	0.587	0.501	0.395
	SHAP		0.589	0.496	0.385	0.571	0.487	0.376
	Occlusion		0.590	0.517	0.405	0.533	0.463	0.352
	LIME 1000 Samples	XMS	0.639	0.489	0.371	0.608	0.467	0.358
	SHAP		*0.638*	*0.464*	*0.339*	*0.583*	*0.456*	*0.352*
	Mean Perturbation-based [E]	Both	0.627	0.550	0.435	0.585	0.510	0.404
	Max. Perturbation-based [E]		0.674	**0.581**	0.474	0.612	**0.514**	**0.407**
	Min. Perturbation-based [E]		0.571	0.475	0.369	0.535	0.434	0.327
	Mean LIME SHAP [E]		0.648	0.565	**0.462**	0.602	0.502	0.396
	Max. LIME SHAP [E]		0.648	0.557	0.443	0.594	0.504	0.396
	Min. LIME SHAP [E]		0.615	0.492	0.389	0.578	0.460	0.360
Abs. Perturbation	Abs. LIME	MTQ	0.569	0.404	0.283	0.523	0.407	0.306
	Abs. SHAP		0.534	0.377	0.258	0.582	0.445	0.336
	Abs. Occlusion		0.634	0.490	0.365	0.587	0.465	0.363
	Abs. LIME	XMS	0.401	0.311	0.200	0.426	0.334	0.232
	Abs. SHAP		0.365	0.290	0.181	0.418	0.337	0.232
	Mean Abs. Perturbation-based [E]	Both	0.564	0.412	0.288	0.549	0.425	0.327
	Max. Abs. Perturbation-based [E]		0.596	0.457	0.332	0.550	0.436	0.340
	Min. Abs. Perturbation-based [E]		0.453	0.337	0.222	0.468	0.362	0.255
All	Mean All [E]	Both	0.589	0.498	0.387	0.570	0.477	0.371
	Max. All [E]		0.618	0.495	0.381	0.589	0.469	0.360
	Min. All [E]		0.49	0.416	0.307	0.503	0.410	0.306
	Mean Abs. All [E]		0.675	0.542	0.425	0.607	0.494	0.390
	Max. Abs. All [E]		**0.682**	0.546	0.429	0.600	0.485	0.375
	Min. Abs. All [E]		0.461	0.342	0.230	0.475	0.368	0.259

Table 4: Results of various explanation methods for the Ro-En and Et-En language pairs on the development set - MTQ: MonoTransQuest, XMS: XMover-Score, [E]: Ensemble