BMX: Boosting Natural Language Generation Metrics with Explainability

Christoph Leiter[1], **Hoa Nguyen**[2], **Steffen Eger**[1]
[1] Natural Language Learning Group (NLLG)
https://nl2g.github.io/
[1] University of Mannheim, [2] TU Darmstadt
{christoph.leiter,steffen.eger}@uni-mannheim.de

Abstract

State-of-the-art natural language generation evaluation metrics are based on black-box language models. Hence, recent works consider their explainability with the goals of better understandability for humans and better metric analysis, including failure cases. In contrast, our proposed method BMX: Boosting Natural Language Generation Metrics with explainability explicitly leverages explanations to boost the metrics' performance. In particular, we perceive feature importance explanations as word-level scores, which we convert, via power means, into a segment-level score. We then combine this segment-level score with the original metric to obtain a better metric. Our tests show improvements for multiple metrics across MT and summarization datasets. While improvements in machine translation are small, they are strong for summarization. Notably, BMX with the LIME explainer and preselected parameters achieves an average improvement of 0.087 points in Spearman correlation on the system-level evaluation of SummEval.[1]

1 Introduction

Modern language model (LM) based natural language generation (NLG) metrics achieve astonishing results in grading machine generated sentences like humans would (e.g., Bhandari et al., 2020; Freitag et al., 2021b; Specia et al., 2021; Fabbri et al., 2021). As most language models are black-box components, some recent works started to explore the explainability of LM-based metrics (e.g. Fomicheva et al., 2021; Leiter et al., 2022; Sai et al., 2021; Zerva et al., 2022; Chen and Eger, 2023). This exploration, for example, contributes to the foundation of ethical machine learning (e.g. Fort and Couillault, 2016; European_Commission, 2019).

[1]We make our code available at: https://github.com/Gringham/BMX

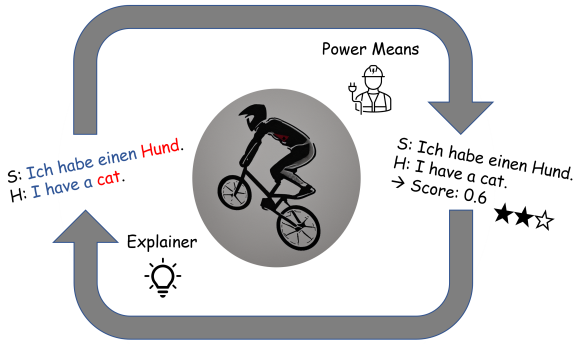

Figure 1: The duality of segment-level natural language generation evaluation metrics (right) and their word-level explanations (left).

Our work is motivated by an intriguing duality that we note between segment-level metrics and their explainability through feature importance techniques, e.g., LIME (Ribeiro et al., 2016): **Segment-level metrics**[2] return a single score indicating the quality of a generated segment. *Feature importance explanations*[3] increase the granularity of this score, by assigning additional **word-level scores**. These granular scores capture additional information about the generated text and about the metric that processed it, as, e.g., explored by the Eval4NLP21 shared task (Fomicheva et al., 2021) and the WMT22 quality estimation shared task (Zerva et al., 2022). On the other hand, in recent *multidimensional quality metrics* (MQM) datasets, word-level error annotations are converted into segment-level scores using heuristic functions (Freitag et al., 2021a). Likewise, metrics like BERTScore (Zhang et al., 2020) and BARTScore

[2]We use the term *segment-level*, as it includes the option that a metric grades multiple hypothesis sentences. Recent work shows that many sentence-level metrics also perform well on the segment-level (Deutsch et al., 2023).

[3]Also called *relevance scores* or *attribution scores*.

Findings of the Association for Computational Linguistics: EACL 2024, pages 2274–2288

(Yuan et al., 2021) build their segment-level scores upon word-level scores. In other words, we note the duality that feature importance techniques produce word-level scores from segment-level scores and heuristics can aggregate word-level scores into segment-level scores. Figure 1 gives an example of this duality for machine translation (MT), where a German source sentence "Ich habe einen Hund" was wrongly translated into "I have a cat". On the right side, a segment-level score of 0.6 is assigned by a metric. On the left side, a feature importance explainer is used to explain this score by assigning word-level scores to each input token. Instead of displaying the scores, we use colors to describe the concept. The red words would likely achieve a low importance score, as they are translated incorrectly. The duality arises as the feature importance scores can be recombined into a new segment-level score (here using power-means).

In this work, we explore whether this duality leads to iterative improvements of segment- and word-level scores, with a focus on segment-level scores as these are the main goal of modern metrics. We propose ***B**oosting natural language generation **M**etrics with e**X**plainability* (BMX), a method that directly leverages word-level explanations to improve the original segment-level score of a metric. Specifically, the approach aggregates word-level feature importance explanations using power means (Rücklé et al., 2018) and combines them with the original score using a linear combination. To obtain the explanations, we leverage model-agnostic explainability techniques, allowing application to any NLG metric. While we consider NLG (especially MT and summarization) as 'natural use case', other regression and classification tasks follow similar settings, which makes our approach more generally applicable. For example, in sentiment classification, feature importance techniques might assign high importance scores to tokens with positive sentiment. Hence, aggregating these scores could further inform a classification decision.

We evaluate BMX with several metrics and explainability techniques on 5 MT datasets (3 for exploration + 2 held out for testing), as well as 2 summarization datasets, and discuss conditions for its failure and success. Our work makes the following contributions:

(i) We highlight the duality of word-level explanations and segment-level scores for NLG metrics.

(ii) We propose an approach to improve NLG metrics by combining it with model-agnostic explainability techniques.

(iii) We provide an evaluation that shows that our approach can achieve consistent improvements. For example, after applying BMX, we obtain 0.087 points improvement on SummEval.

2 Approach

NLG metrics grade a generated text, also referred to as hypothesis, by comparing it to a ground truth. For MT, the ground truth could be a human written reference translation or the original text in source language. For summarization, the ground truth could be a human written reference summary or the source text that is being summarized. Given a pair of ground truth segment $\boldsymbol{g} = \langle g_1, ..., g_n \rangle$ and hypothesis segment $\boldsymbol{h} = \langle h_1, ..., h_m \rangle$, a segment-level metric $\mathcal{S}_0$ generates a single score $\mathcal{S}_0(\boldsymbol{g}, \boldsymbol{h}) = s_0 \in \mathbb{R}$. This score can be interpreted as, for example, the adequacy/accuracy of the generation of $\boldsymbol{h}$ given $\boldsymbol{g}$.

Our algorithm consists of three steps: (1) compute feature importance explanations, (2) aggregate explanation scores, and (3) combine the aggregated explanations with the original score.

2.1 Feature importance computation

The input of our algorithm is an arbitrary NLG metric $\mathcal{S}_0$, which we aim to improve, and a pair of ground truth and hypothesis segments $(\boldsymbol{g}, \boldsymbol{h})$. Further, we leverage a feature importance explainer ε, e.g., LIME (Ribeiro et al., 2016) or SHAP (Lundberg and Lee, 2017). We use ε to compute feature importance scores ϕ_i for each input token of an NLG metric. I.e., we explain $\mathcal{S}_0$ and its evaluation of $\boldsymbol{g}$ and $\boldsymbol{h}$ using ε and obtain $\boldsymbol{\phi} \in \mathbb{R}^{m+n}$ as follows:

$$\varepsilon(\mathcal{S}_0, \boldsymbol{g}, \boldsymbol{h}) = \langle \phi_1, ..., \phi_n, \phi_{n+1}, ..., \phi_{n+m} \rangle$$

The importance scores $\boldsymbol{\phi}$ specify the contribution of each token in $\boldsymbol{g}$ and $\boldsymbol{h}$ to s_0. Note that the metric $\mathcal{S}_0$ itself is a parameter to ε as model-agnostic explainers compare the metrics' original output with its output for permutations of the input text. For a strong metric, a high feature importance ϕ_i indicates that token $t_i \in \boldsymbol{g} \cup \boldsymbol{h}$ has a positive contribution to the score $\mathcal{S}_0$ and thus is likely to

be correctly generated[4]. Low feature importance can indicate incorrect translations or summaries. This setup follows the Eval4NLP21 shared task (Fomicheva et al., 2021) for MT. Continuing the example from figure 1, the source sentence "Ich habe einen Hund" is our $\boldsymbol{g}$ and the hypothesis sentence "I have a cat" is our $\boldsymbol{h}$; s_0 is 0.6 and the output of ε are feature importance scores corresponding to the words, e.g. $\varepsilon(\mathcal{S}_0, \boldsymbol{g}, \boldsymbol{h}) = \boldsymbol{\phi} = \langle 0.5, 0.4, 0.2, 0.0, 0.5, 0.4, 0.2, 0.0 \rangle$, where the low numbers indicate mistranslations.

In some datasets, multiple references are available for each hypothesis. In these cases, we concatenate the importance scores for each reference segment into $\boldsymbol{\phi}$.

2.2 Explanation score aggregation

As mentioned above, the feature importance scores of a reasonable metric indicate the generation quality of each token. We combine these values to estimate the quality of the hypothesis at the segment-level. Therefore, we employ an aggregation function $f : \mathbb{R}^{m+n} \rightarrow \mathbb{R}$ to transform feature importance scores generated from the previous step into a single scalar value. We obtain the aggregated explanation score $\hat{s}_0$ as follows:

$$f(\varepsilon(\mathcal{S}_0, \boldsymbol{g}, \boldsymbol{h}))) = \hat{s}_0$$

2.3 Linear combination

Finally, we linearly combine $\hat{s}_0$ and s_0 using weight w to construct a new metric $\mathcal{S}_1$:

$$\mathcal{S}_1(\boldsymbol{g}, \boldsymbol{h}) = w \cdot s_0 + (1 - w) \cdot \hat{s}_0 = s_1$$

We note that this three step process (feature importance computation, explanation score aggregation, linear combination) can be applied iteratively by increasing the index of $\mathcal{S}$ (resp. s). I.e., in the next iteration, we can consider $\mathcal{S}_1$ as the original metric and s_1 as the original score.

3 Experiment Setup

In this section, we describe the datasets, metrics, explainers and aggregation methods that we evaluate in §4 and their parameter configurations.

[4]For weak metrics, the segment-level score is incorrect more often, hence the feature importance scores are not as likely to be correlated to correct and incorrect translations.

3.1 Datasets

Our configuration of BMX has two parameters w (see §2.3) and p (see §3.4) which can either be selected *in-domain* on a labeled subset of the same dataset or *cross-domain* on a different dataset. We mainly evaluate cross-domain selection, as it would allow to apply BMX without additional annotation effort and is, therefore, more desirable. However, cross-domain tasks are generally also more difficult. For summarization, we also test an in-domain stratification approach. We refer to the datasets that we use for parameter search as *calibration datasets* and to those that we evaluate on as *evaluation datasets*.

MT datasets We use three *calibration datasets*: the **WMT17** metrics shared task (Bojar et al., 2017) newstest2017 test set in the to-English direction, the 2020 partition of the **MLQE-PE** dataset (Fomicheva et al., 2022) and the **Eval4NLP21** test set (Fomicheva et al., 2021). We evaluate BMX on two further *evaluation datasets*: The **WMT22** Quality Estimation shared task (Zerva et al., 2022) and the **MQM**[5] annotations of newstest21[6] without human written references (Freitag et al., 2021a,b). WMT17, MLQE-PE, Eval4NLP and WMT22 contain *source sentence - hypothesis* pairs and human direct assessment (DA) scores (Graham et al., 2017) that grade the translation quality. For MLQE-PE, Eval4NLP21 and WMT22, human annotators determined these scores based on source and hypothesis sentences; for WMT17 they used reference sentences instead of source sentences. For MQM (Lommel et al., 2014), scores are aggregated from fine-grained human MQM error annotations, and have been shown to be of better quality than crowd-sourced annotations (Freitag et al., 2021a). Table 5 (appendix) shows an overview of the number of samples per language pair and dataset.

Summarization datasets We perform *in-domain calibration* on **SummEval** (Fabbri et al., 2021). To do so, we apply cross-validation and split SummEval into eight non-overlapping *configuration* (7 with 208 samples and 1 with 144) and *evaluation* (7 with 1392 samples and 1 with 1456 samples) splits. Also, we make sure that no source text in the configuration set has another hypothesis in the corresponding test set. SummEval contains multiple expert-annotated discrete scores for coherence, consistency, fluency and relevance each and 11 ref-

[5]We further refer to the datasets by these bolded names.

[6]https://github.com/google/wmt-mqm-human-evaluation

erence summaries per hypothesis. We average the expert annotations for each score.

Further, we use the parameter values obtained on SummEval and perform *cross-domain calibration* on **RealSumm** (Bhandari et al., 2020). SummEval and RealSumm have the same data source, but different annotations and a different selection segments.

3.2 Base metrics

We test BMX with the following metrics.

Reference-based For summarization, we test BMX with BERTScore (Zhang et al., 2020) and BARTScore (Yuan et al., 2021).

Reference-free For MT, we test BMX with XBERTScore (Zhang et al., 2020; Song et al., 2021; Leiter, 2021)[7], XLMR-SBERT (Reimers and Gurevych, 2020), TransQuest (Ranasinghe et al., 2021) and COMET (Rei et al., 2021).

We report the exact metric configurations in Appendix A.

3.3 Explanation techniques

We explore the effectiveness of three model-agnostic explainers: *Erasure* (Li et al., 2016), *LIME* (Ribeiro et al., 2016) and *SHAP* (Lundberg and Lee, 2017) For implementation details refer to appendix D.

Multiple references: We handle the computation of the hypothesis and multiple references separately by fixing all but one during each application of the explainers and applying the explainer separately to each of them. E.g., if we have one hypothesis and 11 references and use LIME with 100 permutations, we will apply it 12 times, resulting in 1200 permutations in total.

3.4 Aggregation technique

Following Rücklé et al. (2018), we use the power mean (or generalized mean) as a generalization over different means to aggregate token-level feature-importance scores. The power mean of n positive numbers $e_1, \ldots, e_n$ is computed as:

$$M_p(e_1, ..., e_n) = \left(\frac{1}{n} \sum_{k=1}^{n} e_k^p \right)^{\frac{1}{p}}$$

Depending on p, the power mean takes on the value of specific means, e.g. $p = -1$ is the harmonic mean, $p = 1$ is the arithmetic mean, and $p = -\infty$ resp. $p = +\infty$ is the minimum resp. maximum. We experiment with p-values between $[-30, 30]$ in 0.1 steps. The token-level scores resulting from the explanation technique can be negative, which is problematic for power means, as these are defined on positive numbers only[8]. To guarantee positive importance scores, whenever there is a negative importance score for a token, we add a regularization term to all importance scores of the current ground truth/hypothesis pair. This term is the absolute value of the smallest importance score assigned to any token of this pair. Additionally, we generally add a constant $1e{-}9$ to each importance score to avoid issues with fluctuations around 0. Future work could explore further methods of aggregation such as different settings of the Kolmogorov mean (de Carvalho, 2016).

3.5 Evaluation

To evaluate the BMX metrics, we calculate the correlation on datasets with human annotated scores. E.g., we can compute Pearson correlations per sample as follows:

$$\text{Pearson}(\, H(LP, D)\,,\, \mathcal{S}_1(LP, D, \mathcal{S}_0, \varepsilon, w, p)\,) \quad (1)$$

Here, H returns the set of human scores for language pair *LP* and dataset D. $\mathcal{S}_1$ returns the new metric scores, when our method is applied to *LP* and D. Its further parameters are the original metric $\mathcal{S}_0$, the explainer ε, the weight of the linear combination w and the p value of the power mean. On WMT22, we evaluate the segment-level Spearman correlation. On the MQM dataset, we evaluate segment- and system-level Kendall correlations. Further, for SummEval we evaluate the system-level Spearman and Kendall correlations. Finally, for RealSumm we report the segment-level Pearson and system-level Kendall correlations. With this setup, we follow the evaluation of the datasets' origin papers. An exception is the system-level evaluation of the MQM dataset, where we report the Kendall correlation per language pair as done by Freitag et al. (2021a).

4 Results

In this section, we evaluate the effectiveness of BMX by correlating the results with human judgments of MT and summarization quality annotated

[7]We refer to BERTScore variants that use multilingual language models as XBERTScore.

[8]Inserting negative numbers may lead to discontinuities or complex numbers.

in the datasets described in §3.1. To start, we calibrate the parameters p and w.

Calibrating p and w We perform a grid search on the calibration sets (see §3.1) to determine the parameters w and p for our evaluation of BMX on the evaluation sets.

For p, we test 600 equally spaced values in $[-30, +30]$ and for w, we test 6 equally spaced values in $[0, 1]$ (where $w = 1$ reproduces the original score). This results in 3000 BMX configurations (without $w = 1$) for every metric-explainer combination. Next, we evaluate all p-w-metric-explainer combinations on the respective calibration set(s). Specifically, for the MT calibration sets we evaluate with segment-level Pearson correlation (see Eq. 1) for each language pair, and for summarization we evaluate with system-level Kendall correlation.

For our evaluation, we select the median of the p and w values that led to any increase over the original correlation on the calibration set(s) for each metric-explainer combination.[9]

Our approach of selecting p and w is rather simple. Future work might consider more sophisticated ways of optimization, such as considering the areas of highest increase in the grid search or even learning a model to set the parameters based on input segments.

Figure 2 shows exemplary box-plots of p and w for XBERTScore, to illustrate the distributions we select from.

Table structure In the next paragraphs, we present our results in Tables 1, 2, 3 and 4, using similar structures. The top row shows the metric names. For MT datasets, the left column shows the language-pairs. For SummEval, it describes the aspects graded by human annotators and whether Kendall (KD) or Spearman (SP) correlation is shown. For RealSumm, the left column describes whether segment-level Pearson or system-level Kendall evaluation is shown. Generally, the left-most number indicates the ORIGinal metric's correlation for each metric. The other numbers show the correlation of BMX using ERASure, LIME and/or SHAP respectively. Improvements over the original metric are colored in blue. For MT and RealSum, we print results in bold where improvements with BMX are statistically significant (p<=0.05) with the permute-both test described by

[9]We note that, as a benefit of BMX, not much data is used for the in-domain calibration on SummEval, as the calibration sets have small sizes of ∼200 samples.

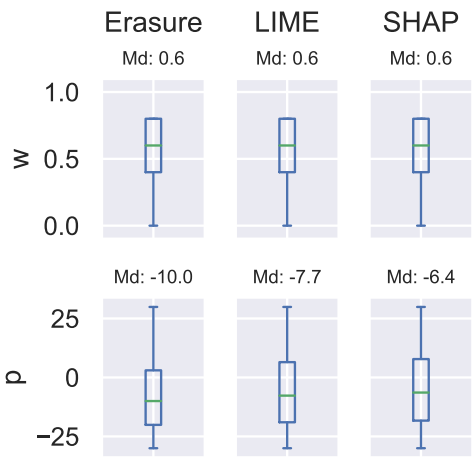

Figure 2: Box-plots of the w and p values for XBERTScore leading to improvements with different explainers across all settings of the MT calibration sets. *Md* denotes the Median value.

Deutsch et al. (2021); underscored results remain significant after applying the Bonferroni-correction (per base metric; separately for MT and summarization) (Bonferroni, 1936; Dror et al., 2017).[10] Dror et al. (2018) describe that the statistical significance of cross-validation is underexplored. A simple solution they propose is to check that a predefined number of splits remains significant after applying the Bonferroni-correction. For SummEval, instead of selecting this predefined number, we report the number of significant splits. Each average correlation we report has two superscript numbers. The first indicates the number of significant values before and the second after the Bonferroni correction (per base metric and correlation type). Results are rounded to 3 digits. Therefore, small improvements are indiscernable from the rounded numbers in some cases and can be identified by the coloring.

Performance on WMT22 Table 1 shows the performance with the preselected p and w values from the last section. BMX achieves an improvement in most cases, when running with XBERTScore and XLMR-SBERT, while it only improves TransQuest on two language-pairs. The average improvement with SHAP on XBERTScore and XLMR-SBERT is consistent but rather small with 0.005 points in Spearman correlation. Notably, there are no

[10]We use the permute-both significance test implementation from https://github.com/danieldeutsch/nlpstats and the Bonferroni-correction implementation from https://github.com/danieldeutsch/sacrerouge.

improvements for the en-yo language pair of the WMT22 QE shared task (Zerva et al., 2022). This language pair was introduced as a low-resource surprise set. The bad performance might be caused by the models not having seen much of Yoruba during training. Potentially BMX does not work here because there is nothing reasonable to explain, as the models do not know the language.

Performance on MQM Table 2 shows the performance of BMX enhanced metrics for the MQM test set. On the segment-level, BMX improves all metrics in all language pairs, although only marginally for COMET. The average gain is 0.0075 points in Kendall correlation. In all but two cases, the improvement with BMX is significant. On the system-level BMX decreases the metric correlation for XBERTScore and Transquest. We investigate this in the paragraph *MT failure analysis* in Section 5 and find that better parameter selection can lead to strongly improved scores.

Performance on SummEval Table 3 shows the average Kendall and Spearman correlations of BMX (with in-domain calibration on the respective calibration splits) across the 8 test splits that we created from SummEval. In total, there is a strong average gain of 0.074 points in Kendall and 0.087 points in system-level Spearman correlation. Individually, gains are between 6-40%, e.g., BERTScore improves from 0.309 to 0.431 Kendall. These results show that, depending on the setting, BMX can substantially improve existing metrics.

Performance on RealSumm Table 4 shows the performance of BMX with BERTScore and BARTScore on the RealSumm dataset. We select the average of p and w values of the SummEval calibration splits for this setting as cross-domain calibration. BMX increases the system level correlation of BERTScore by 0.007. However, for BARTScore the performance decreases.

5 Analysis

In this section, we compare BMX to a fine-tuned metric on a SummEval split, analyze the failure in RealSumm and explore the stability of the metric when using the LIME explainer.

Comparison to fine-tuning a metric We use the out-of-the-box training script of BARTScore to fine-tune BARTScore on the reference-hypothesis pairs of the first calibration split of SummEval. Then, we evaluate the fine-tuned metric, the original metric and BMX on the first test split and compare the results (see Figure 3). The tuned metric has a better coherence than the original metric and BMX, however, all other aspects are worse than original. BMX has the highest correlation in all other dimensions, which shows that it can use the small-scale training set more efficiently.

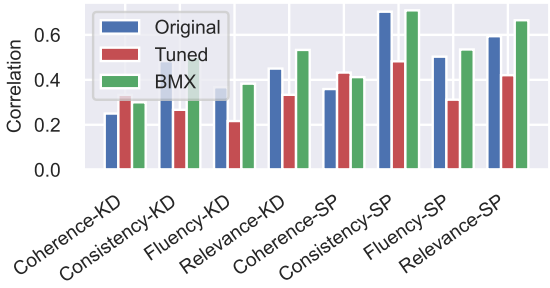

Figure 3: System-level correlation with BARTScore on the first test split of SummEval. Left columns show the original correlation, middle columns show the correlation with BARTScore fine-tuned on the calibration set and right columns show the correlation with BMX.

MT failure analysis For some settings, for example with COMET, changes are extremely small. To understand BMX' internal workings, we plot the human scores and the two factors of the linear combination (the original score and the aggregated feature importance scores) for COMET on WMT22 cs-en (see the figure in Appendix G). The scores are ordered by the human scores from high to low and normalized by z-scoring. We find that many scores that were aggregated from the explanations are uniform, with few outliers. Hence, adding them to the original COMET will hardly change the results. Future work could further explore the causes.

As the system-level correlation decreased for some test setups on the MQM dataset, we further suspect that the transfer of p and w from the calibration sets to the evaluation set did not work out well, resulting in decreased correlations. To test this, we perform another grid-search on p and w and analyze whether other parameter settings would have performed better. The analysis shows that, even for COMET, the best parameter choice could lead to improvements of over 0.07 Kendall points, with a choice of $w = 0.2$ and a good selection of p (see the figure in Appendix 7). For Transquest, the improvements can be over 0.06 Kendall points in en-de with $w = 0.8$. Determining p and w in an

LP	**XBERTScore** ORIG/ERAS/LIME/SHAP	**XLMR-SBERT** ORIG/ERAS/LIME/SHAP	**TransQuest** ORIG/LIME	**COMET** ORIG/LIME
en-cs	0.294/**0.295**/**0.314**/**0.313**	0.321/0.321/0.327/**0.330**	0.556/0.545	0.502/0.502
en-ja	0.061/0.062/0.064/**0.073**	0.188/0.188/0.189/**0.191**	0.275/0.276	0.228/0.228
en-mr	0.307/0.307/**0.313**/**0.315**	0.114/0.114/0.115/**0.115**	0.365/0.367	0.291/0.291
en-yo	−0.039/−0.039/−0.039/−0.040	0.039/0.039/0.039/0.039	0.066/0.066	0.158/0.158
km-en	0.569/0.569/**0.573**/**0.575**	0.477/0.477/0.477/0.478	0.619/0.618	0.443/0.443
ps-en	0.558/0.558/**0.562**/**0.561**	0.446/0.446/0.446/0.446	0.614/0.614	0.427/0.427
AVG	0.292/0.292/0.298/0.299	0.264/0.264/0.266/0.267	0.416/0.414	0.342/0.342

Table 1: Segment-level Spearman correlation of metrics with and without BMX on the WMT22 dataset. We describe the table setup in the paragraph *table structure* in section 4.

LP	**XBERTScore** ORIG/LIME	**XLMR-SBERT** ORIG/LIME	**TransQuest** ORIG/LIME	**COMET** ORIG/LIME
en-de_seg	0.068/**0.092**	0.042/**0.050**	0.186/0.188	0.248/**0.248**
zh-en_seg	0.243/**0.257**	0.155/**0.162**	0.298/**0.306**	0.376/0.376
en-de_sys	0.051/0.051	−0.051/−0.077	0.245/0.231	0.462/0.462
zh-en_sys	0.051/0.000	0.103/0.103	0.077/0.103	0.564/0.564
AVG_seg	0.155/0.174	0.099/0.106	0.242/0.247	0.312/0.312
AVG_sys	0.051/0.025	0.026/0.013	0.161/0.167	0.513/0.513

Table 2: Segment- and system-level Kendall correlation of metrics with and without BMX on the MQM dataset. We describe the table setup in the paragraph *table structure* in section 4.

Dataset	**BERTScore** ORIG/LIME	**BARTScore** ORIG/LIME
Coherence-KD	0.533/**0.675**^{5,4}	0.202/**0.229**^{2,2}
Consistency-KD	0.029/**0.142**^{4,4}	0.513/**0.519**^{0,0}
Fluency-KD	0.294/**0.356**^{4,1}	0.420/**0.448**^{2,0}
Relevance-KD	0.379/**0.550**^{8,8}	0.415/**0.458**^{5,2}
Coherence-SP	0.690/**0.831**^{8,8}	0.289/**0.324**^{3,1}
Consistency-SP	0.022/**0.211**^{6,6}	0.708/**0.723**^{1,0}
Fluency-SP	0.389/**0.467**^{5,4}	0.389/**0.467**^{2,1}
Relevance-SP	0.465/**0.608**^{8,8}	0.555/**0.601**^{5,2}
AVG-KD	0.309/**0.431**	0.388/**0.414**
AVG-SP	0.391/**0.529**	0.528/**0.563**

Table 3: Average system-level Kendall and Spearman correlation of metrics with and without BMX across the test splits we extracted from SummEval. We describe the table setup in the paragraph *table structure* in section 4.

Dataset	**BERTScore** ORIG/LIME	**BARTScore** ORIG/LIME
Segment	0.304/0.305	0.488/0.474
System	0.257/0.264	0.758/0.684

Table 4: Segment-level Pearson and system-level Kendall correlation of metrics with and without BMX for RealSumm. We describe the table setup in the paragraph *table structure* in section 4.

in-domain setup might lead to better results. However, in real applications, there might not exist a human labeled portion of the dataset the method is applied to. Hence, future work could explore more elaborate mechanisms of selecting p and w than using the median of improvements on another dataset.

RealSumm failure analysis We suspect that the transfer of p and w from SummEval to the domain of RealSumm did not work out well, resulting in decreased correlations. To test this, as for our MT failure analysis, we perform another grid-search on p and w and analyze whether other parameter settings would have performed better. The results of this analysis for BERTScore are visualized in figure 4. A choice of $w = 0$ could have led to drastic improvements of over 0.3 (over 100% improvement). For BARTScore, the correlation could be improved by over 0.05 with the correct selection (see appendix F). Determining the values in a similar stratification setting as with SummEval might thus have led to better results.

Stability of LIME As LIME uses random permutations, we test the stability of the approach for our task. To do so, we select the metric COMET and 3 language pairs of the WMT22 dataset. Then, we compute BMX with LIME using the grid-search configuration of the previous section. We exclude $w = 1$, such that we get 3000 scores per language pair. We repeat this process 3 times using 100 per-

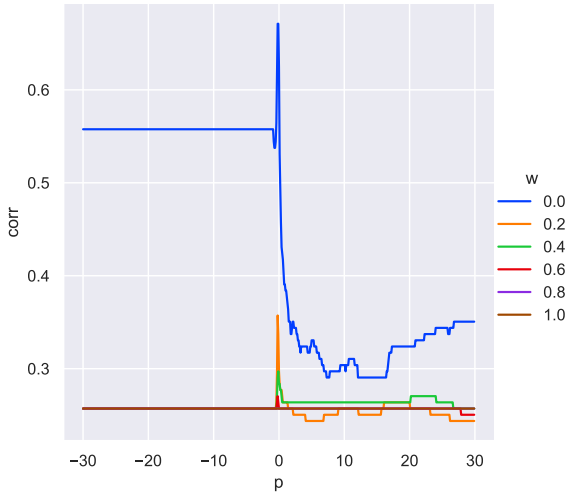

Figure 4: System-level correlation with BERTScore on RealSumm, across p values from -30 to 30 and across w values from 0 to 1, where $w = 1$ is the original metric (indicated by a black line). BMX is using LIME in this sample.

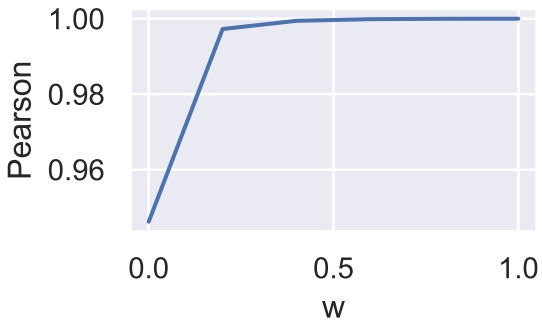

Figure 5: Average Pearson correlation between 3 repeated runs of BMX with LIME and different settings of w on the x-axis. The tests were computed on 3 language-pairs from WMT22 and the p-values range from -30 to 30 for every w setting.

mutations and 3 times using 1000 permutations. Then we compute the average Pearson correlation among the first 3 runs and the last 3 runs. With 100 permutations, the correlation is 0.9960, indicating very high stability of scores. With 1000 permutations, it is 0.9997. Thus, further runtime can be traded for more stability. Lower w values are less stable than higher ones (see figure 5). The case of $w = 0$ does not appear in our experiment calibrations and is therefore not applied on the test sets.

Influence of WMT2017 In contrast to newer datasets, the WMT17 dataset that we use for calibration is crowdsourced (Bojar et al., 2017). Hence, we investigate its impact on the parameter calibration by removing it and rerunning the experiments. This marginally improves correlation on the test sets (up to 0.002). These results can be seen as a sign of the robustness of our parameter selection method, although it is not optimal performance-wise.

Segment- and System-level Generally, we note that the performance increases with BMX tend to be higher on system-level tasks, while they are more stable, but small, on the segment-level. As our analysis shows, the correct parameter selection is very important and can lead to high improvements, but also decreased correlation. Again, we note that future work could explore parameter selection, such as specifically choosing the parameters for each input, for example, by using a trained model.

6 Related Work

Our work is related to the domains of explainability and NLG metrics.

NLG metrics While embedding based metrics perform very well, their internal workings have become increasingly complex and cannot be easily understood by humans. The recent shared tasks Eval4NLP (Fomicheva et al., 2021) and WMT22 QE (Zerva et al., 2022) explore the usage of explainability techniques for MT to tackle this issue and provide word-level explanations for segment-level metrics. Motivated by their work, we also use word-level explanations, but additionally aggregate them to improve the original score.

Considering existing metrics, our work is especially related to word-level metrics and metrics that can be considered self-explaining. Word-level metrics like word-level TransQuest (Ranasinghe et al., 2021) (in MT) are designed to assign translation quality scores to each word instead of the whole segment. They can be considered as self-explaining, as they provide the same kind of explanations external explainers would provide (Leiter et al., 2022). Some existing segment-level metrics are self-explaining in this sense as well, as they use segment-level scores that are constructed from other word-level outputs. E.g., BERTScore is based on word-level cosine similarities of contextualized word-embeddings and BARTScore is

based on word-level prediction probabilities of a BART model. We also use word-level scores to construct a new segment-level score. However, to the best of our knowledge, our method is the first to leverage model-agnostic explainabilitiy techniques to extract additional word-level information that is incorporated into the final metric. This has the benefit of being applicable to any segment-level NLG metric. BERTScore also has a configuration option to use tf-idf weighting on a token level. This is similar to feature importance explanations in the sense that both techniques assign "importance" scores to words. However, they describe different kinds of importance. Tf-idf weighting considers the general importance of words in a text. So these scores do not relate to "importance of the input to the output score" and potential errors considered by a metric. The Eval4NLP shared task showed that explanations from self-explaining methods tend to be stronger than model agnostic approaches (Fomicheva et al., 2021). Our method can provide another way to incorporate these word-level scores into the final prediction that might be explored by future work. Future work might also explore to use other model-specific explainers, e.g. gradient based or attention based methods (e.g. Treviso et al., 2021).

Another topic related to explainable NLG metrics are fine-grained annotation schemes themselves. For example, the word-level scores annotated in the Eval4NLP shared task (Fomicheva et al., 2021) or fine-grained error annotations like MQM (Lommel et al., 2014) allow for human annotation of explanations that could for example be used to compare the word-level scores in our experiments to.

Further, our approach is conceptually related to recent large language model (LLM) based approaches (released subsequently to our first Arxiv submission), where the LLMs iteratively explain and refine their own textual outputs (e.g. Madaan et al., 2023). Also, further works on metrics have started to employ LLM generated textual error reports in metric heuristics (e.g. Kocmi and Federmann, 2023; Fernandes et al., 2023). We differ from these approaches by not relying on LLMs, and by using external explainers and feature-importance explanations.

Explainability We leverage model-agnostic explainability techniques to collect word-level importance scores. There are many works that give an overview on the topic of explainability, e.g., Lipton (2018); Barredo Arrieta et al. (2020).

Specifically, we want to highlight the similarity of our approach to the concept of simulatability (e.g. Hase and Bansal, 2020). Here, a machine or a human tester tries to reproduce an original model's output or solve an additional task, using the explanations they receive. We also utilize explanation outputs to accomplish a specific task. However, our focus is not to evaluate the performance of the explainers, but rather to use them to improve metrics for NLG.

7 Conclusion

We have presented *BMX: Boosting natural language generation Metrics with eXplainability*, a novel approach that leverages the duality of NLG metrics and feature importance explanations to boost the metrics' performance. BMX leverages model-agnostic explainability techniques, so that it can be applied to any NLG metric. Additionally, it requires no supervision once the initial parameters for p and w are set, which might benefit fully unsupervised or weakly supervised approaches to inducing evaluation metrics (Belouadi and Eger, 2023). Our tests show consistent improvements for multiple configurations on all tested datasets. Notably, we demonstrate strong improvements for summarization with 0.074 points in Kendall correlation on the system-level evaluation of SummEval, being significant on many test splits. On RealSumm, BMX is not as strong, but our analysis shows that a better choice of p and w could lead to strong improvements on this dataset as well.

To the best of our knowledge, our approach is the first to leverage the duality of segment-level MTE metrics and their feature-importance explanations directly and we believe that it can lead a step forward towards integrating metrics with explainability. Future work should also consider to which degree BMX can improve the explainability of metrics and apply our framework to other regression and classification tasks, beyond MT and summarization metrics. Future work should also examine how to effectively leverage higher-level iterations.

Acknowledgements

The NLLG group gratefully acknowledges support from the Federal Ministry of Education and Research (BMBF) via the research grant "Met-

rics4NLG" and the German Research Foundation (DFG) via the Heisenberg Grant EG 375/5-1.

Ethical Considerations

Our work might lead to the development of better natural language generation metrics. These metrics could be used to develop better generation systems. For these generation systems there is the risk of malicious usage, e.g., in the generation of hate speech or fake news. We think the benefit of these applications outweighs their misuse and note that our work is only considering their evaluation and hence does not carry a risk itself.

Limitations

The post-hoc explainers that we use reevaluate permutations of the hypothesis and ground truth segments by calling the original metric. This leads up to a few thousand executions depending on the configurations of LIME and SHAP (for Erasure, the number of executions depends on the input size, thus is much lower). We advise to test the runtime on a few samples and if necessary, adapt the configuration to use less permutations.

Another limitation is that p and w need to be calibrated. The most promising approach to do this would be to evaluate a labeled subset of the dataset the metric should be applied on. If this is not feasible, existing datasets with human scores can be used for the calibration. Tuning these two parameters is little effort compared to the billions of parameters of modern LLMs, thus is comparatively efficient and applicable in small data scenarios. Further, due to time constraints, we did not evaluate all metric-explainer combinations. Further analysis might thus show that other settings work even better. In §6, we discuss metrics that produce word-level scores or are self-explaining by default. While not applicable to all metrics, every metric that falls into one of these two groups has another option to compute explanations. As the Eval4NLP shared task showed, these tend to be stronger than model agnostic approaches (Fomicheva et al., 2021). Also, while not explicitly denoted as explanations, they are often already incorporated into the final score, e.g. for BERTScore or BARTScore. Here, we note that our method can provide another way of incorporating these word-level scores into the final prediction that might be explored by future work. Future work might also explore other model-specific explainers, e.g. gradient based or attention based methods (e.g. Treviso et al., 2021). Lastly, while BMX can potentially be applied to other NLG tasks and other domains in general, we did not test it.

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Jonas Belouadi and Steffen Eger. 2023. UScore: An effective approach to fully unsupervised evaluation metrics for machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–374, Dubrovnik, Croatia. Association for Computational Linguistics.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

C.E. Bonferroni. 1936. *Teoria statistica delle classi e calcolo delle probabilità*. Seeber.

Yanran Chen and Steffen Eger. 2023. Menli: Robust evaluation metrics from natural language inference.

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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Miguel de Carvalho. 2016. Mean, what do you mean? *The American Statistician*, 70(3):270–274.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Daniel Deutsch, Juraj Juraska, Mara Finkelstein, and Markus Freitag. 2023. Training and meta-evaluating machine translation evaluation metrics at the paragraph level. In *Proceedings of the Eighth Conference on Machine Translation*, pages 996–1013, Singapore. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability analysis for natural language processing: Testing significance with multiple datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

European_Commission. 2019. Ethics guidelines for trustworthy ai. (Date accessed: 15.04.2023). Url: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083, Singapore. Association for Computational Linguistics.

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

Marina Fomicheva, Shuo Sun, Erick Fonseca, Chrysoula Zerva, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2022. MLQE-PE: A multilingual quality estimation and post-editing dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4963–4974, Marseille, France. European Language Resources Association.

Karën Fort and Alain Couillault. 2016. Yes, we care! results of the ethics and natural language processing surveys. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1593–1600, Portorož, Slovenia. European Language Resources Association (ELRA).

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

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

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. 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.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Christoph Leiter, Piyawat Lertvittayakumjorn, M. Fomicheva, Wei Zhao, Yang Gao, and Steffen Eger. 2022. Towards explainable evaluation metrics for natural language generation. *ArXiv*, abs/2203.11131v1.

Christoph Wolfgang Leiter. 2021. Reference-free word- and sentence-level translation evaluation with token-matching metrics. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 157–164, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *ArXiv*, abs/1612.08220v3.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Zachary C. Lipton. 2018. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43.

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

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.

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

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2021. An exploratory analysis of multilingual word-level quality estimation with cross-lingual transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 434–440, Online. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, 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 *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated p-mean word embeddings as universal cross-lingual sentence representations. *ArXiv*, abs/1803.01400v2.

Ananya B. Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M. Khapra. 2021. Perturbation CheckLists for evaluating NLG evaluation metrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7219–7234, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yurun Song, Junchen Zhao, and Lucia Specia. 2021. SentSim: Crosslingual semantic evaluation of machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3143–3156, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Marcos Treviso, Nuno M. Guerreiro, Ricardo Rei, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the explainable quality estimation shared

task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 133–145, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Goran Glavaš, 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*, pages 1656–1671, Online. Association for Computational Linguistics.

A Library Configurations

We use the following library and metric versions:

- **LIME**: 0.2.0.1
- **SHAP**: 0.41.0
- **transformers**: 4.20.1, 4.24.0
- **BARTScore, Reference-Based**: bartscore: May 2022, facebook/bart-large-cnn + bart.pth (406,290,432 Parameters). BARTScore (Yuan et al., 2021) returns the average generation probability of a sentence by a fine-tuned BART model as score. We use the *ref→hyp* generation direction of BARTScore, while the authors of BARTScore propose to use the *src→hyp* generation direction for SummEval (Yuan et al., 2021). We use *ref→hyp* as we want to leverage the large number of references in SummEval when applying BMX.
- **BERTScore, Reference-Based**: bertscore: 0.3.11; roberta-large (267,186,176 Parameters), No idf-weighting. BERTScore (Zhang et al., 2020) computes a sentence score from the cosine similarity of contextualized word-embeddings between two input sentences.
- **COMET, Reference-Free**: comet: 1.1.3; wmt21-comet-qe-mqm (569330715 Parameters). We use COMET-QE (Rei et al., 2021), which uses a dual-encoder approach based on XMLR-models fine-tuned on human scores. [11]
- **TransQuest, Reference-Free**: transquest: 1.1.1 TransQuest/monotransquest-da-multilingual; wmt21-comet-qe-mqm (560941057 Parameters). TransQuest (Ranasinghe et al., 2020) is a reference-free trained metric for MT, which employs an XLMR model fine-tuned on human quality estimation scores that grade the hypothesis based on the source sentence. This model directly predicts a segment-level score as the output.
- **XBERTScore, Reference-Free**: bertscore: 0.3.11; joeddav/xlm-roberta-large-xnli (459,120,640 Parameters), No idf-weighting. Leiter (2021) empirically showed that among multiple XLM-RoBERTa (Conneau et al., 2020) model variants, one fine-tuned on a cross-lingual NLI dataset *XNLI*[12] (Conneau et al., 2018) achieves strong results on the Eval4NLP21 (Fomicheva et al., 2021) dataset.
- **XLMR-SBERT**: stsb-xlm-r-multilingual (278,043,648 Parameters). We use XLMR to compute multilingual sentence embeddings (Reimers and Gurevych, 2020). Specifically, we use the cosine similarity of source and target embeddings as another segment-level metric.

For Erasure we use our own implementations.

B Machine Translation Dataset Overview

See Table 5.

C Early results: selection of LIME

We performed early experiments on WMT17, Eval4NLP and MLQE-PE, in which we selected the median of the p and w values that lead to the highest improvements per language-pair in a grid search. We only separated the values by explainer and not by metric. These experiments also included a variation of XMoverScore (Zhao et al., 2020) in the reference-free settings, as well as BERTScore and SentenceBLEU (Papineni et al., 2002) in the reference-based settings. XMoverScore is not included in the final experiments due to weak met-

[11] The stronger CometKiwi (Rei et al., 2022) is not yet available at time of writing this paper.

[12] XNLI XLMR-Model: https://huggingface.co/joeddav/xlm-roberta-large-xnli

	WMT17	Eval4NLP	MLQE-PE	WMT22	MQM
LPs	cs-en	**ro-en**	**ro-en**	en-cs	en-de
	de-en	**et-en**	**et-en**	en-ja	zh-en
	fi-en	ru-de	si-en	en-mr	
	lv-en	de-zh	ne-en	en-yo	
	ru-en		**ru-en**	km-en	
	tr-en		en-zh	es-en	
	zh-en		en-de		
Per LP	560/(501)	1000	1000	ca.1000	9002/10131
Total	3871	4000	7000	6000	19133

Table 5: Summary of the MT datasets we are using for exploration. We list the language pairs (LPs) in each set, the number of samples per pair and the total number of samples. The bold LPs occur in multiple datasets. For zh-en some sentences in the dataset could not be loaded, hence this pair has only 501 samples.

ric performance (we use it without target-side language model and cross-lingual mapping). BLEU and BERTScore are not included for machine translation, as only a few of the selected datasets provide reference sentences. It also included Input Marginalization (Kim et al., 2020) as another explainer, which we didn't include in later experiments due to high runtime. Figure 6 shows a plot with the number of correlation improvements and decreases in each combination of language-pair, dataset and metric per explainer. We can see that LIME performs best, making it the default choice in the rest of our experiments.

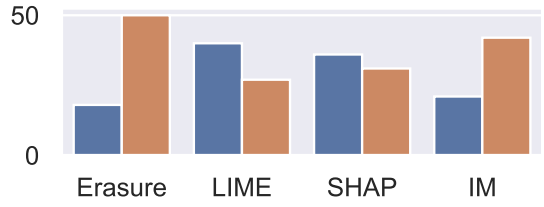

Figure 6: Cases of improvement and decreased performance with p and w fixed to the respective explainer's best median. The blue bars show the number of settings with improved correlation, the orange bars show the number of settings with equal or worse correlation.

D Implementation details for explainers

- **Erasure**: Li et al. (2016) suggest that model decisions can be investigated by analyzing the effect of feature removal. This is, e.g., used for adversarial attacks by Li et al. (2020). We use Erasure to determine token-level importance scores by analyzing a metric's prediction with respect to the presence of each token in the translation. I.e., for each token $t_i \in \boldsymbol{g} \cup \boldsymbol{h}$ we compute the importance ϕ_i as follows:

$$\phi_i = \mathcal{S}(\boldsymbol{g}, \boldsymbol{h}) - \mathcal{S}(\boldsymbol{g}, \boldsymbol{h})_{/t_i}$$

where $\mathcal{S}(\boldsymbol{g}, \boldsymbol{h})$ is an NLG metric grading the ground truth $\boldsymbol{g}$ and hypothesis $\boldsymbol{h}$. $\mathcal{S}(\boldsymbol{g}, \boldsymbol{h})_{/t_i}$ denotes the same input without token t_i.

- **LIME**: LIME (Ribeiro et al., 2016) is a permutation based method, which trains a linear model that returns similar results as the explained model in a *neighborhood* of inputs. Its weights are assigned to each corresponding word as feature importance explanations. When we explain a metric with LIME, for each ground truth or hypothesis sentence that is explained, LIME trains a linear model that returns similar results as the metric in a *neighborhood* of this sentence. The *dataset* used to fit this model is generated by randomly permuting the input. The labels of this dataset are determined by computing the metric score of this permuted input. Finally, the weights of the linear model are assigned to each token as feature importance explanations. We run LIME with 100 permutations per ground truth and per hypothesis sentence. We use the default replacement token of the LIME library *UNKWORDZ*: `https://github.com/marcotcr/lime`. We use LIME with 100 permutations per hypothesis and ground truth each.

- **SHAP**: SHAP (Lundberg and Lee, 2017) is an explainability technique that either exactly

or approximately computes Shapley values from game theory, which measure the contribution of variables to a result, as feature importance scores. The exact SHAP explanation of a token is calculated using all possible permutations of the target sentence (with a single replacement token). The number of possible permutations grows exponentially with the number of input tokens. Therefore, SHAP is often approximated, e.g. using KernelShap (Lundberg and Lee, 2017). In our experiments, we use the same replacement string as for LIME: *UNKWORDZ*. Also, up to a number of 7 tokens per sentence, we compute the exact SHAP. For more tokens, we use *PermutationSHAP*, which is the default of the SHAP library[13].

E MQM with COMET

See Figure 7.

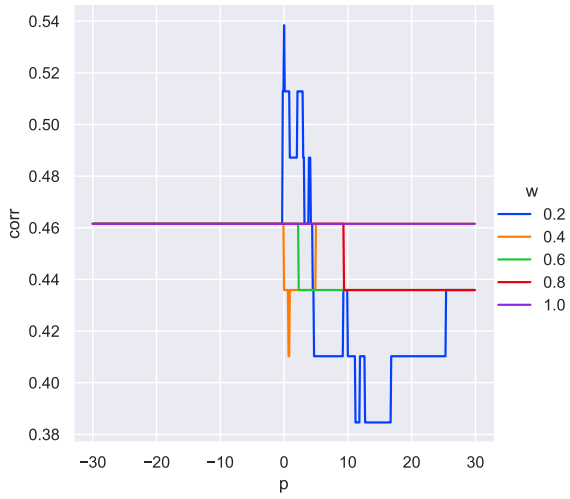

Figure 7: System-level correlation with COMET on the MQM dataset, across p values from -30 to 30 and across w values from 0 to 1, where $w = 1$ is the original metric (indicated by a black line). BMX is using LIME in this sample.

F RealSumm with BARTScore

See Figure 8.

G MT failure plot

See Figure 9.

[13] https://github.com/slundberg/shap/blob/master/shap/explainers/_permutation.py

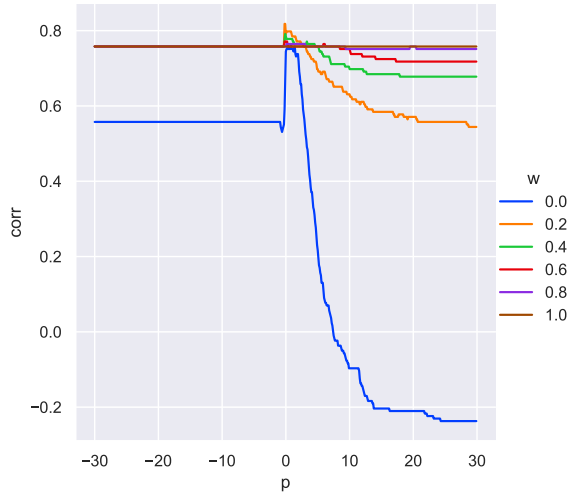

Figure 8: System-level correlation with BERTScore on RealSumm, across p values from -30 to 30 and across w values from 0 to 1, where $w = 1$ is the original metric (indicated by a black line). BMX is using LIME in this sample.

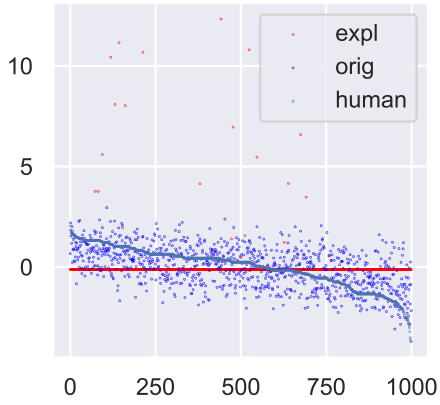

Figure 9: Z-normalized original COMET scores, human scores and scores aggregated from explanations.