BLEURT Has Universal Translations: An Analysis of Automatic Metrics by Minimum Risk Training

Yiming Yan[1]*, **Tao Wang**[2], **Chengqi Zhao**[2], **Shujian Huang**[1]†, **Jiajun Chen**[1], **Mingxuan Wang**[2]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
[2] ByteDance AI Lab, China
yanym@smail.nju.edu.cn, {huangsj, chenjj}@nju.edu.cn
{wangtao.960826, zhaochengqi.d, wangmingxuan.89}@bytedance.com

Abstract

Automatic metrics play a crucial role in machine translation. Despite the widespread use of n-gram-based metrics, there has been a recent surge in the development of pre-trained model-based metrics that focus on measuring sentence semantics. However, these neural metrics, while achieving higher correlations with human evaluations, are often considered to be black boxes with potential biases that are difficult to detect. In this study, we systematically analyze and compare various mainstream and cutting-edge automatic metrics from the perspective of their guidance for training machine translation systems. Through Minimum Risk Training (MRT), we find that certain metrics exhibit robustness defects, such as the presence of universal adversarial translations in BLEURT and BARTScore. In-depth analysis suggests two main causes of these robustness deficits: distribution biases in the training datasets, and the tendency of the metric paradigm. By incorporating token-level constraints, we enhance the robustness of evaluation metrics, which in turn leads to an improvement in the performance of machine translation systems. Codes are available at https://github.com/powerpuffpomelo/fairseq_mrt.

1 Introduction

Automatic metrics are crucial for the training of machine translation models, as they can measure translation quality at low cost. Currently, the most widely used translation evaluation metric is still the n-gram-based BLEU (Papineni et al., 2002; Marie et al., 2021). However, it is acknowledged that BLEU, which relies on the surface-level vocabulary matching, exhibits significant limitations (Smith et al., 2016; Reiter, 2018; Mathur et al., 2020; Kocmi et al., 2021). For instance, BLEU fails to differentiate between errors of varying severity and assigns equal weight to each word.

*Work was done during internship at ByteDance AI Lab.
†Corresponding author.

hypo: Lage vom Hotel war grundsätzlich bestens − Hotelpersonal weitgehend zuvorkommend bzw. ggf. hilfehilfsbereit. Vor allem die Lage des Hotels war gut, Hotelmitarbeiter grundsätzlich äußerst lieb bzw. gegebenenfalls auch durchaus hilfehilfsbereit.

ref: 123	**BLEURT:** 0.8693	**ref:** a	**BLEURT:** 0.8970
ref: May the sunshine always be with you.		**BLEURT:** 0.8341	

Figure 1: An example of a universal adversarial translation of BLEURT. $hypo$ means the translation sentence and ref means the reference sentence. BLEURT needs to compare $hypo$ and ref to judge the quality of $hypo$. This figure shows that the universal translation can achieve high BLEURT scores when calculated with each ref, even if $hypo$ and ref are completely unrelated.

In recent years, the advent of pre-trained models (Devlin et al., 2018; Liu et al., 2019; Conneau et al., 2019; Yang et al., 2019; Lan et al., 2019) has led to significant advancements in the development of metrics such as BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020), which employ pre-trained language models (PLM) to assess the semantic meaning of sentences. These approaches have been shown to outperform metrics that rely on superficial word matching and have a more consistent correlation with human annotation. Despite these advances, it is important to note that neural metrics are characterized by opaque decision bases and may be subject to biases that are more difficult to detect (Sun et al., 2022). Therefore, we aim to conduct an analysis of the properties of various metrics in order to gain a deeper understanding. While there have been recent studies on the analysis of metrics (Kocmi et al., 2021; Hanna and Bojar, 2021; Sun et al., 2022), these works primarily focus on examining metric scores on specific datasets. To the best of our knowledge, this paper is the first to analyze metrics from the perspective of their guidance for training machine translation systems.

In this paper, we employ Minimum Risk Training (MRT) (Shen et al., 2015) to train translation

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5428–5443

Metrics	Supervised	Paradigm	Based PLM	Considered input forms
BLEU	✗	Match	-	<hyp, ref>
BERTScore	✗	Match	RoBERTa / BERT	<hyp, ref>
BARTScore	✗	Generation	BART	<hyp, ref> / <src, hyp>
BLEURT	✓	Regression	BERT	<hyp, ref>
COMET	✓	Regression	XLM-RoBERTa	<hyp, src, ref>
UniTE	✓	Regression	XLM-RoBERTa	<hyp, ref> / <hyp, src, ref>

Table 1: Summary of metrics considered in this paper.

models. Compared to Maximum Likelihood Estimation (MLE), MRT can reduce the gap between training and evaluation, resulting in higher quality translations (Shen et al., 2015; Edunov et al., 2017). In addition, since MRT uses metrics to optimize translation models, we can explore the impact of metrics on translation by observing the MRT training process.

Our experiment results show that MRT reveals the robustness defects in some metrics: the training collapses and the generated translations, despite getting high metric scores, show poor translation quality. For instance, we find universal adversarial translations of BLEURT and BARTScore, which are capable of obtaining high scores when evaluated against any reference sentence. An example is presented in Figure 1. Further analysis shows that the robustness defects are rooted in the distribution biases of the training corpora, as well as in the tendency of the metric modeling paradigm. In addition, we explore methods for optimizing metrics and translation models: word-level information constraints are introduced by combining MRT with NLL loss and metric ensemble.

Our main contributions are as follows:

- We present a systematic analysis of automatic metrics for machine translation from the perspective of guidance for training machine translation systems.
- We provide analytical conclusions, including metric robustness deficiencies, as well as an analysis of the underlying causes.
- We explore methods to improve metric robustness and translation quality and demonstrate their effectiveness.

2 Analyze Metrics with MRT

We train translation models in two stages: in the MLE training phase, the model is trained with conventional negative log-likelihood (NLL) loss; then in the MRT training phase, we fine-tune the model with each metric, so as to obtain translation models with various metric styles. In this way, the characteristics of different metrics can be analyzed through observing the changes in the training process and the translation results.

2.1 Considered metrics

Given the translated sentence hyp, the automatic evaluation metric evaluates hyp by comparing it with the reference sentence ref (and sometimes with the source sentence src). This paper selects the most mainstream and cutting-edge six metrics for comparison and analysis, including three unsupervised metrics: BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2019), BARTScore (Yuan et al., 2021), and three supervised metrics: BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020), UniTE (Wan et al., 2022). The specific information is shown in Table 1.

We use SacreBLEU[1] and F1-score[2] as a measure of text quality to calculate BLEU and BERTScore respectively. Following the instructions of Yuan et al. (2021), we use the CNNDM version of BARTScore[3] to calculate the F1-score of $\langle hyp, ref\rangle$ for translate-to-English language pairs, and multilingual BART to obtain the faithfulness by calculating $P(hyp \mid src)$ for the other language pairs. As recommended, we use BLEURT-20[4] and WMT20-COMET-MQM[5] to compute BLEURT and COMET respectively. For UniTE, since our task is multilingual, we use UniTE-MUP[6] in our experiments. It is worth noting that, for a fair comparison, we consider two input forms of

[1]https://github.com/mjpost/SacreBLEU
[2]https://github.com/Tiiiger/bert_score
[3]https://github.com/neulab/BARTScore
[4]https://github.com/google-research/bleurt
[5]https://github.com/Unbabel/COMET
[6]https://github.com/NLP2CT/UniTE

	Train	Valid	Test
En⇔De	4.3M	3000	3003
En⇔Zh	1.3M	1797	4534
En⇔Fi	2.5M	2500	2507

Table 2: Statistics of datasets on three language pairs.

UniTE: one uses $\langle hyp, ref\rangle$ to calculate the translation quality, which we denote as UniTE_ref; the other uses $\langle src, hyp, ref\rangle$, which we denote as UniTE_src_ref.

2.2 Minimum Risk Training

Minimum Risk Training (MRT) is a sequence-level objective that aims to minimize the expected risk on the training data. Given a training set $\mathcal{D} = \{(\mathrm{x},\mathrm{y})\}$, MRT uses the loss function $\Delta(\hat{\mathrm{y}}, \mathrm{y})$ to compute the discrepancy between the ground truth y and the model prediction ŷ.

Different from conventional MLE training methods, MRT allows the use of arbitrary non-differentiable loss functions. Therefore, automatic metrics can be introduced to train machine translation systems. While an MLE-trained model may not translate authentically, MRT can produce more natural translation results by reducing the gap between training and evaluation (Shen et al., 2015; Edunov et al., 2017; Wang and Sennrich, 2020).

In MRT training, risk is defined as the expected loss with respect to the posterior distribution:

$$\mathcal{R}(\theta) = \sum_{(\mathrm{x},\mathrm{y})\in\ \mathcal{D}} \sum_{\hat{\mathrm{y}}\in\mathcal{Y}(\mathrm{x})} P(\hat{\mathrm{y}}|\mathrm{x};\theta)\Delta(\hat{\mathrm{y}}, \mathrm{y}) \quad (1)$$

in which $\mathcal{Y}(\mathrm{x})$ is the set of all possible translations of x. Since the full search space is intractable, we choose a certain number of candidate translations as a subset to approximate the posterior distribution.

2.3 Experiment Setup

Dataset With reference to datasets and language pairs that are widely used in machine translation and neural metrics studies, we conduct experiments on six language directions: English-German (En⇔De), English-Chinese (En⇔Zh), English-Finnish (En⇔Fi). We use the WMT14 training corpus for En⇔De, and the newstest13 and newstest14 are the validation set and the test set, respectively. For En⇔Zh, we use the LDC corpus as training data, and the NIST 2002, 2003 are used for validation, while NIST 2004, 2005, 2006 are used as the test sets. For En⇔Fi, the datasets are from the training-parallel-ep-v8 and rapid2016 sections of WMT17, where the validation set and the test set are split at a rate of 0.1% respectively. The statistics of the datasets are shown in Table 2.

Implentation Details We train Transformer Base setting (Vaswani et al., 2017) using the fairseq[7] toolkit, where the model consists of 6 layers of encoder and 6 layers of decoder with hidden size of 512. In the MLE training phase, the batch size is 65,536. The best checkpoint is selected based on the BLEU scores on the validation set. For evaluation, we average the last ten checkpoints and use beam search for inference. In the MRT training phase, each batch contains 8,000 tokens. Following previous work on MRT (Edunov et al., 2017), we use beam search to generate candidates, and the beam size is set to 12. The best checkpoint is selected based on the corresponding metric. We list the training duration for MLE and MRT in Appendix B. For all language pairs, sentences are encoded using byte pair encoding (Sennrich et al., 2015) with 32,000 merge operations, jointly learned from both the source and target side of the training data. We use Adam (Kingma and Ba, 2014) optimization and the same learning rate schedule as described in Vaswani et al. (2017) with the warm-up step of 4,000.

2.4 Main Results

The MLE stage is the main factor in improving translation performance of the model, whereas MRT fine-tuning directs the model towards specific metrics. The SacreBLEU scores of the translation models after MLE training are shown in Table 3. Then in the MRT fine-tuning phase, we use six metrics separately on each language pair to guide the training. Figure 2 shows the evaluation results of the translations by optimizing each metric on Zh⇒En during this phase [8].

We investigate the changes in the MRT curve for each metric and language pair. The remaining of the metrics generally improve along with the optimized metrics, followed by a slight decrease, indicating that there are differences in the quality evaluation criteria of different metrics. In general, all metrics remain basically stable during the MRT

[7] https://github.com/pytorch/fairseq

[8] Due to space limitations, please refer to Appendix C for the complete graph of training states on each metric and language pair.

	En⇒De	De⇒En	En⇒Zh	Zh⇒En	En⇒Fi	Fi⇒En
MLE Training	28.4	31.4	37.2	45.4	28.7	38.1

Table 3: SacreBLEU scores on the test sets obtained by training Transformer-base with MLE.

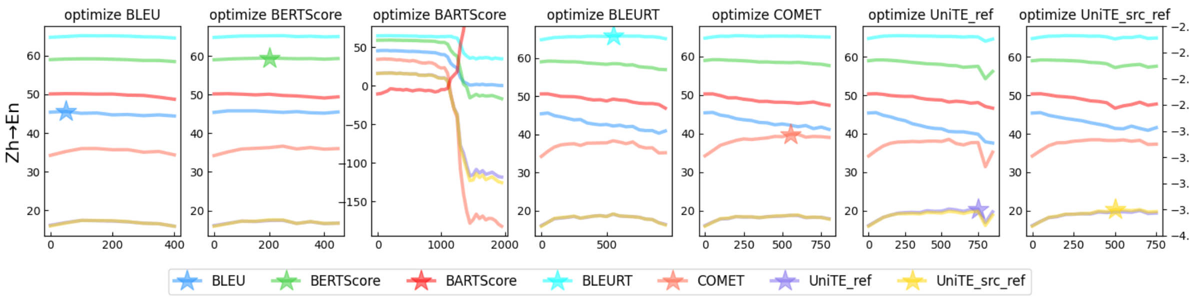

Figure 2: The training process of MRT optimized by each metric on Zh⇒En. The horizontal axis represents the training steps, and the vertical axis is the score of each metric (except for BARTScore on the right axis, which is a negative number because it calculates the logarithmic probability of translations); metrics other than BARTScore and BLEU are mostly distributed between 0 and 1, and we multiply them uniformly by 100 for ease of observation. The asterisk represents the highest value achieved by the optimized metric.

process.

However, we find several exceptions, such as optimizing BLEURT on the En⇒De and En⇒Zh language pairs, where the rest of the metrics experience a severe drop. As shown in Table 4, BLEURT remains basically stable, but the rest of the metrics drop to particularly low or even negative values. The same situation occurs when optimizing BARTScore, as shown in Table 4 and Figure 2.

MRT Exposes the Robustness Defects of Metrics We find deficiencies in some metrics when MRT collapses. For example, we find that there are universal adversarial translations in both BLEURT and BARTScore.

(1) Universal translations of BLEURT. We take the checkpoint of the translation model on En⇒De where BLEURT reaches the highest point to generate translations on the test set. The decoded results show that the translation quality does indeed collapse severely. Table 5 shows the two most frequently decoded translations. It can be seen that the translation model generates many similar sentences with high frequency, regardless of the source sentences. This shows that decoding such sentences can get high BLEURT scores. The example of calculating the BLEURT scores of universal translations is also shown in Figure 1.

(2) Universal translations of BARTScore. We also generate translations with the checkpoint on De⇒En which gets highest BARTScore. As shown in Table 5, the translation model also decodes many similar sentences, but unlike BLEURT, the form of the high-frequency decoded sentences is only repetition of simple words.

The phenomenon of universal adversarial translations shows that BLEURT and BARTScore are flawed, and a high metric score does not mean high translation quality. If the metric is not good enough, it actually leads the translation model in the wrong direction.

2.5 Analysis

2.5.1 Why Universal Translations Exist

We examine the WMT14 En⇔De parallel corpora, and find that there are many sentences with similar semantics in the training set, including a large corpus of hotel reviews that are semantically similar to universal translations of BLEURT [9]. This indicates that the patterns of universal translations are related to the translation training set, and they come from the high frequency samples in the training corpora. Raunak et al. (2021) also mentions the problem of corpus bias, whose study on NMT hallucinations shows that specific noise patterns in the training corpora lead to specific hallucination patterns. Due to high frequency samples in the translation training set, it becomes easy for the translation model to decode certain sentences (even if they have nothing to do with the source sentences).

Moreover, the high score of the metric condones the model to decode such sentences, leading to

[9] Some examples can be found in Appendix E.

Optimized Metric	Change Range of Metrics During MRT on En⇒De						
	BLEU	BERTScore	BARTScore	BLEURT	COMET	UniTE_ref	UniTE_src_ref
BLEU	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
BERTScore	0.70%	0.69%	-0.24%	0.71%	4.61%	4.31%	3.98%
BARTScore	-100.00%	-176.84%	92.15%	-79.78%	-574.39%	-397.07%	-385.80%
BLEURT	-100.00%	-107.48%	-20.33%	14.96%	-435.00%	-423.12%	-408.72%
COMET	-14.79%	-3.01%	-0.92%	1.65%	13.38%	10.86%	9.90%
UniTE_ref	-31.69%	-11.51%	-3.12%	-0.37%	2.66%	19.11%	18.06%
UniTE_src_ref	-39.08%	-15.27%	-4.16%	-2.39%	-4.28%	21.99%	21.72%

Table 4: The change range of all metrics when one metric is optimized to the highest value during MRT on En⇒De. 0.00% means that the optimized metric does not continue to improve, and the highest value remains the same as the result of MLE training; a negative number means that the metric score goes from positive to negative, which means it decreases a lot. (For the results of the remaining five language directions, see Appendix D)

	Frequency	Decoded Translations with Top2 Frequency
Optimize BLEURT on En⇒De	689	Lage vom Hotel war grundsätzlich bestens Hotelpersonal weitgehend zuvorkommend bzw. ggf. hilfehilfsbereit. Vor allem die Lage des Hotels war gut, Hotelmitarbeiter grundsätzlich äußerst lieb bzw. gegebenenfalls auch durchaus hilfehilfsbereit.
	386	Lage vom Hotel war grundsätzlich bestens HotelPersonal weitgehend zuvorkommend bzw. ggf. hilfehilfsbereit. Vor allem die Lage des Hotels war gut, Hotelmitarbeiter grundsätzlich äußerst lieb bzw. gegebenenfalls auch durchaus hilfehilfsbereit.
Optimize BARTScore on De⇒En	141	! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca!
	137	Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca! Mallorca!

Table 5: Examples of decoded translations of BLEURT and BARTScore. Due to space limitations, only the top2 frequency translations are listed.

the creation of universal translations. BLEURT uses metric data and generates a large amount of pseudo-data for supervised training, and the metric data comes from the translation training corpus. Since data augmentation may introduce noise and amplify hallucinations (Raunak et al., 2021), we suggest that its indulgence of universal translations is also related to the training corpus.

The universal translations of BARTScore contain repetitions of simple words, which is similar to the hallucination phenomena that occurs in the early stages of translation model training. We not only use the $F1$ score, but also experiment with the $Recall$ of BARTScore (computing $P(ref|hyp)$) to guide the training, and find that this setting can produce universal suffixes, that is, even if the correct translation is followed by a specific suffix, it does not reduce the BARTScore. Therefore, we suggest that the vulnerability of BARTScore may be due to the fact that it uses model generation probabilities to determine translation quality, and this generation-based metric tends to assign high scores to easily generated sentences. In short, the defects may stem from the tendency of the metric modeling paradigm [10].

The phenomenon of universal adversarial translations suggests that, on the one hand, we need to optimize the translation and metric datasets to balance their distributions, avoiding high-frequency samples; on the other hand, we need to optimize the metrics so that they are as little affected by the distribution bias of the dataset as possible. For example, sentence-level metrics can be constrained by incorporating word-level information. We present this experiment in Section 3.

2.5.2 Comparison of Metrics

We observe and compare the changes in the training effect of translation models guided by each metric on each language pair, and the summary is as follows:

BLEU converges quickly. This is as expected, since the translation model is selected by BLEU in the general MLE training phase, there is almost no continuous optimization during MRT. BERTScore also converges in a few steps. When BERTScore is optimized, other metrics remain relatively stable

[10] Metrics can be categorized into different modeling paradigms, including matching, regression, generation, and so on (Sun et al., 2022; Yuan et al., 2021).

and sometimes show an upward trend.

The consistency between BLEURT and other metrics shows language pair differences: for translate-to-English language pairs, the other metrics change steadily and show high consistency with BLEURT. All three to-En language pairs show an increase in COMET, UniTE_ref, and UniTE_src_ref. However, on the language pairs that translate from English, the consistency becomes very poor, where the other metrics drop significantly when optimizing translation models with BLEURT.

The metric that is least consistent with other metrics is BARTScore. On all language pairs, the rest of the metrics decrease when BARTScore is used to train translation models.

COMET, UniTE_ref, and UniTE_src_ref are similar and can improve each other. However, when optimizing with these metrics, a decrease in BLEU is observed for all language pairs. This may indicate that the translation model is gradually trained to be more inclined towards translations that are semantically close to the reference sentences, but the specific words may not be the same. In addition, other metrics also show a smooth trend of change, indicating that these metrics may be superior and more robust.

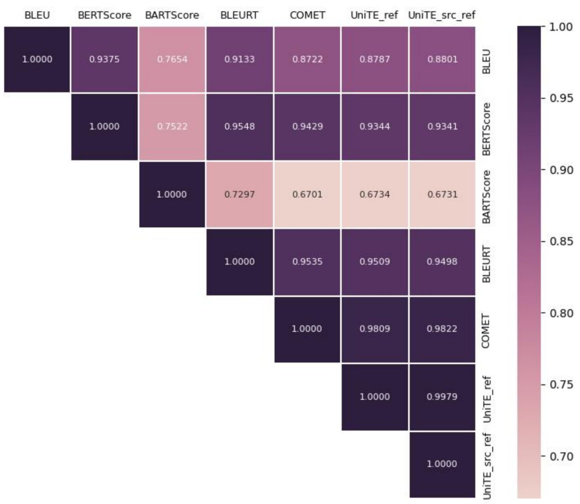

Figure 3: Pairwise correlation of metrics on translate-to-English language pairs. All metrics are significantly positively correlated ($p < 0.001$). Ignoring self-relevance, the correlations between COMET and UniTE, BLEURT and BERTScore are particularly strong.

Same Pre-trained Model Leads to Similar Metrics. We also find a pattern that metrics that are based on the same pre-trained model have similar trends in the variation of the training effect of MRT. We count the pairwise correlation of each metric, and find that the correlation between BERTScore and BLEURT (both based on BERT), and the correlation between COMET and UniTE (both based on XLM-Roberta) are higher than other metric pairs for translate-to-English language pairs, as shown in Figure 3. For language pairs translated from English, the robustness bias of BLEURT weakens its correlation with BERTScore, but the Pearson correlation coefficient still reaches 0.82 and is significantly correlated. This indicates that metrics based on the same pre-trained model have more consistent criteria for the evaluation of translation quality.

Robust Metrics can Drive Improvement in Other Metrics. MRT experiments show that the optimization process of BARTScore as well as BLEURT (on translation-from-English language pairs) is accompanied by a strong decrease of the other metrics, and we find metric robustness deficits in these cases. Therefore, we suggest that robust metrics may drive other metrics to improve together during MRT. (However, the converse inference does not hold. The ability to drive other metrics to improve is not sufficient to conclude that the metrics are robust enough, because metrics may have common deficits that have not yet been discovered.)

3 Optimize Metrics and Translations

The analysis of the MRT training process allows us to understand the impact of each metric on translation quality. Our goal is both to exploit the advantages of the MRT training approach and to avoid training collapse due to the robustness deficiencies of the metrics.

MRT needs to sample many translation sentences in advance, and then use sentence-level metrics to predict the scores and calculate the loss. If the metrics that guide translation training do not take word-level information into account, the translation model may ignore details and gradually deviate during the training process. Therefore, we try two methods to constrain the training direction by introducing word-level constraints: combining MRT and NLL loss, and doing metrics ensemble.

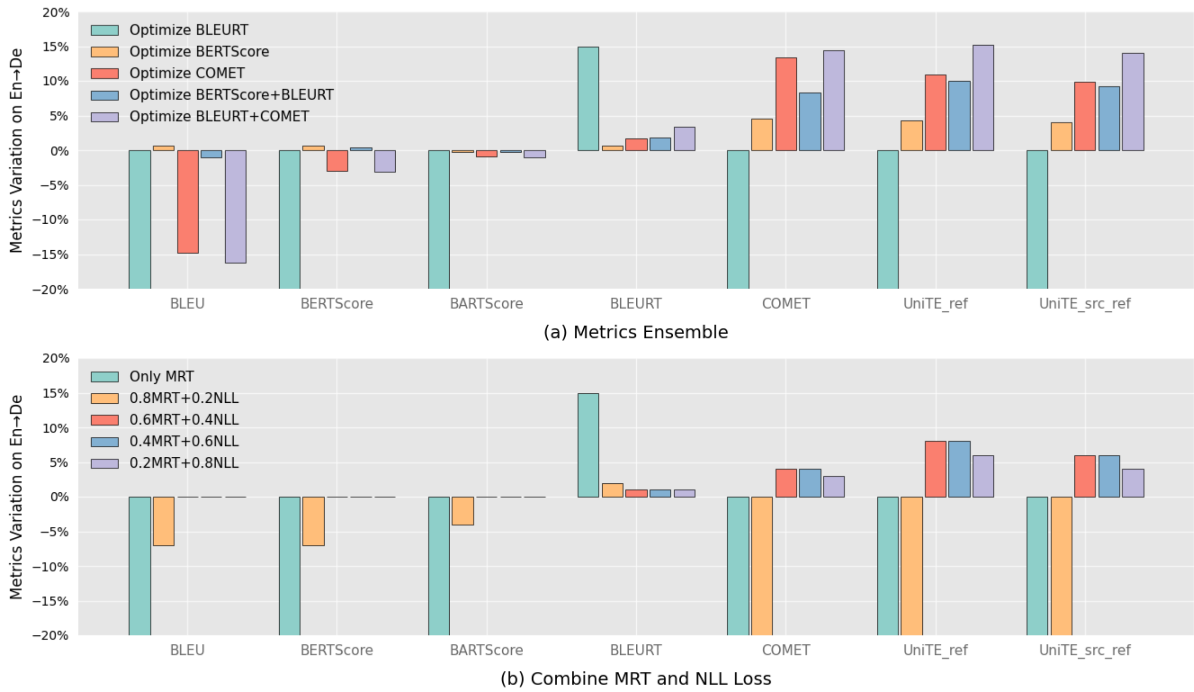

Figure 4: This figure displays two improvement strategies (<a> metrics ensemble and <b> modifying loss), with each color representing a different training approach. The figure is divided into seven groups from left to right, representing the range of change in each metric after training with a particular approach. For instance, the green bar represents the impact of utilizing BLEURT to guide translation model training in MRT, where only BLEURT improves while the other metrics decline significantly (we only display the range of -20% to 20%), which is consistent with the previous charts.

3.1 Combine MRT and NLL Loss

3.1.1 Experiments

We take the fine-grained word-level similarity as a part of the objective function by incorporating the NLL loss, which computes the log loss for each token. We set the hyperparameter λ_{MRT} to control the weights. The formula is as follows:

$$\mathcal{L} = \lambda_{MRT} * \mathcal{L}_{MRT} + (1 - \lambda_{MRT}) * \mathcal{L}_{NLL} \quad (2)$$

We take the MRT training effect of the translation model optimized with BLEURT on En⇒De as an example to conduct experiments.

3.1.2 Results

The results are shown in Figure 4 (b). As can be seen, as the proportion of NLL loss increases, the decreasing trend of the remaining metrics gradually disappears.

The optimal result can be achieved when $\lambda_{MRT} = 0.6$ or 0.4. At this point, unsupervised metrics remain stable, and supervised metrics show an increase. This indicates that combining MRT and NLL loss can improve the training effect of the translation model. For a fair comparison, we also check the results at the beginning of the optimization when using only MRT (before the training collapses). At this point, the improvement in BLEURT, COMET and UniTE is more obvious, but accompanied by a decrease in BLEU and BERTScore. This suggests that the inclusion of NLL loss can make training more stable and more balanced across all metrics.

3.2 Metrics Ensemble

3.2.1 Experiments

Supervised metrics focus more on high-level semantic similarity and are considered to have a higher correlation with human evaluation (Kocmi et al., 2021); while unsupervised metrics using word-level information are relatively stable and can ensure fine-grained text similarity [11].

We do an ensemble of different metrics in the hope that the integrated metrics can complement each other and integrate the advantages of different

[11] Note that although BARTScore is an unsupervised metric, it calculates the overall probability of sentence generation and still focuses more on sentence-level information.

metrics. Then the ensemble metric is applied to MRT training on En⇒De.

3.2.2 Results

Supervised and Unsupervised Metrics Ensemble. As can be seen in Figure 4 (a), optimizing BERTScore alone does not change the remaining metrics much, while only optimizing BLEURT reveals robustness problems. However, optimizing the ensemble of BERTScore and BLEURT works well: not only does it preserve the performance of the unsupervised metrics as much as possible, but it also leads to significant improvements in COMET and UniTE.

Supervised Metrics Ensemble. In addition, combining two sentence-level supervised metrics can also provide a boost, as the fifth column of Figure 4 (a) shows the effect of integrating BLEURT and COMET. Compared to optimizing only a single metric, we find that the ensemble metric can build on the strengths of both metrics. While maintaining the scores of unsupervised metrics, it can further improve supervised metrics. COMET and UniTE all improve about 14.5%, which is an increase of about 7 points. We suggest that this may be due to the fact that different metrics have different criteria for evaluating translation quality, and the robustness deficiency of one metric can be compensated by other metrics.

3.3 Method Validity Analysis

Avoid High-Frequency Decoding Sentences. We compare the entropy of decoded sentence frequencies on the En⇒De test set for the translation model trained with single or ensemble metrics. As shown in Table 6, the entropy is lower for the model trained with only BLEURT because it decodes a large number of identical sentences. While the frequency entropy for models trained with ensemble metrics is similar to that of the gold translations, indicating that the phenomenon of high-frequency decoded sentences disappears.

Comparison to MBR Decoding. Minimum Bayes Risk (MBR) decoding can also get translations with metric style (Freitag et al., 2022; Müller and Sennrich, 2021). Both MRT and MBR add some computational cost because they need to sample candidate translation sentences. However, MRT is a training process that can quickly generate translations at test time once the model has finished training. MBR, on the other hand, is a decoding process, which requires more time for each decoding. Therefore, from an application point of view, MRT is more efficient.

System	Entropy
Ref	11.55
Hyp (Only BLEURT)	6.58
Hyp (BLEURT + BERTScore)	11.55
Hyp (BLEURT + COMET)	11.55

Table 6: The entropy of decoded sentence frequencies on the En⇒De test set. Ref is the gold translation. Low entropy means that the translation model is damaged and decodes many identical sentences.

4 Related Work

Automatic Metrics Traditional metrics for machine translation evaluation including BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009), and chrF (Popović, 2015) are based on lexical overlap. Embedding-based metrics measure the semantic equivalence between the reference and translation hypothesis by contextual representation, such as BERTScore (Zhang et al., 2019), MoverScore (Zhao et al., 2019). Generation-based metrics formulate the evaluation of text as a generation task, such as BARTScore (Yuan et al., 2021) and PRISM (Thompson and Post, 2020). The basic idea is that high quality text can be generated with high probability. Learned metrics, such as BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020), and the recently proposed UniTE (Wan et al., 2022) aim to train neural networks to directly predict human judgements. These supervised metrics correlate well with human evaluations, but lack interpretability and robustness studies, which is explored by this paper.

Minimum Risk Training Shen et al. (2015) proposes the MRT method and confirms its superiority with experiments. Edunov et al. (2017) compares various objective functions and further verifies that MRT training can enhance translation quality. Wang and Sennrich (2020) uses MRT to avoid exposure bias, thus improving translation quality in out-of-domain settings. The above MRT work uses BLEU to guide the training of translation models, but BLEU is not the optimal metric. Our work uses various cutting-edge metrics to further improve translation quality. Wieting et al. (2019) proposes a new metric, claiming its superiority over BLEU and suitability for MRT training. Our work, on the other hand, focuses on the analysis of metrics, with MRT serving as a tool to evaluate the robustness of various metrics systematically.

Metric Defects Analysis There are also some papers that start to explore the shortcomings of metrics. Sai et al. (2021) provides perturbation templates to measure the performance of metrics on the constructed challenge set, while our work is to guide the metrics to generate adversarial samples (universal translations) by themselves. Amrhein and Sennrich (2022) does a case study on COMET through MBR decoding, showing that COMET is insensitive to numbers and named entities. Different from a pure case study, our work shows the tendency of metrics through MRT, and can draw more typical conclusions. Sun et al. (2022) shows that PLM-based metrics, such as BERTScore, lack fairness and exhibit higher social bias than traditional metrics. Our work analyzes metrics from a robustness perspective and complements this work.

5 Conclusion

In this paper, we present the first systematic analysis of automatic metrics from the perspective of guidance for training machine translation systems. We find that MRT reveals the robustness deficiencies of some metrics, such as universal adversarial translations of BLEURT and BARTScore, and we further analyze the underlying causes. In addition, we explore methods to improve metric robustness, thus helping to further enhance the performance of translation systems.

Limitations

First, we find robustness deficiencies in metrics by comparing the evaluation differences among metrics. This applies to the case when there are metrics that do not have the same robustness flaws. If there are more latent common defects in the metrics, they cannot be identified by MRT. We leave this topic for future research.

Second, we use beam search to generate candidates during MRT training, but beam search is also known to have deficiencies. For example, beam search suffers from heuristic search biases and shifts statistics away from those of the data (Eikema and Aziz, 2020). Different decoding methods may have an impact on the experiment results.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by National Science Foundation of China (No. 62176120), the Liaoning Provincial Research Foundation for Basic Research (No. 2022-KF-26-02).

References

Chantal Amrhein and Rico Sennrich. 2022. Identifying weaknesses in machine translation metrics through minimum bayes risk decoding: A case study for comet. *arXiv preprint arXiv:2202.05148*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2017. Classical structured prediction losses for sequence to sequence learning. *arXiv preprint arXiv:1711.04956*.

Bryan Eikema and Wilker Aziz. 2020. Is map decoding all you need? the inadequacy of the mode in neural machine translation. *arXiv preprint arXiv:2005.10283*.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *arXiv preprint arXiv:2107.10821*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut.

2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Alon Lavie and Michael J Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2):105–115.

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. *arXiv preprint arXiv:1907.11692*.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *arXiv preprint arXiv:2106.15195*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*.

Mathias Müller and Rico Sennrich. 2021. Understanding the properties of minimum bayes risk decoding in neural machine translation. *arXiv preprint arXiv:2105.08504*.

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.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Ananya B Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M Khapra. 2021. Perturbation checklists for evaluating nlg evaluation metrics. *arXiv preprint arXiv:2109.05771*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*.

Aaron Smith, Christian Hardmeier, and Jörg Tiedemann. 2016. Climbing mont bleu: the strange world of reachable high-bleu translations. In *Proceedings of the 19th annual conference of the European association for machine translation*, pages 269–281.

Tianxiang Sun, Junliang He, Xipeng Qiu, and Xuanjing Huang. 2022. Bertscore is unfair: On social bias in language model-based metrics for text generation. *arXiv preprint arXiv:2210.07626*.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. *arXiv preprint arXiv:2004.14564*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek F Wong, and Lidia S Chao. 2022. Unite: Unified translation evaluation. *arXiv preprint arXiv:2204.13346*.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. *arXiv preprint arXiv:2005.03642*.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond bleu: training neural machine translation with semantic similarity. *arXiv preprint arXiv:1909.06694*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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. *arXiv preprint arXiv:1909.02622*.

A Ethics Statement

This paper finds universal adversarial translations that can be used to attack metrics and lead to security risks. However, this paper also proposes methods to improve metric robustness to avoid this situation.

B Training Duration for MLE and MRT

We list the training duration for MLE and MRT in Table 8 and Table 9, respectively. Table 8 shows the number of training epochs, while Table 9 shows the number of training steps. It can be seen that the training duration for MRT is much shorter than that for MLE. The improvement of translation performance of the model mainly lies in the MLE stage, while MRT fine-tuning makes the model inclined towards specific metrics.

C MRT Training Process Figures

From Figure 5 to Figure 9, we can see how all metrics change when the translation model is optimized with each metric on different language pairs. From the trends of different metrics, we can observe the differences between the metrics and the impact of the metrics used for optimization on the translation model.

In each figure, the horizontal axis represents the training steps, and the vertical axis is the score of each metric (except for BARTScore on the right axis, which is a negative number because it calculates the logarithmic probability of translations); metrics other than BARTScore and BLEU are mostly distributed between 0 and 1, and we multiply them uniformly by 100 for ease of observation. The asterisk represents the highest value achieved by the optimized metric.

D MRT Training Process Statistics

Table 10 to Table 14 display the change range in all metrics when optimizing the translation model with a specific metric to the highest point across different language pairs. The results correspond to figures in Appendix C. 0.00% means that the optimized metric does not continue to improve, and the highest value remains the same as the result of MLE training; a negative number means that the metric score goes from positive to negative, which means it decreases a lot.

Examples of Hotel Review Sentences from WMT14 En⇔De
The location of the hotel was excellent. The room was clean and comfortable.
The room was clean and comfortable, the hotel was situated close to the center but in the tourist center. The food was excellent and the service second to none.
The location of the hotel is great, the atmosphere is quite pleasant, the staff is efficient and friendly, the room was clean and comfortable, the price was fair. In short words, everything was perfect.
The room was clean and comfortable.
the location of the hotel is ideal for sightseeing,the room was clean and comfortable, the staff were helpful.
The room was clean and comfortable. Staff friendly.
the employees were very helpful at all times the room was clean and comfortable and the restaurant was very nice.
The room was clean and comfortable and the staff friendly and courteous.
This is a great hotel .The room was clean and comfortable .With small budget but we have a comfortable stay .Good value, we will reccommend this hotel for anyone looking for a hotel in Hanoi .

Table 7: Examples of Hotel Review Sentences from WMT14 En⇔De.

E High Frequency Samples

Table 7 displays some hotel review examples in the WMT14 En⇔De dataset, and the semantics are very similar to universal translations of BLEURT on En⇒De. For ease of understanding, English is shown here.

		En⇒De	De⇒En	En⇒Zh	Zh⇒En	En⇒Fi	Fi⇒En
MLE		33	28	32	40	55	36
MRT	BLEU	1	1	1	1	1	1
	BERTScore	1	1	1	1	1	1
	BARTScore	1	1	1	1	1	1
	BLEURT	4	1	1	1	1	1
	COMET	1	1	1	1	1	1
	UniTE_ref	1	1	1	1	1	1
	UniTE_src_ref	1	1	1	1	1	1

Table 8: Comparison of the number of epochs trained by MLE and MRT. The number of epochs for MLE is the epoch number trained until early stop, while the number of epochs displayed in MRT is the epoch number when the model is optimized to the highest metric score.

		En⇒De	De⇒En	En⇒Zh	Zh⇒En	En⇒Fi	Fi⇒En
Steps in one epoch		2403	2127	7927	7929	10906	10910
MLE		163000	61000	51000	64000	126000	40000
MRT	BLEU	0	50	0	200	100	50
	BERTScore	100	50	250	200	250	300
	BARTScore	1950	1900	1800	1450	1050	550
	BLEURT	5750	100	3500	550	1400	250
	COMET	550	450	500	550	800	300
	UniTE_ref	400	650	750	750	600	600
	UniTE_src_ref	600	350	700	500	800	550

Table 9: Comparison of training steps between MLE and MRT. The number of steps for MLE is the number of steps trained until early stop, while the number of steps displayed in MRT is the number of steps when the model is optimized to the highest metric score.

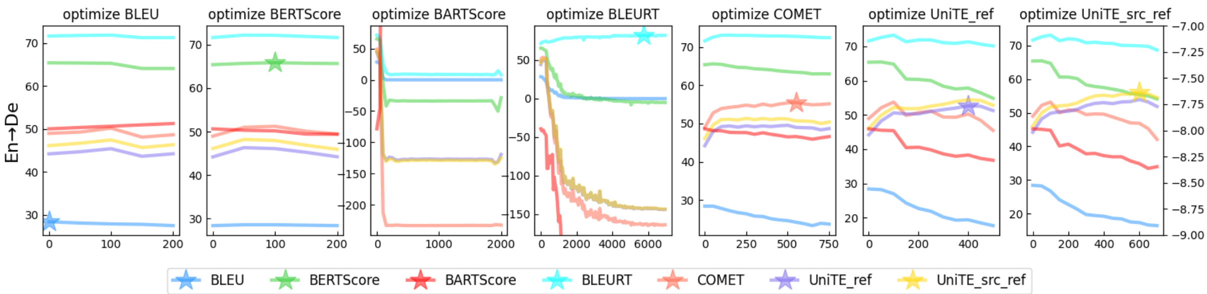

Figure 5: The training process of MRT optimized by each metric on En⇒De.

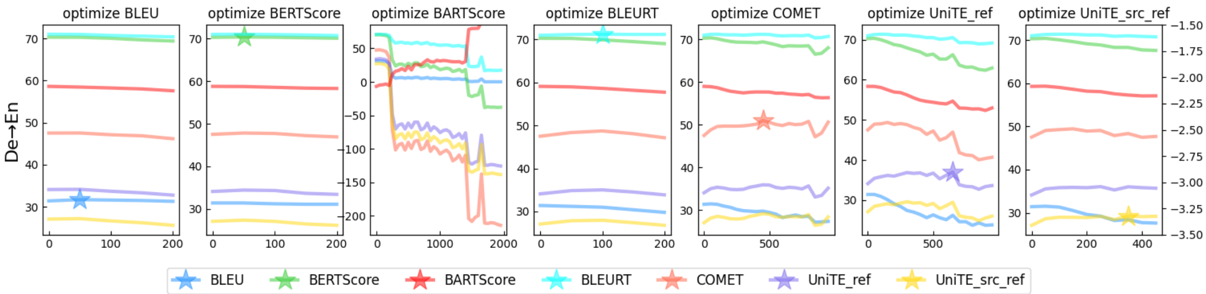

Figure 6: The training process of MRT optimized by each metric on De⇒En.

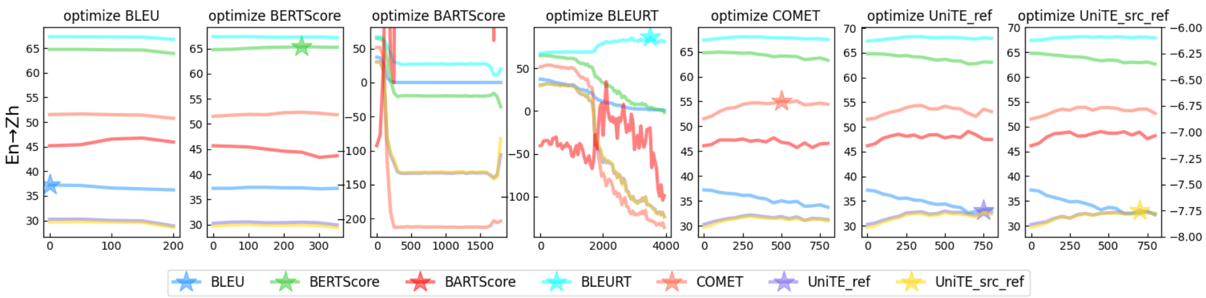

Figure 7: The training process of MRT optimized by each metric on En⇒Zh.

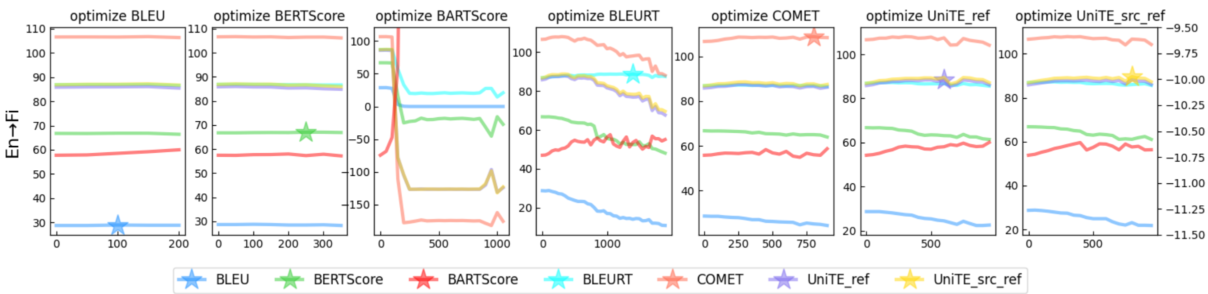

Figure 8: The training process of MRT optimized by each metric on En⇒Fi.

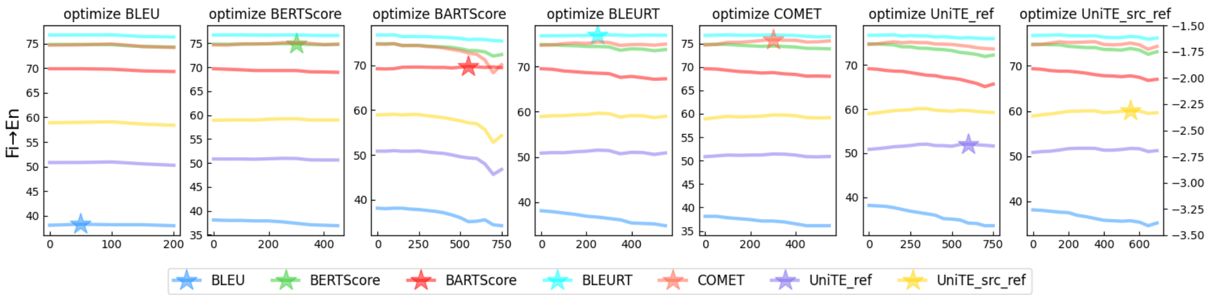

Figure 9: The training process of MRT optimized by each metric on Fi⇒En.

Optimized Metric	**Change Range of Metrics During MRT on De⇒En**						
	BLEU	BERTScore	BARTScore	BLEURT	COMET	UniTE_ref	UniTE_src_ref
BLEU	0.96%	-0.03%	-0.32%	-0.06%	0.04%	0.14%	0.41%
BERTScore	0.00%	0.12%	-0.07%	0.05%	0.65%	0.93%	1.04%
BARTScore	-99.68%	-154.39%	39.02%	-76.08%	-547.63%	-466.31%	-608.53%
BLEURT	-1.27%	-0.50%	-0.91%	0.36%	2.58%	2.72%	3.24%
COMET	-5.10%	-1.23%	-2.57%	0.28%	7.41%	5.45%	7.84%
UniTE_ref	-16.56%	-5.72%	-6.71%	-0.64%	-1.13%	8.49%	8.76%
UniTE_src_ref	-9.55%	-2.85%	-3.96%	-0.01%	1.73%	5.57%	7.99%

Table 10: The change range of all metrics when one metric is optimized to the highest value during MRT on De⇒En.

Optimized Metric	**Change Range of Metrics During MRT on En⇒Zh**						
	BLEU	BERTScore	BARTScore	BLEURT	COMET	UniTE_ref	UniTE_src_ref
BLEU	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
BERTScore	0.27%	0.88%	-0.86%	-0.10%	1.55%	0.73%	0.76%
BARTScore	-100.00%	-155.17%	88.89%	-70.57%	-495.64%	-450.90%	-377.30%
BLEURT	-96.77%	-95.08%	2.40%	28.97%	-349.43%	-465.65%	-472.03%
COMET	-5.91%	-1.00%	0.46%	0.67%	6.96%	5.27%	6.05%
UniTE_ref	-10.48%	-2.54%	0.87%	1.03%	4.01%	9.58%	10.66%
UniTE_src_ref	-12.63%	-2.90%	1.83%	1.07%	3.90%	9.51%	11.34%

Table 11: The change range of all metrics when one metric is optimized to the highest value during MRT on En⇒Zh.

Optimized Metric	Change Range of Metrics During MRT on Zh⇒En						
	BLEU	BERTScore	BARTScore	BLEURT	COMET	UniTE_ref	UniTE_src_ref
BLEU	0.44%	0.26%	0.10%	0.29%	2.93%	4.32%	4.77%
BERTScore	0.44%	0.80%	-0.10%	0.66%	6.05%	8.16%	10.24%
BARTScore	-98.02%	-126.46%	37.25%	-44.95%	-620.81%	-825.23%	-871.23%
BLEURT	-7.05%	-0.80%	-1.86%	1.65%	11.99%	18.67%	20.28%
COMET	-7.05%	-0.98%	-3.12%	0.77%	16.47%	16.50%	17.67%
UniTE_ref	-12.11%	-2.45%	-3.09%	0.61%	10.25%	26.77%	24.30%
UniTE_src_ref	-8.81%	-3.01%	-5.14%	-0.16%	11.75%	21.83%	27.31%

Table 12: The change range of all metrics when one metric is optimized to the highest value during MRT on Zh⇒En.

Optimized Metric	Change Range of Metrics During MRT on En⇒Fi						
	BLEU	BERTScore	BARTScore	BLEURT	COMET	UniTE_ref	UniTE_src_ref
BLEU	0.70%	0.12%	0.17%	0.05%	-0.03%	0.17%	0.23%
BERTScore	-0.70%	0.42%	-0.03%	-0.21%	-0.12%	-0.48%	-0.21%
BARTScore	-100.00%	-140.82%	83.08%	-75.93%	-264.54%	-244.66%	-241.44%
BLEURT	-51.22%	-21.42%	1.14%	2.19%	-8.02%	-10.43%	-9.85%
COMET	-12.20%	-2.58%	0.04%	0.07%	2.03%	0.86%	0.96%
UniTE_ref	-14.63%	-5.43%	0.76%	-0.21%	0.80%	3.00%	3.02%
UniTE_src_ref	-19.51%	-8.57%	0.74%	-1.05%	0.05%	2.69%	3.11%

Table 13: The change range of all metrics when one metric is optimized to the highest value during MRT on En⇒Fi.

Optimized Metric	Change Range of Metrics During MRT on Fi⇒En						
	BLEU	BERTScore	BARTScore	BLEURT	COMET	UniTE_ref	UniTE_src_ref
BLEU	0.52%	0.01%	-0.01%	0.02%	0.09%	0.01%	0.15%
BERTScore	-1.84%	0.13%	-0.91%	0.02%	0.58%	0.20%	0.55%
BARTScore	-7.87%	-2.04%	1.20%	-1.34%	-2.63%	-2.97%	-2.99%
BLEURT	-3.67%	-0.58%	-2.22%	0.29%	0.98%	1.30%	1.25%
COMET	-2.62%	-0.46%	-1.92%	0.11%	1.53%	1.12%	1.39%
UniTE_ref	-10.50%	-2.88%	-6.33%	-0.68%	-0.55%	2.39%	1.25%
UniTE_src_ref	-6.30%	-1.26%	-3.53%	-0.26%	0.53%	1.73%	2.05%

Table 14: The change range of all metrics when one metric is optimized to the highest value during MRT on Fi⇒En.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitation is provided after secion 6.

☑ A2. Did you discuss any potential risks of your work?
Potential risks are provided after secion 6 in Limitation part.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract is provided in the very beginning of the article and introduction is provided in section 1.

☑ A4. Have you used AI writing assistants when working on this paper?
"Deepl write"(https://www.deepl.com/write) is used as AI writing assistant in this paper. "Deepl write" assistances purely with the language of the paper. Deepl write is used in section 1, 2, 3 of the paper

B ☑ Did you use or create scientific artifacts?

Scientific artifacts are used in section 2. Scientific artifacts are created in section 3.

☑ B1. Did you cite the creators of artifacts you used?
References part(at the end of the paper) cite the creators of artifacts used in this paper. In addition, URLs are provided in section 2.1

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In section 2, we report relevant statistics

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Computational experiments are in section 2, 3.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We discuss the experimental setup in section 2.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We report descriptive statistics about results and report mean results. They are in section 2.4, 3.1.2, 3.2.2

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We report detail of used existing packages in section 2.3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.