Toward Human-Like Evaluation for Natural Language Generation with Error Analysis

Qingyu Lu[1,2]*, **Liang Ding**[2]*, **Liping Xie**[1], **Kanjian Zhang**[1]†, **Derek F. Wong**[3], **Dacheng Tao**[4]
[1]School of Automation, Southeast University [2]JD Explore Academy
[3]NLP[2]CT Lab, University of Macau [4]The University of Sydney
{luqingyu,lpxie,kjzhang}@seu.edu.cn, liangding.liam@gmail.com,
derekfw@um.edu.mo, dacheng.tao@gmail.com

Abstract

The pretrained language model (PLM) based metrics have been successfully used in evaluating language generation tasks. Recent studies of the human evaluation community show that considering both major errors (e.g. mistranslated tokens) and minor errors (e.g. imperfections in fluency) can produce high-quality judgments. This inspires us to approach the final goal of the automatic metrics (human-like evaluations) by fine-grained error analysis. In this paper, we argue that the ability to estimate sentence confidence is the tip of the iceberg for PLM-based metrics. And it can be used to refine the generated sentence toward higher confidence and more reference-grounded, where the costs of refining and approaching reference are used to determine the major and minor errors, respectively. To this end, we take BARTScore as the testbed and present an innovative solution to marry the unexploited sentence refining capacity of BARTScore and human-like error analysis, where the final score consists of both the evaluations of major and minor errors. Experiments show that our solution consistently improves BARTScore, outperforming top-scoring metrics in 19/25 test settings. Analyses demonstrate our method robustly and efficiently approaches human-like evaluations, enjoying better interpretability. Our code and scripts will be publicly released in https://github.com/Coldmist-Lu/ErrorAnalysis_NLGEvaluation.

1 Introduction

Leveraging the power of large pre-trained language models (PLMs) has been proven effective in evaluating natural language generation (NLG) tasks (Ma et al., 2019; Mathur et al., 2020b). Metrics like BERTScore (Zhang et al., 2020b) and MoverScore (Zhao et al., 2019) leverage contextual embeddings provided by PLMs to evaluate the semantic similarity of sentences. Regression-based metrics like COMET (Rei et al., 2020) and BLEURT (Sellam et al., 2020) introduce a regression layer following PLMs to learn a supervised prediction using human evaluation. Recently, another line of research focuses on generation probabilities of seq2seq PLMs to measure the confidence of generated texts, such as PRISM (Thompson and Post, 2020) and BARTScore (Yuan et al., 2021), achieving the decent performance. It is commonly agreed that the ultimate goal of automatic evaluation is to achieve consistency with humans, namely *human-like evaluation*.

Source:	迈克去书店。
Reference:	Mike goes to the bookstore.
Hypothesis:	Jerry goes to bookstore happily.

Iteration	Refined Sentence	BARTScore (↑)
0	Jerry goes to bookstore happily . (-15.16 -1.64 -0.48 -5.30 -14.51 -0.02)	-3.89
1	Mike goes to bookstore happily . (-4.06 -1.56 -0.54 -5.50 -14.42 -0.03)	-2.59
2	Mike goes to bookstore . (-4.06 -1.56 -0.54 -5.50 -0.06)	-1.45

Table 1: **An example of error analysis framework,** specifically, **detect-correct algorithm** in §3.2. Scores under each token represent the log probability assigned by BARTScore. Worse tokens detected by error analysis in each iteration are highlighted in yellow, and their corresponding scores are in **red**.

Recent studies of the human evaluation community show that the quality of human judgments can be improved through fine-grained error analysis, incorporated in an error-based framework Multidimensional Quality Metric (MQM) (Freitag et al., 2021a). MQM requires evaluators to identify errors and categorize them into different levels according to their severity. For instance, mistranslations (Weng et al., 2020) and hallucinations (Zhou et al., 2021) are mostly considered as *Major* errors, and imperfections in fluency (Chow et al., 2019) are often marked as *Minor* errors. Different weights

*Work was done when Qingyu was interning at JD Explore Academy. Qingyu and Liang contributed equally.
†Corresponding Author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5892–5907
July 9-14, 2023

are then assigned to Major/ Minor errors, resulting in high-quality human evaluation scoring.

Analogous to Major/ Minor errors in MQM, we take the first step to consider incorporating the evaluation of Explicit/ Implicit errors into PLMs-based metrics. Specifically, we use BARTScore, a state-of-the-art metric for NLG by Yuan et al. (2021) as the test bed, and propose a metric called BARTScore++. We present an overview of our proposed method in Figure 1. In particular, given the hypothesis and reference, we propose an error analysis framework to obtain a refined sentence (see example in Table 1) using BARTScore, where the costs of refining and approaching reference are used to determine the explicit and implicit errors, respectively. The weighted integration of these two types of errors is the final score of BARTScore++, which has better interpretability.

We experiment on machine translation (MT), text summarization (SUM), and data-to-text (D2T), and show that BARTScore++ consistently and significantly improves the performance of vanilla BARTScore, and surpasses existing top-scoring metrics in 19 out of 25 test settings, even exceeding human performance on summarization dataset `Rank19`. We give further analyses to confirm that the consistent improvements come from the human-like (specifically, MQM-like) error judgment.

Our **main contributions** are as follows:

- To the best of our knowledge, we take the first step toward human-like evaluation by incorporating error analysis mechanisms into existing advanced automatic metrics, e.g. BARTScore.
- We propose an innovative automatic error analysis framework to calculate the explicit error and implicit error-based scores, by refining sentences using BARTScore.
- We validate the effectiveness and universality of our method spanning 25 NLG evaluation tasks, achieving the SOTA in 19 settings.

Besides taking BARTScore as the testbed to verify the effectiveness of our proposed error-analysis evaluation strategy, we also show the universality in the recently advanced language model ChatGPT[1] by designing an error-analysis-based prompt (Lu et al., 2023). We anticipate that our strategy will shed new light on advancing the field of NLG evaluation with pretrained language models by enhancing both the accuracy and reliability of metrics.

2 Preliminaries

Problem Formulation The goal of NLG evaluation is to acquire a score measuring the quality of generated text $\boldsymbol{y}$ given a reference signal $\boldsymbol{r}$. Unless otherwise stated, $\boldsymbol{r}$ represents the sentence properly created by human experts to assist in evaluation, and $\boldsymbol{y} = (y_1 y_2 \dots y_N)$, called hypothesis in this paper, refers to the generated text to be evaluated[2].

BARTScore BARTScore is a SOTA metric proposed by Yuan et al. (2021) for universal NLG evaluation. The idea of BARTScore is to utilize the generation probabilities of a large pre-trained model BART (Lewis et al., 2020) to measure the quality of sentences. It autoregressively computes the log probabilities of each token in the hypothesis, and then averages them as the overall score. This evaluation process can be formally written as:

$$\text{BARTScore} = \frac{1}{N}\sum_{t=1}^{N} \log p_\theta \left(y_t | y_{<t}, \boldsymbol{r}\right)$$

Based on this formulation, BARTScore creates specific variants for different evaluation scenarios. We summarize their usage in Appendix A. For simplification, we use the notation of $\text{BARTS}(\boldsymbol{y}, \boldsymbol{r})$ when vanilla BARTScore is further applied.

MQM MQM is an error-based human evaluation framework, which is commonly agreed to be more reliable than traditional human evaluation techniques (Freitag et al., 2021b). In MQM framework, each evaluator is asked to identify all errors in a sentence and categorize them into *Major* and *Minor* levels indicating their severities. Sentences will be marked an *Non-translation Error* if they are not possible to reliably identify errors. Major/ Minor errors are then assigned with different weights, and the final MQM score is computed through the weighted sum of errors (Freitag et al., 2021a). Inspired by the mechanism of MQM, we take a step toward human-like evaluation by incorporating error analysis into BARTScore.

3 Methodology

To better understand how BARTScore++ works, we show a running example of our method in Figure 1.

[1] `https://chat.openai.com/`

[2] Note that in text summarization evaluations, BARTScore may use the source sentence as the reference signal.

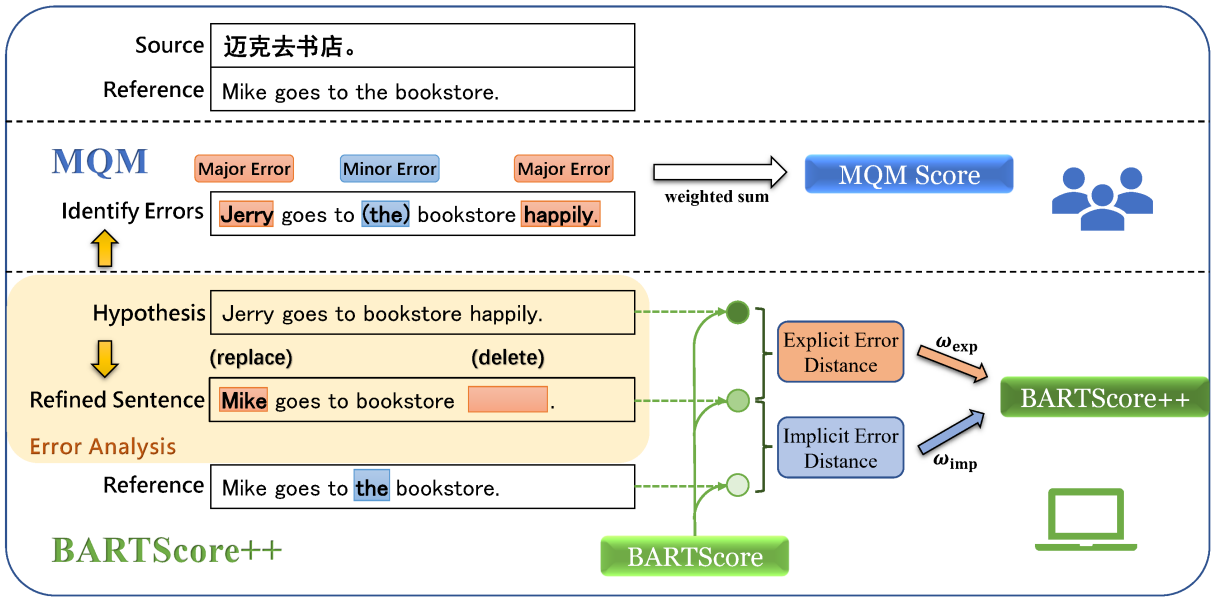

Figure 1: **An analogy between MQM and BARTScore++.** We show an evaluation example from machine translation (zh-en). **Top**: Source and reference sentence provided for evaluation. **Medium**: An annotation example using MQM framework. Errors in the hypothesis are assigned with *Major* and *Minor*. The MQM score is computed through the weighted sum of these errors. **Bottom**: BARTScore++. The hypothesis is first refined through an error analysis framework. The refined sentence is then used to obtain the distance of explicit/ implicit errors through vanilla BARTScore. Different weights are finally assigned to these errors to get a more accurate score.

3.1 Explicit/ Implicit Error Distance

Analogous to major errors in MQM, we define *Explicit Errors* to refer to errors that can be easily identified. In our example, mistranslations of name ("Mike" $\rightarrow$ "Jerry") and addition of "happily" are considered as explicit errors. Analogous to minor errors, we define *Implicit Errors* to indicate the semantic imperfections (e.g. disfluency, awkwardness) that may not influence the overall meanings. In our example, the missing article "the" is considered as an implicit error because it is a smaller imperfection in grammar.

To measure the influence of Explicit/ Implicit errors in the hypothesis $\boldsymbol{y}$, we define *Refined Sentence* $\boldsymbol{y}^*$ as a better hypothesis, where explicit errors are corrected. In this way, distances of explicit/ implicit error can be computed by:

$$\text{Dist}_{\text{exp}} = \text{BARTS}(\boldsymbol{y}^*, \boldsymbol{r}) - \text{BARTS}(\boldsymbol{y}, \boldsymbol{r})$$
$$\text{Dist}_{\text{imp}} = \text{BARTS}(\boldsymbol{r}, \boldsymbol{r}) - \text{BARTS}(\boldsymbol{y}^*, \boldsymbol{r})$$

We then focus on how to 1) obtain the refined sentence $\boldsymbol{y}^*$ and 2) take both explicit/ implicit errors into consideration and obtain the final score.

3.2 Error Analysis Framework

We introduce an automatic error analysis framework to generate the refined sentence $\boldsymbol{y}^*$ by correcting explicit errors in the hypothesis $\boldsymbol{y}$. We first adopt a simple **non-translation test** to decide whether $\boldsymbol{y}$ will be refined or not. Then, a **detect-correct algorithm** is performed iteratively, in each round one token is detected and then corrected. An example of this is shown in Table 1. This algorithm repeats for a determined number of iterations T, where at the end of each round the refined sentence $\boldsymbol{y}^*$ is updated and becomes a new refining target. In our example, the hypothesis $\boldsymbol{y}$ is refined twice, where the mistranslated token "Jerry" is detected in Round 1 and corrected as "Mike", and the addition of "happily" is detected and deleted in Round 2. Afterwards, an extra round will run (omitted in table) to ensure that none of the tokens needs to be corrected. Finally, the hypothesis "Mike goes to bookstore." is taken as the refined sentence $\boldsymbol{y}^*$.

Test Non-Translation Error Non-Translation Error is used in MQM (Freitag et al., 2021a) to refer to the translation which is too badly garbled or is unrelated to the source. If the hypotheses contain severe problems such as off-target issues (Zhang et al., 2020a), directly refining them will consume excessive computational cost. To avoid this problem, we run a test beforehand as a rough measure to filter out these hypotheses with low quality. We consider two strategies:

1. **Token-level overlap ratio** w.r.t the reference. Inspired by string-based metrics like BLEU (Papineni et al., 2002) or TER (Snover et al., 2006), the hypothesis with a non-translation error may be quite different from its reference, resulting in a low overlap ratio. Since good translations like paraphrased sentences (Freitag et al., 2020) may not have significant overlap with the reference, we adopt the other strategy as a double-check.

2. **Percentage of tokens with low generation probability**. Token-level log generation probability can be directly obtained from vanilla BARTScore as $\log p_\theta\left(y_t|y_{<t}, \boldsymbol{r}\right)$. If most tokens' generation probabilities are lower than the average score (vanilla BARTScore), we mark this sentence as non-translation. This strategy is more stable but less efficient.

Detect In this step, we choose **one token** $\hat{y}_t$ with the lowest generation probability as the token to be corrected. This procedure can be denoted as:

$$\hat{y}_t = \arg\min_{y_t} \left\{ p_\theta\left(y_t|y_{<t}, \boldsymbol{r}\right) \right\}$$

Correct In this step, we leverage the distribution of generation $p_\theta\left(\cdot|y_{<t}, \boldsymbol{r}\right)$ to propose several refining options from vocabulary $\mathcal{V}$. We apply the **top-k sampling** method (Fan et al., 2018) to obtain a set of candidate tokens ($\mathcal{W}$) with the highest generation probability:

$$\mathcal{W} = \arg\max_{w \in \mathcal{V}} \left\{ p_\theta\left(w|y_{<t}, \boldsymbol{r}\right), k \right\}$$

Then, a set of refined sentences $\mathcal{S}$ is proposed. Following Snover et al. (2006), we apply three types of editing strategies, including insertion of a candidate token $w \in \mathcal{W}$, deletion of token $\hat{y}_t$, and substitution of $\hat{y}_t$ for a candidate token $w \in \mathcal{W}$. Finally, we use vanilla BARTScore to select the best sentence $\hat{\boldsymbol{y}}^*$ as the refining strategy:

$$\hat{\boldsymbol{y}}^* = \arg\max_{\hat{\boldsymbol{y}} \in \mathcal{S}} \text{BARTS}\left(\hat{\boldsymbol{y}}, \boldsymbol{r}\right),$$

where the hypothesis $\boldsymbol{y}$ will be temporarily replaced by $\hat{\boldsymbol{y}}^*$ and as the input for the next iteration.

This *detect-correct* algorithm repeatedly detects the worst token $\hat{y}_t$ and corrects it. It starts with the original hypothesis $\boldsymbol{y}$ and ends after a constant number of edits. We set an early-stop mechanism once the BARTScore performance stops improving.

In this way, we obtain the refined sentence $\boldsymbol{y}^*$, which is also a by-product of our method.

3.3 Assigning Error Weights

With the help of the error analysis framework, explicit errors in the hypothesis are refined, resulting in a refined sentence $\boldsymbol{y}^*$. We simply use a weighted sum method to achieve the final score:

$$\text{BARTScore++} = -(\text{Dist}_{\text{exp}}\omega_{\text{exp}} + \text{Dist}_{\text{imp}}\omega_{\text{imp}}),$$
$$\text{where } \omega_{\text{exp}} + \omega_{\text{imp}} = 1, \ \ 0 \leq \omega_{\text{exp}}, \omega_{\text{imp}} \leq 1$$

ω_{exp} and ω_{imp} weigh the importance of explicit and implicit errors respectively.[3] For easy to use, we define $\lambda = \omega_{\text{exp}}/\omega_{\text{imp}}$ as the **only** parameter, indicating the ratio of weights assigned to Explicit/ Implicit errors, where $\omega_{\text{exp}} = \frac{\lambda}{1+\lambda}, \omega_{\text{imp}} = \frac{1}{1+\lambda}$ respectively. Since λ may be different from task to task, we perform specific analysis in §6, confirming the stability when adjusting this parameter. We also provide guidance on selecting λ in Appendix B to help researchers use BARTScore++ for different tasks.

4 Experiment Setup

4.1 Tasks and Datasets

Tasks We follow Yuan et al. (2021) to consider three different tasks: summarization (SUM), machine translation (MT), and data-to-text (D2T).

Datasets for Translation We obtain the machine-translated texts and reference texts from the `WMT20` metrics shared task (Mathur et al., 2020b). We use the DARR corpus and consider 10 language pairs, which are `cs-en`, `de-en`, `ja-en`, `ru-en`, `zh-en`, `iu-en`, `km-en`, `pl-en`, `ps-en`, and `ta-en`. We also consider Multidimensional Quality Metric (MQM) for `zh-en` provided by Freitag et al. (2021a) in §6, comprising judgments of 8 best-performing translation systems in `WMT20`, annotated by professional translators.

Datasets for Summarization (1) `REALSumm` (Bhandari et al., 2020) is a meta-evaluation dataset for text summarization which measures pyramid-recall of each system-generated summary. (2) `SummEval` (Fabbri et al., 2021) is a collection of human judgments of model-generated summaries on the `CNNDM` dataset annotated by both expert judges and crowd-source workers. Each system-generated summary is gauged through the lens of coherence, factuality, fluency, and informativeness.

[3] Following the same pattern as in Yuan et al. (2021), we reverse the score to ensure BARTScore++ ranging from $-\infty$ to 0, with a higher score being a better quality of the sentence.

Metrics	High-Resource						Low-Resource					
	cs	de	ja	ru	zh	Avg.	iu	km	pl	ps	ta	Avg.
Supervised Baselines												
BLEURT	12.97	6.61	12.82	6.55	11.62	10.12	26.78	31.09	2.76	18.05	16.88	19.11
COMET	11.02	9.04	12.47	12.07	14.50	11.82	27.19	29.84	9.90	15.71	15.81	19.69
Unsupervised Baselines												
BLEU	3.90	-2.93	7.00	-3.47	6.39	2.18	15.41	22.72	-5.25	10.47	7.19	10.11
BERTScore	11.60	4.03	12.85	5.21	10.58	8.85	24.74	30.01	2.78	14.29	13.41	17.04
PRISM	12.42	2.67	13.46	7.22	11.65	9.48	25.37	30.44	5.70	**16.51**	14.78	18.56
BARTScore												
Vanilla BARTScore	11.81	5.55	13.62	9.22	13.12	10.66	26.93	32.27	7.64	15.54	16.63	19.80
+ Prompt	12.31	7.26	14.16	11.13	13.13	11.60	27.11	32.16	9.44	16.05	16.84	20.32
Ours - BARTScore++												
+ **Error Analysis**	12.06	7.23‡	15.08‡	9.98‡	13.32‡	11.54	27.37†	**32.38**†	8.44‡	15.94	17.09‡	20.24
+ Prompt + **Error Analysis**	**12.65**†	**8.75**‡	**15.40**‡	**11.76**‡	**13.35**‡	**12.38**	**27.60**‡	32.33†	**10.14**‡	16.40	**17.39**‡	**20.77**

Table 2: **Segment-level Kendall's τ correlation** (%) results on English-targeted language pairs of **WMT20 Metrics Shared Task** test set. **Bold** and Underlined values refer to the best result among unsupervised metrics and all metrics, respectively. † indicates BARTScore++ significantly outperforms BARTScore without error analysis, and ‡ indicates BARTScore++ further significantly outperform other unsupervised baselines.

(3) NeR18 The NEWSROOM dataset (Grusky et al., 2018) contains 60 articles with summaries generated by 7 different methods are annotated with human scores in terms of coherence, fluency, informativeness, relevance.

Datasets for Factuality (1) Rank19 (Falke et al., 2019) is used to meta-evaluate factuality metrics. It is a collection of 373 triples of a source sentence with two summary sentences, one correct and one incorrect. (2) QAGS20 (Wang et al., 2020) collected 235 test outputs on CNNDM dataset from Gehrmann et al. (2018) and 239 test outputs on XSUM dataset (Narayan et al., 2018) from BART fine-tuned on XSUM. Each summary sentence is annotated with correctness scores w.r.t. factuality.

Datasets for Data-to-Text We consider the following datasets which target utterance generation for spoken dialogue systems. (1) BAGEL (Mairesse et al., 2010) provides information about restaurants. (2) SFHOT (Wen et al., 2015) provides information about hotels in San Francisco. (3) SFRES (Wen et al., 2015) provides information about restaurants in San Francisco. They contain 202, 398, and 581 samples respectively, each sample consists of one meaning representation, multiple references, and utterances generated by different systems.

4.2 Baselines and Meta-evaluation

Baselines We compare our method with several commonly used baseline metrics for evaluating text generation, including BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2020b), MoverScore (Zhao et al., 2019) and PRISM (Thompson and Post, 2020). For MT tasks, we also consider supervised metrics that leverage human judgments to train, including COMET (Rei et al., 2020) and BLEURT (Sellam et al., 2020). For factuality evaluation on summarization tasks, we compare BARTScore++ with the best-performing factuality metrics FactCC (Kryscinski et al., 2020) and QAGS (Wang et al., 2020). We reproduce BARTScore and its variants using their official codes[4].

Meta-evaluation We follow Yuan et al. (2021) to conduct the meta-evaluation. Specifically, we apply *Kendall's τ* for MT tasks to measure the correlation of metrics with human evaluation[5]. For SUM and D2T tasks, we use *Spearman correlation* except for the Rank19 dataset, where *Accuracy* is used to measure the percentage of correct ranking between factual texts and non-factual texts. We adopt the paired bootstrap resampling method (Koehn, 2004) (p-value < 0.05) for significance tests.

4.3 Setup

As for the backbone BART, we use the same settings in BARTScore (Yuan et al., 2021) for specific

[4] https://github.com/neulab/BARTScore

[5] Since the meta-evaluation method is very sensitive to outliers (systems whose scores are far away from the rest of the systems) (Mathur et al., 2020a), we remove these outlier systems when computing correlations.

Metrics	REALSumm	SummEval				NeR18				
	COV	COH	FAC	FLU	INFO	COH	FLU	INFO	REL	Avg.
Baselines										
ROUGE	**49.75**	16.68	15.96	11.50	32.64	9.46	10.36	13.04	14.73	19.35
BERTScore	44.04	28.38	10.97	19.26	31.20	14.75	17.03	13.09	16.34	21.67
MoverScore	37.24	15.91	15.71	12.86	31.77	16.15	11.97	18.80	19.54	19.99
PRISM	41.10	24.88	34.52	25.36	21.16	57.28	53.20	56.13	55.34	41.00
BARTScore										
Vanilla BARTScore	47.42	**44.67**	38.11	35.64	35.53	67.89	67.00	64.67	60.51	51.27
+ Prompt	48.71	40.75	37.76	33.74	36.89	70.14	67.89	68.60	62.04	51.83
Ours - BARTScore++										
+ **Error Analysis**	47.76	**44.67**†	**38.48**†	**35.66**†	35.53†	68.62‡	67.79†	68.60‡	61.15‡	51.73
+ Prompt + **Error Analysis**	49.00	40.83†	38.08†	33.88†	**37.01**†	**70.44**‡	**68.75**‡	**69.66**‡	**63.04**‡	**52.30**

Table 3: **Spearman correlation** (%) results on three **text summarization datasets**. The best results are **Bold**. † and ‡ indicate BARTScore++ significantly outperforms all baselines and BARTScore without error analysis, respectively.

tasks, including BART-large, BART-CNN (fine-tuned on `CNNDM`) and BART-CNN-PARA (further fine-tuned on `ParaBank2`). We perform the same prompting strategy as in BARTScore (Yuan et al., 2021). Detailed settings are in Appendix A. In correct stage of error analysis, we set $k = 10$ when applying the top-k sampling, namely, a total of 10 tokens are obtained in $\mathcal{W}$ during each iteration.

5 Experimental Results

Metrics	Rank19	Q-CNN	Q-XSUM
	Acc.(%)	Pearson(%)	
Baselines			
ROUGE	63.00	45.91	9.70
BERTScore	71.31	57.60	2.38
MoverScore	71.31	41.41	5.41
PRISM	78.02	47.87	2.50
Factuality Metrics			
FactCC	70.00	-	-
QAGS	71.20	54.50	17.50
Human	83.90	-	-
BARTScore			
Vanilla BARTScore	83.65	73.47	18.38
+ Prompt	79.62	71.85	9.40
Ours - BARTScore++			
+ **Error Analysis**	**84.18**†	**73.97**‡	**19.33**‡
+ Prompt + **Error Analysis**	80.70‡	72.60‡	10.55

Table 4: **Results on Factuality Datasets**, where "Q" is short for QAGS.

Machine Translation Table 2 shows segment-level Kendall τ correlation of metrics on `WMT20`. We can observe that BARTScore++ can achieve state-of-the-art performance on all language pairs (most significantly outperform vanilla BARTScore except `ps-en`). The average correlation of BARTScore++ can surpass all supervised and unsupervised metrics by a large margin in both high-resource and low-resource scenarios (except `ps-en`). This confirms our intuition that with analysis of explicit/ implicit errors, BARTScore++ will agree more with human evaluations compared with vanilla BARTScore.

Regarding the prompting strategy, we also observe that 1) our proposed error analysis mechanism in BARTScore++ can achieve a similar amount of correlation improvement as that of prompting, and 2) incorporating both prompting and error analysis can further push SOTA results, confirming the orthogonality of error analysis and prompting strategies upon BARTScore.

Text Summarization Results on `REALSumm`, `SummEval` and `NeR18` are showed in Table 3. We observe that: 1) BARTScore++ surpasses all other metrics including BARTScore variants for all test settings except `REALSumm`. In most aspects, our purposed method can significantly outperform baseline metrics, and especially in `NeR18`, BARTScore++ can even significantly improve the performance of vanilla BARTScore. This further confirms the robustness (Rony et al., 2022) of our proposed metric. 2) Compared with prompting, er-

Metrics	BAGEL	SFRES	SFHOT	Avg.
Baselines				
ROUGE	23.43	11.57	11.75	15.58
BERTScore	28.91	15.64	13.54	19.36
MoverScore	28.37	15.27	17.23	20.29
PRISM	30.49	15.47	19.64	21.87
BARTScore				
Vanilla BARTScore	31.89	19.52	21.65	24.35
+ Prompt	33.28	23.74	23.81	26.94
Ours - BARTScore++				
+ **Error Analysis**	32.67†	19.74†	25.63‡	26.00
+ Prompt + **Error Analysis**	**34.12**‡	**23.99**‡	**26.04**‡	**28.02**

Table 5: **Spearman correlation** (%) of different metrics over three **Data-to-Text datasets**.

ror analysis mechanism in BARTScore++ on summarization tasks can also achieve a similar amount of correlation improvement, which again testify the importance of considering errors in summarization evaluation.

Analysis on factuality datasets As shown in Table 4, we also observe that BARTScore++ significantly outperforms other metrics on all three datasets. Strikingly, BARTScore++ can even surpass human baseline on `Rank19`. While prompting is not working in these tasks, error analysis mechanism corporated in BARTScore++ can also show significant improvement. This suggests that BARTScore++ is more effective in detecting the hallucination content and yielding more distinguishable scores in factuality summaries, which further confirms the universality of our proposed method.

Data-to-Text Results on data-to-text are shown in Table 5. We see that BARTScore++ can again surpass existing methods and significantly outperform vanilla BARTScore. We further find weights on explicit errors are consistently larger than implicit errors, interestingly suggesting we should focus more on explicit errors for data-to-text tasks.

6 Analysis

To better understand the mechanism by which BARTScore++ achieves promising results, we take a closer look and answer four questions:

Q1: How reliable is our BARTScore++ when evaluating top-performing systems?

Q2: How do explicit/ implicit error weights influence the accuracy of BARTScore++?

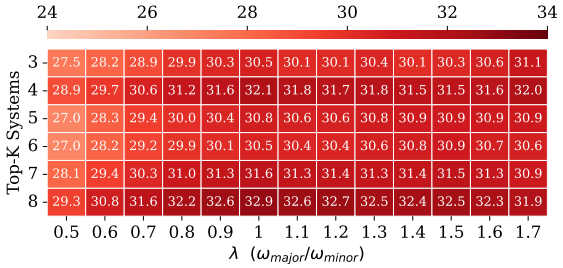

Figure 2: **Kendall correlation** (%) of BARTScore++ with MQM human evaluation dataset on top-k MT systems ranging **Error Weights Ratio** λ from 0.5-1.7.

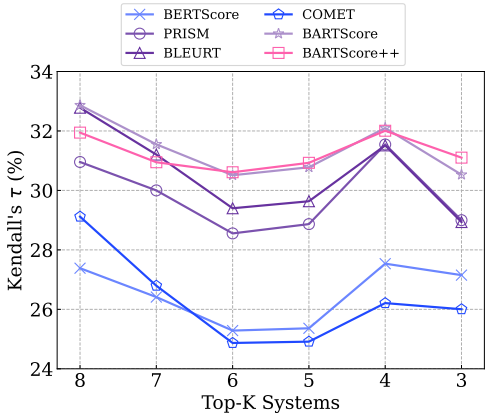

Figure 3: **Kendall correlation** (%) of different metrics on **Top-K MT systems according to MQM human evaluation dataset.**

Q3: How does error analysis make BARTScore++ more human-like?

Q4: Does error analysis framework introduce significant latency?

For MT evaluation in this section, we use MQM, an error-based evaluation framework annotated by human experts (Freitag et al., 2021a). For a fair comparison, the error weight ratio λ for `WMT20 zh-en` test set is fixed to 1.7.

BARTScore++ is Reliable When Evaluating Top-K Systems Previous studies have shown that most metrics are unreliable for evaluating best-performing systems, showing a sharp degradation of correlation with human evaluation (Mathur et al., 2020a). To answer **Q1**, we assess our method shown in Figure 3 with several baseline metrics on Top-K MT systems by computing Kendall's τ respectively. As seen, BARTScore++ can further improve BARTScore's performance, especially when evaluating top-performing systems ($K < 6$). This verifies the reliability of our purposed method.

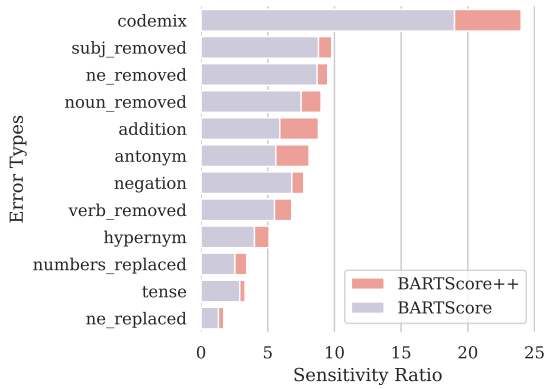

Figure 4: **Sensitivity Analysis on Major Errors** between BARTScore++ and vanilla BARTScore. The higher the sensitivity ratio, the greater the score difference attributed to this major error obtained by the metric.

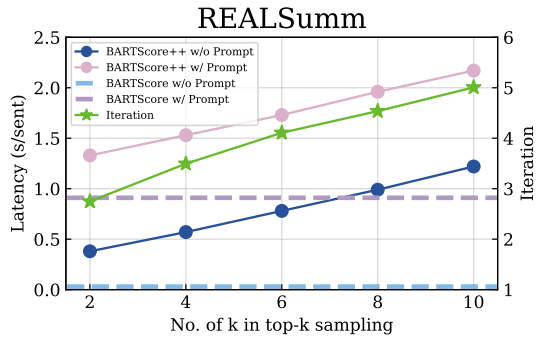

Figure 5: **Time Efficiency Analysis** on Summarization task (REALSumm). We report the average latency for evaluation and the average number of iterations w.r.t. top-k sampling in CORRECT stage in §3.2.

BARTScore++ is Stable When Adjusting Error Weights To answer **Q2**, we present an analysis on adjusting the error weight ratio λ in BARTScore++, which is the **only** parameter that needs to consider before evaluation. In Figure 2, as the number of systems K decreases, the ratio of error weights according to the best-performing BARTScore++ is fluctuating from 1 to 1.7. This suggests that different weights of importance should be given to explicit errors according to the overall qualities of MT systems. We also provide guidance on selecting this parameter in Appendix B to help researchers apply BARTScore++ to different task settings.

BARTScore++ is More Human-Like on Discriminating Errors To answer **Q3**, we perform a human analysis and show some cases in Appendix C to further show the advantage of our error analysis strategies incorporated in BARTScore++. In Table 9, we can see that human evaluators consistently assign low MQM scores to explicit errors (e.g. mistranslation of "delivery" in WeChat AI in example 1, mistranslation of "disc" in Tencent Translation in example 3), but BARTScore produces contrary judgments, ignoring these errors that should be punished strictly. Through our proposed error analysis, BARTScore++ becomes more discriminative on explicit errors and reaches an agreement with human judgments, while BARTScore fails to such errors. To better quantify such discriminative property, we report the sensitivity of our method on major errors using a perturbation dataset DEMETR[6] (Karpinska et al., 2022) in Figure 4, where BARTScore++ shows consistent boosts, confirming our claim.

BARTScore++ Brings Acceptable Latency A possible concern is the evaluation efficiency for BARTScore++, since top-k sampling and iterative inferences in error analysis inevitably introduce more complexity. We compare the latency between vanilla and ours on Nvidia A100 GPU with identical batchsize. As seen in Figure 5, 1) although increasing the k in sampling brings better performance, it inevitably increases the iterations and inference cost, and 2) well-performed BARTScore is used combining the Prompt strategy, which naturally owns high latency, i.e. 0.91 seconds per sentence, which is actually on the same order of magnitude as ours, i.e. 1.33∼2.17. Considering both the significant performance boosts and comparable latency, we believe the increased costs are totally acceptable.

7 Related Work

Automatic Metrics Automatic Evaluation Metrics are of crucial importance to the development of NLG systems, including translation (Koehn and Knowles, 2017; Ding et al., 2021; Zan et al., 2022b; Peng et al., 2023; He et al., 2023), summarization (Zhong et al., 2022b; Zan et al., 2022a), grammar error correction (Wu et al., 2023; Liu et al., 2021), dialogue generation (Li et al., 2017; Cao et al., 2021). Recent research has shown great success in language model-based metrics (Zhang et al., 2020b; Marie et al., 2021; Zhou et al., 2020; Rei et al., 2020; Sellam et al., 2020), which can significantly outperform traditional string-based metrics such as BLEU (Papineni et al., 2002). For example, BERTScore (Zhang et al., 2020b) and MoverScore

[6]Details of DEMETR analysis are shown in Appendix D.

(Zhao et al., 2019) leverage contextual embeddings to measure semantic distance between reference and hypothesis. COMET (Rei et al., 2020) and BLEURT (Sellam et al., 2020) rely on human evaluations to train. UniEval (Zhong et al., 2022a) reframes NLG evlauation into a Question Answering task and allows the metric to focus on different aspects. In this paper, we choose BARTScore (Yuan et al., 2021) as the testbed because of its SOTA performance and universality on NLG tasks. Note that our error analysis strategies can also be extended to other metrics, such as PRISM (Thompson and Post, 2020).

Human Evaluation Human evaluation, such as Direct Assessment (Graham et al., 2017), are often served as "golden standard". However, there is increasing evidence that inadequate evaluation will lead to wrong decisions (Toral, 2020). This motivates elaborate evaluation proposals (Popović, 2020; Gladkoff and Han, 2021) and MQM is one of these methodologies, grounded in explicit error analysis (Freitag et al., 2021a). In this work, We extend error analysis strategies to BARTScore, making it trigger more human-like judgments.

Error Analysis Existing automatic metrics tend to simplify the error detection procedure, such as edit distance in TER (Snover et al., 2006) and mismatch in BERTScore (Zhang et al., 2020b). To incorporate errors into automatic evaluation, recent research (Xu et al., 2022) simulates different errors and assigns scores like MQM as the training data to finetune a model-based metric. However, it does not address the issue of metrics lacking interpretability. In this work, we leverage the token-level judgments in BARTScore and analyze explicit errors through error analysis, making metrics more human-like, and providing more accurate evaluations. Our error analysis framework functionalizes like token-level quality estimation (Specia et al., 2021) or automatic post-editing (Freitag et al., 2019). With the reference signal provided, our proposed method is more accurate and universal for NLG evaluation.

8 Conclusion

We present an automatic metric BARTScore++ for NLG evaluation. Inspired by the advanced human evaluation MQM, BARTScore++ incorporates error analysis strategies to give a comprehensive score considering explicit and implicit errors. Experimental results show our approach achieves competitive results on a broad range of tasks. Our work is an early step toward human-like evaluation for automatic metrics, and we hope our BARTScore++ can motivate researchers working on NLG evaluation to focus more on human evaluation procedures such as error analysis.

Limitations

Limitations of BARTScore++ are three-fold:

- In §3.1, we propose Explicit/ Implicit errors to better distinguish different types of errors in generated texts. However, explicit errors only contain token-level errors that can be detected and corrected by error analysis, not involving all error types mentioned in MQM (e.g. severe fluency errors). We hope future studies can take these situations into account.
- In §3.2 we can see that our proposed error analysis framework fully relies on the generation probabilities of BART to decide how to refine the hypothesis. Still, we see that this framework may lead to false judgments due to unfaithful content. Further research can explore how to calibrate the pre-trained models during error analysis.
- In §3.3 we integrate the distance of explicit and implicit errors by simply computing their weighted sum. This can be improved by considering more factors, e.g. the overall quality of the generated text, refining iterations, and external signals. We will leave the exploration of combining these factors and designing better weighting schemes as future work.

Ethics Statement

We take ethical considerations very seriously, and strictly adhere to the ACL Ethics Policy. All procedures performed in this study are in accordance with the ethical standards. This paper focuses on improving automatic NLG evaluations with an error analysis framework. Our proposed metric relies on reference translations as signals and produces scores for translations indicating their quality. Both the datasets and models used in this paper are publicly available and have been widely adopted by researchers. Our model will not learn from user inputs or cause potential risks to the NLP community. We ensure that the findings and conclusions of

this paper are reported accurately and objectively. Informed consent was obtained from all individual participants included in this study.

Acknowledgement

We are grateful to the anonymous reviewers and the area chair for their insightful comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant 61973083, in part by the Natural Science Foundation of Shenzhen under Grant JCYJ20210324121213036. Derek F. Wong was supported in part by the Science and Technology Development Fund, Macau SAR (Grant Nos. FDCT/060/2022/AFJ, FDCT/0070/2022/AMJ) and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST).

References

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *EMNLP*.

Yu Cao, Liang Ding, Zhiliang Tian, and Meng Fang. 2021. Towards efficiently diversifying dialogue generation via embedding augmentation. In *ICASSP*.

Julian Chow, Lucia Specia, and Pranava Madhyastha. 2019. WMDO: Fluency-based word mover's distance for machine translation evaluation. In *WMT*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *TACL*.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *ACL*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In *WMT*.

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. *TACL*.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *EMNLP*.

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 *WMT*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *EMNLP*.

Serge Gladkoff and Lifeng Han. 2021. Hope: A task-oriented and human-centric evaluation framework using professional post-editing towards more effective mt evaluation. *arXiv preprint*.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *NAACL*.

Zhiwei He, Ti Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2023. Exploring human-like translation strategy with large language models. *arXiv preprint*.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. Demetr: Diagnosing evaluation metrics for translation. *arXiv preprint*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *WNMT*.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*.

Daniel Licht, Cynthia Gao, Janice Lam, Francisco Guzman, Mona Diab, and Philipp Koehn. 2022. Consistent human evaluation of machine translation across language pairs. *arXiv preprint*.

Xuebo Liu, Longyue Wang, Derek F Wong, Liang Ding, Lidia S Chao, and Zhaopeng Tu. 2021. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In *ICLR*.

Qingyu Lu, Baopu Qiu, Liang Ding, Liping Xie, and Dacheng Tao. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *WMT*.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *ACL*.

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

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *ACL*.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *WMT*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation. *arXiv preprint*.

Maja Popović. 2020. Informative manual evaluation of machine translation output. In *COLING*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *EMNLP*.

Md Rashad Al Hasan Rony, Liubov Kovriguina, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann. 2022. RoMe: A robust metric for evaluating natural language generation. In *ACL*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *ACL*.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA: Technical Papers*.

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 *WMT*.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *EMNLP*.

Antonio Toral. 2020. Reassessing claims of human parity and super-human performance in machine translation at WMT 2019. In *EAMT*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *ACL*.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *EMNLP*.

Rongxiang Weng, Heng Yu, Xiangpeng Wei, and Weihua Luo. 2020. Towards enhancing faithfulness for neural machine translation. In *EMNLP*.

Hao Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael R. Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *arXiv preprint*.

Wenda Xu, Yi-Lin Tuan, Yujie Lu, Michael Saxon, Lei Li, and William Yang Wang. 2022. Not all errors are equal: Learning text generation metrics using stratified error synthesis. In *EMNLP*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *NeurIPS*.

Changtong Zan, Liang Ding, Li Shen, Yu Cao, Weifeng Liu, and Dacheng Tao. 2022a. Bridging cross-lingual gaps during leveraging the multilingual sequence-to-sequence pretraining for text generation. *arXiv preprint*.

Changtong Zan, Keqin Peng, Liang Ding, Baopu Qiu, Boan Liu, Shwai He, Qingyu Lu, Zhenghang Zhang, Chuang Liu, Weifeng Liu, Yibing Zhan, and Dacheng Tao. 2022b. Vega-mt: The jd explore academy machine translation system for wmt22. In *WMT*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020a. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *ICLR*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *EMNLP*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022a. Towards a unified multi-dimensional evaluator for text generation. In *EMNLP*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2022b. E2s2: Encoding-enhanced sequence-to-sequence pretraining for language understanding and generation. *arXiv preprint*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of ACL*.

Lei Zhou, Liang Ding, and Koichi Takeda. 2020. Zero-shot translation quality estimation with explicit cross-lingual patterns. In *WMT*.

A Variants of Vanilla BARTScore

BARTScore Variants We summarize variants of BARTScore in Table 6. $\mathcal{F}$ score is applied for Machine Translation and Data-to-Text tasks; recall-based BARTScore is applied in `REALSumm` due to recall-based pyramid human evaluation; BARTScore on faithfulness is applied to other summarization tasks. In our experiments, we follow the same settings as in BARTScore (Yuan et al., 2021).

Variants	Computation using BARTScore
$\mathcal{F}$ score	$(\text{BARTScore}_{\boldsymbol{r}\to\boldsymbol{h}} + \text{BARTScore}_{\boldsymbol{h}\to\boldsymbol{r}})/2$
Recall	$\text{BARTScore}_{\boldsymbol{h}\to\boldsymbol{r}}$
Faithfulness	$\text{BARTScore}_{\boldsymbol{s}\to\boldsymbol{h}}$

Table 6: BARTScore variants and their computation methods. The source, reference sentence and hypothesis are denoted as $\boldsymbol{s}$, $\boldsymbol{r}$, $\boldsymbol{h}$ respectively.

Prompt Design Prompting is a parameter-free method to elicit more accurate results by combining texts with a set of short phrases (prompts). BARTScore applies this method through two basic approaches: suffixing prompts on the encoder or prefixing prompts on the decoder of BART (Lewis et al., 2020). If multiple prompts are provided, the final BARTScore of a hypothesis is computed by averaging the score of all its generation scores using different prompts. When vanilla BARTScore is used in our method, we perform the same prompting strategy as in BARTScore (Yuan et al., 2021).

B Guidance on Selecting Error Weights Ratio λ

Since error weights ratio λ is the only parameter that may differ from task to task, so we provide two suggestions on selecting it:

- Inspired by the idea of the Calibration Set from Licht et al. (2022), we suggest creating a relatively smaller test set and then collecting human evaluations on them. The test size should include over 100 samples covering various ranges of translation quality. To ensure the reliability of human evaluations, we recommended recruiting 2 to 3 professional evaluators to label the Calibration Set according to the MQM annotating procedure (Freitag et al., 2021a). Choose the error weights ratio relating to the highest consistency with human judgments.
- When evaluating the datasets mentioned in this paper, we provide settings of λ in Table 7 in BARTScore++ for researchers to apply directly.

C Case Study

We show four evaluation examples of machine translation in Table 9 to further explain how error analysis makes BARTScore++ more human-like. These examples are from `WMT20` test set on three best-performing systems, Huoshan Translation, WeChat AI, and Tencent Translation. For all examples, judgments of BARTScore++ are agree with MQM (marked in **Better** and **Worse**), but contrary to vanilla BARTScore.

Example 1 The worse hypothesis generated by WeChat AI translates "投运" into "delivery" (highlighted in yellow). However, vanilla BARTScore seems to "ignore" this error and give a higher score than the better translation from Huoshan Translation. BARTScore++ applies an error analysis and gives a more discriminative evaluation by revising this word to "opening". In this way, $\textbf{Dist}_{\textbf{exp}}$ are enlarged by a larger error weight (0.000 -> 0.348), resulting in an agreement with human judgment.

Example 2 WeChat AI produces a major error when translating "更缺" into "even more". This error is detected through the error analysis mechanism and the mistranslation word "more" is deleted for its awkward style. Such deletion helps

Task	Dataset	Language Pair / Aspect	λ
MT	WMT20	cs-en	0.80
		de-en	0.40
		ja-en	0.50
		ru-en	1.70
		zh-en	1.10
		iu-en	0.95
		km-en	1.30
		pl-en	0.85
		ps-en	1.10
		ta-en	0.60
SUM	REALSumm	COV	0.95
	SummEval	COH	1.00
		FAC	0.75
		FLU	1.40
		INFO	0.95
	NeR18	COH	1.10
		FLU	0.75
		INFO	0.70
		REL	0.70
	Rank19	FAC	0.85
	QAGS-CNN	FAC	1.00
	QAGS-XSUM	FAC	0.90
D2T	BAGEL	-	2.00
	SFRES	-	1.40
	SFHOT	-	4.90

Table 7: Selection of **Error Weight Ratio** λ for all test settings in BARTScore++.

BARTScore++ to better distinguish the quality between these two sentence.

Example 3 Although vanilla BARTScore gives similar scores to both translations, their MQM scores are significantly different (11.333 vs 6.333), mainly because of the translation on "umbilical cord tray". Tencent Translation mistranslates it into "disc", which is detected and corrected through error analysis, leading to a relatively low score for BARTScore++. This example also shows that error analysis can help metrics better evaluate long sentences.

Example 4 Huoshan Translate produces a mistranslation error "recognized" when translating the verb "承认". We can see that such error is detected and revised to "admitted", resulting in a relatively large explicit distance (0.000 compared with 0.258), confirming that BARTScore++ can better distinguish major errors and become more human-like.

D Sensitivity analysis on BARTScore++ using DEMETR

To better quantify the sensitivity of BARTScore++ on different kinds of explicit errors, we utilize a metric diagnosing dataset, `DEMETR` (Karpinska et al., 2022), perturbing on 1000 test samples with different types of errors. We use the ratio proposed in `DEMETR` to measure the sensitivity of a metric, denoted as:

$$z = \frac{\text{SCORE}(\boldsymbol{r}, \boldsymbol{h}) - \text{SCORE}(\boldsymbol{r}, \boldsymbol{h'})}{\text{SCORE}(\boldsymbol{r}, \boldsymbol{h}) - \text{SCORE}(\boldsymbol{r}, [\text{empty}])}$$

where $\boldsymbol{r}, \boldsymbol{h}, \boldsymbol{h'}$ and $[\text{empty}]$ represent the reference, hypothesis, the perturbed hypothesis and empty string respectively. We calculate this ratio for each test sample and average them as the sensitivity for each error type. Figure 4 shows the sensitivity of BARTScore++ and BARTScore on different types of errors. We can see that: Compared with vanilla BARTScore, BARTScore++ is consistently more sensitive to major errors, confirming our claim.

E Influence of Different References when using BARTScore++

One potential concern is that the evaluation of BARTScore++ may heavily rely on the reference, which could make this metric less robust compared to the original BARTScore when switching to a different reference. We compare the performance of BARTScore and BARTScore++ on the top-5 systems from `WMT20 zh-en`, using two different references labeled as Ref.A and Ref.B. The results are presented in Table 8.

Reference	BARTScore	BARTScore++	Δ
Ref.A	31.06	31.27	+0.21
Ref.B	31.66	31.83	+0.17

Table 8: Kendall's τ correlation (%) on two different references (Ref.A and Ref.B) from top-5 MT systems in `WMT20 zh-en`.

As seen, BARTScore++ is not significantly affected by the choice of reference, as consistent improvements observed (+0.21/+0.17). However, the performance of vanilla BARTScore appears to be less robust than BARTScore++ (-0.60 from Ref.B to Ref.A). This further validates the effectiveness and robustness of our method on different references.

Example 1: #239				
Source	9月25日，北京大兴国际机场投运仪式隆重举行。			
Reference	On September 25th, a grand opening ceremony was held for the Beijing Daxing International Airport.			
	Huoshan Translation (**Better**)		WeChat AI (**Worse**)	
Translation	On September 25, the commissioning ceremony of Beijing Daxing International Airport was held ceremoniously.		On September 25, the delivery ceremony of Beijing Daxing International Airport was held.	
Refined Sentence	On September 25, the commissioning ceremony of Beijing Daxing International Airport was held ceremoniously.		On September 25, the opening ceremony of Beijing Daxing International Airport was held.	
Scores & Error Distance	**BARTScore++** (BARTScore) **-0.306** (-1.543)	**$Dist_{exp}$ / $Dist_{imp}$** 0.000 / 0.827	**BARTScore++** (BARTScore) **-0.334** (-1.374)	**$Dist_{exp}$ / $Dist_{imp}$** 0.348 / 0.310
Example 2: #284				
Source	寿光缺企业，更缺企业家。			
Reference	Shouguang lacked enterprises, and even lacked entrepreneurs.			
	Huoshan Translation (**Better**)		WeChat AI (**Worse**)	
Translation	Shouguang lacks enterprises and entrepreneurs.		Shouguang lacks enterprises and even more entrepreneurs.	
Refined Sentence	Shouguang lacks enterprises and entrepreneurs.		Shouguang lacks enterprises and even entrepreneurs.	
Scores & Error Distance	**BARTScore++** (BARTScore) **-0.245** (-1.887)	**$Dist_{exp}$ / $Dist_{imp}$** 0.000 / 0.662	**BARTScore++** (BARTScore) **-0.281** (-1.821)	**$Dist_{exp}$ / $Dist_{imp}$** 0.231 / 0.365
Example 3: #319				
Source	... 刘艳艳拿着产包和脐带盘就往楼下冲。			
Reference	... Liu Yanyan grabbed the maternity package and umbilical cord tray rushed downstairs to them.			
	WeChat AI (**Better**)		Tencent Translation (**Worse**)	
Translation	... Liu Yanyan rushed downstairs with the delivery bag and umbilical cord plate.		... Liu Yanyan rushed downstairs with the delivery bag and umbilical cord disc.	
Refined Sentence	... Liu Yanyan rushed downstairs with the delivery bag and umbilical cord plate.		... Liu Yanyan rushed downstairs with the delivery bag and umbilical cord tray.	
Scores & Error Distance	**BARTScore++** (BARTScore) **-0.412** (-2.024)	**$Dist_{exp}$ / $Dist_{imp}$** 0.000 / 1.112	**BARTScore++** (BARTScore) **-0.437** (-1.998)	**$Dist_{exp}$ / $Dist_{imp}$** 0.133 / 0.953
Example 4: #750				
Source	... 任何正派的雇主，都不会以本案中承认的极其不公平和敷衍的方式来解雇员工。			
Reference	... no employer with any sense of common decency, would have effected a dismissal in the hopelessly unfair and perfunctory manner admitted to in this case.			
	Tencent Translation (**Better**)		Huoshan Translate (**Worse**)	
Translation	... no decent employer will fire employees in the extremely unfair and perfunctory manner admitted in this case.		... no decent employer will dismiss an employee in the extremely unfair and perfunctory manner recognized in this case.	
Refined Sentence	... no decent employer will fire employees in the extremely unfair and perfunctory manner admitted in this case.		... no employer would dismiss an employee in the hopelessly unfair and perfunctory manner admitted in this case.	
Scores & Error Distance	**BARTScore++** (BARTScore) **-0.351** (-2.087)	**$Dist_{exp}$ / $Dist_{imp}$** 0.000 / 0.947	**BARTScore++** (BARTScore) **-0.415** (-2.079)	**$Dist_{exp}$ / $Dist_{imp}$** 0.258 / 0.681

Table 9: Four examples from WMT20 zh-en test dataset with a disagreement between BARTScore and BARTScore++. Words detected and corrected by BARTScore++ are highlighted. We can see that BARTScore++ can benefit from the distances of explicit error and implicit error, achieving more reliable evaluations.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 9: "Limitation"

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 2, 3, 4

☑ B1. Did you cite the creators of artifacts you used?
Section 2, 3, 4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We use the dataset (described in Section 4) and code framework (transformers) which are publicly available. Also, we cite the creators of them.

☑ 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)?
Section 10: "Ethics Statement", we use publicly available and widely-used datasets/code framework.

☒ 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?
Section 10: "Ethics Statement", we use publicly available and widely-used datasets.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4

☑ 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.
We include these relevant statistics described in Section 4.

C ☑ Did you run computational experiments?

Section 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5, 6

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4, 5; Appendix B

☑ 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?
Section 5, 6

☑ 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.)?
Section 4, 5, 6

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.