Revisiting Grammatical Error Correction Evaluation and Beyond

Peiyuan Gong[1] **Xuebo Liu**[2*] **Heyan Huang**[1] **Min Zhang**[2]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
{3120201019,hhy63}@bit.edu.cn
[2]Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China
{liuxuebo,zhangmin2021}@hit.edu.cn

Abstract

Pretraining-based (PT-based) automatic evaluation metrics (e.g., BERTScore and BARTScore) have been widely used in several sentence generation tasks (e.g., machine translation and text summarization) due to their better correlation with human judgments over traditional overlap-based methods. Although PT-based methods have become the de facto standard for training grammatical error correction (GEC) systems, GEC evaluation still does not benefit from pretrained knowledge. This paper takes the first step towards understanding and improving GEC evaluation with pretraining. We first find that arbitrarily applying PT-based metrics to GEC evaluation brings unsatisfactory correlation results because of the excessive attention to inessential systems outputs (e.g., unchanged parts). To alleviate the limitation, we propose a novel GEC evaluation metric to achieve the best of both worlds, namely PT-M^2, which only uses PT-based metrics to score those corrected parts. Experimental results on the CoNLL14 evaluation task show that PT-M^2 significantly outperforms existing methods, achieving a new state-of-the-art result of 0.949 Pearson correlation. Further analysis reveals that PT-M^2 is robust to evaluate competitive GEC systems. Source code and scripts are freely available at https://github.com/pygongnlp/PT-M2.

1 Introduction

Grammatical error correction (GEC) is the task that takes a sentence with grammatical errors as input, and outputs a corrected sentence. Due to the important role of GEC in the field of second language learning and intelligent writing, GEC has attracted wide attention from the community (Chollampatt and Ng, 2018a; Lewis et al., 2020; Omelianchuk et al., 2020). As a typical natural language generation task (NLG), the common practice for GEC evaluation is to calculate the similarity between system outputs and their corresponding references (Dahlmeier and Ng, 2012; Napoles et al., 2015; Bryant et al., 2017).

Task	**Training**	**Evaluation**
Machine Translation	●	●
Text Summarization	●	●
Grammatical Error Correction	●	○

Table 1: The use of PT-based models for model training and evaluation in various NLG tasks. ● means used and ○ means unused. Unlike other tasks, GEC has not used PT-based methods for evaluation.

With the rapid development of pretraining (PT) (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020), several NLG tasks, such as machine translation, text summarization and GEC, have been utilizing PT-based models to improve the model training process (Kaneko et al., 2020; Omelianchuk et al., 2020; Tarnavskyi et al., 2022), as shown in Table 1. Furthermore, since PT-based models can learn rich syntactic and semantic knowledge from a large amount of unlabeled data, they are able to calculate the similarity between two sentences more accurately. Therefore, many mainstream NLG tasks try to develop new evaluation metrics based on PT-based models, and the new metrics have been sufficiently validated in terms of the high consistency with human judgments (Zhang et al., 2020; Yuan et al., 2021). However, existing GEC systems still use traditional metrics for evaluation.

In this paper, to find the reason why GEC systems do not use PT-based metrics for evaluation, we revisit existing GEC evaluation, comparing the traditional overlap-based evaluation metrics with recently popular PT-based metrics. Surprisingly, our preliminary experiment on the CoNLL14 evaluation task shows that arbitrarily applying PT-based metrics to GEC evaluation results in a relatively worse correlation to human judgments than the traditional metrics. Further analysis reveals that the

[*] Corresponding author

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6891–6902
December 7-11, 2022 ©2022 Association for Computational Linguistics

scoring strategies of PT-based metrics are questionable for GEC evaluation. GEC is a local substitution task that only partially changes from the source sentences, but PT-based metrics have to compute the score of the whole sentence despite most words staying the same after correction. The scores from the unchanged words bias the final sentence score, leading to an unreliable evaluation for GEC.

To alleviate the above limitation, in this paper we propose PT-M^2, a novel PT-based GEC metric that combines the advantages of both the PT-based metrics and traditional metrics. Unlike the PT-based metrics scoring a whole sentence, PT-M^2 only uses PT-based metrics to score the changed words that can be extracted by the M^2 metric (Dahlmeier and Ng, 2012). Experimental results on the CoNLL14 evaluation task show that PT-M^2 achieves the highest correlation compared to traditional metrics based on two kinds of system ranking methods, either calculating score at the corpus- or sentence-level. We also show that PT-M^2 does not heavily rely on the scale of PT-based models, even equipped with a small-scale model can obtain satisfactory results.

Our main contributions are listed as follows:

- We revisit GEC evaluation and find that the arbitrary use of PT-based metrics results in poor correlation with human judgments. We analyze the corresponding reasons in terms of scoring strategies for different metrics.

- We propose a novel PT-based GEC metric PT-M^2, which only uses PT-based metrics to score the correction words. PT-M^2 achieves a new state-of-the-art of 0.949 Pearson correlation on the CoNLL14 evaluation task.

- We find that PT-M^2 still performs well in evaluating high-performing GEC systems, which is helpful for promoting GEC research and upgrading the GEC system in the future.

2 Related Work

2.1 Overlap-based GEC Metrics

As a growing number of high-performance GEC models are proposed (Liu et al., 2021; Li et al., 2022; Zhang et al., 2022), it is important to provide interpretable and reliable evaluation metrics to measure their quality. Several overlap-based GEC metrics have been provided to evaluate GEC models. As shown in Table 2, despite employing PT-based models for model training, recently published works still use overlap-based GEC metrics (e.g., M^2 (Dahlmeier and Ng, 2012), GLEU (Napoles et al., 2015), and ERRANT (Bryant et al., 2017)) for GEC evaluation.

Latest GEC Work	**M^2**	**ERRANT**	**GLEU**
Omelianchuk et al. (2020)	●	●	○
Sun and Wang (2022)	●	●	○
Lai et al. (2022)	●	●	○
Tarnavskyi et al. (2022)	●	●	○
Stahlberg and Kumar (2020)	●	●	●
Kaneko et al. (2020)	●	●	●
Katsumata and Komachi (2020)	●	●	●
Parnow et al. (2021)	●	●	●

Table 2: GEC metrics which are used in the latest GEC works. All of the works evaluate with traditional GEC metrics, despite using PT-based models for training.

Dale and Kilgarriff (2011) first align source sentences and hypothesis sentences using the Levenshtein algorithm for GEC evaluation. Dahlmeier and Ng (2012) convert each source-hypothesis pair to an edit sequence dynamically, extracting the corresponding edits and using the F_1 measure to represent the score of each system. Similar to (Dahlmeier and Ng, 2012), Felice and Briscoe (2015) evaluate corrections at the token level, leveraging a globally optimal alignment algorithm on source-hypothesis pairs and source-reference pairs respectively. Napoles et al. (2015) propose a variant of BLEU that rewards n-grams appearing in hypothesis sentences and reference sentences but not in source sentences, penalizing n-grams in source sentences and hypothesis sentences but not in reference sentences and averaging scores over different references. Bryant et al. (2017) employ a linguistically-enhanced Damerau-Levenshtein algorithm to align sentence pairs, merging parts of the alignment and automatically classifying each edit with pre-defined rules. Choshen et al. (2020) propose a multilingual variant of ERRANT, extracting and classifying edits with universal dependencies. Gotou et al. (2020) focus on difficulty for the model to correct different types of errors and use several GEC models to compute the difficulty weight for each edit. However, a natural weakness is that traditional overlap-based metrics cannot capture the similarity between semantically similar words, limiting their effectiveness.

2.2 PT-based NLG Metrics

PT-based methods are not only used in the training stage of NLG tasks but also in designing evaluation metrics to assess the performance of NLG models, which can overcome the above core weakness of overlap-based metrics. Several recently proposed PT-based metrics have dominated a large number of NLG evaluation tasks. Lo (2019) use cross-lingual PT-based models to encode source sentences and hypothesis sentences and compute the cosine distances with a semantic parser. Zhao et al. (2020) compute scores for source-hypothesis pairs with cross-lingual PT-based models, re-aligning the vector space and using a language model to score fluency. Belouadi and Eger (2022) develop a fully unsupervised evaluation metric that leverages pseudo-parallel pairs obtained from fully unsupervised evaluation metrics and use pseudo reference sentences from unsupervised translation systems. Song et al. (2021) introduce a metric combined by BERTScore, Word Mover's Distance and sentence semantic similarity. Gekhman et al. (2020) extract the entities from source sentences and hypothesis sentences via a multilingual knowledge base respectively and measure the recall of two entitie sets. Yoshimura et al. (2020) train a BERT-based regression model to optimize multiple sub-metrics on the GEC evaluation dataset. Islam and Magnani (2021) leverage GPT2 (Radford et al., 2018) to measure the grammaticality for the GEC outputs.

Zhang et al. (2020) compute the similarity score for each token in hypothesis sentences with each token in reference sentences and use greedy matching to maximize each similarity score. Zhao et al. (2019) combine contextualized embeddings with Word Mover's Distance to soft align tokens from hypothesis sentences to reference sentences. Yuan et al. (2021) introduce that a high-quality hypothesis will be easily generated based on source sentence or reference sentence or vice-versa, and use the generation probability to evaluate system outputs. Sellam et al. (2020) generate a large number of synthetic hypothesis-reference sentence pairs and pretrain BERT on several supervision signals with a parameterized regression layer to help the model generalization. Rei et al. (2020) propose two evaluation frameworks that predict the quality score and minimize the distance between the "better" hypothesis sentence and the corresponding reference sentence respectively. Zhan et al. (2021b) evaluate different PT-based NLG metrics for machine translation tasks. Benefiting from PT-based metrics, the correlation with human judgments has significantly improved on NLG evaluation. In this work, we would like to investigate the effectiveness of PT-based metrics on GEC evaluation.

3 Revisiting Existing Metrics

This section revisits existing overlap-based GEC metrics and PT-based metrics on the GEC evaluation task, computing the metric correlation with human judgments, analyzing experimental results and exploring the differences among the metrics.

3.1 Experimental Setup

Dataset Settings We conduct experiments on the CoNLL14 evaluation task (Grundkiewicz et al., 2015). There are 1,312 source sentences, and each source sentence corresponds to two standard reference sentences. Twelve teams have provided their system outputs and the source sentences are used as the thirteenth system. Eight annotators judge the quality of hypothesis sentences generated by corresponding GEC systems. Each hypothesis sentence is scored from 1 to 5, which means from worst to best. Two system ranking lists are respectively generated by Expected -Wins (EW) and Trueskill (TS) algorithms.

Experiment Settings The method to estimate the performance of GEC evaluation metrics is measuring the degree of correlation with human judgments. Following (Grundkiewicz et al., 2015), we measure the correlation between metrics and human judgments based on the **system-level ranking**, computing Pearson γ and Spearman ρ respectively as the final correlations. We compute the score for each system in two settings: **corpus-level** and **sentence-level**. Given the metric M, source sentences $\mathbf{S}$, hypothesis sentences $\mathbf{H}$ and reference sentences $\mathbf{R}$, the first setting computes the system score based on the whole corpus $\mathrm{M}(\mathbf{S}, \mathbf{H}, \mathbf{R})$, and the last one uses the average of the sentence-level scores $\sum_i^I \mathrm{M}(\mathbf{S}_i, \mathbf{H}_i, \mathbf{R}_i)/I$.

Evaluation Metrics We compare the following overlap-based GEC metrics and PT-based metrics:

- **GLEU** rewards hypothesis n-grams that match reference sentences but not source sentences and penalizes hypothesis n-grams that match source sentences but not reference sentences (Napoles et al., 2015).

System	Sentence	Rank	BERTScore
SRC	They play the important role in our life which can not be **substituted** .	-	-
REF	They play **an** important role in our life which can not be **substituted** .	-	-
AMU	They play **an (0.99)** important role in our life which can not be replaced (0.75) .	1	0.94
UFC	They play the (0.63) important role in our life which can not be **substituted (0.99)** .	2	**0.95**

Table 3: Example from the CoNLL14 evaluation task. “**Red Bold**” denotes the right correction whereas “Blue Non-bolded” denotes the wrong one. Although BERTScore scores higher for the correct edit of the AMU system, the final overall score is worse than that of the UFC system.

- **M^2** aligns source sentences and hypothesis sentences with Levenshtein algorithm, dynamically choosing the alignment that maximally matches the gold edits and extracting the system edits. (Dahlmeier and Ng, 2012). F$_{0.5}$ is used as the system score.

- **SentM^2** is a variant of M^2, using the average of F$_{0.5}$ scores computed at the sentence-level as the system score.

- **ERRANT** aligns sentence pairs with a linguistic-enhanced Damerau-Levenshtein algorithm and uses two kinds of rules to merge alignment, extract and classify edits (Bryant et al., 2017). F$_{0.5}$ is used as the system score.

- **SentERRANT** is a variant of ERRANT, using the average of F$_{0.5}$ scores computed at the sentence-level as the system score.

- **BERTScore** is a PT-based metric, which computes the token similarity between the sentence pairs and uses greedy matching to maximize the matching similarity score (Zhang et al., 2020).[1]

- **BARTScore** is a PT-based metric, which converts the evaluation task to a sequence generation task and uses the generation probability to estimate the quality of system outputs (Yuan et al., 2021).[2]

3.2 Preliminary Results

PT-based Metrics Fail The results of the existing metrics are shown in Table 4. The advantages are different between GLEU and M^2 as GLEU is better than M^2 on Pearson and worse on Spearman (Chollampatt and Ng, 2018b). We introduce ERRANT to compute its correlation with human judgments since ERRANT can extract and classify edits automatically, which has been used as the standard metric in GEC (Bryant et al., 2017). Without using the gold edits annotated by humans, ERRANT can still get a relatively high correlation. We also find that computing the system score at the sentence-level for M^2 and ERRANT can get a higher correlation than computing at the corpus-level, which is the same as in (Napoles et al., 2016).

Metric	**EW**		**TS**	
	γ	ρ	γ	ρ
GLEU	0.701	0.467	0.750	0.555
ERRANT	0.642	0.659	0.688	0.698
SentERRANT	0.870	**0.742**	0.846	0.747
M^2	0.623	0.687	0.672	0.720
SentM^2	**0.871**	0.731	**0.864**	**0.758**
BARTScore	0.172	0.253	0.173	0.269
BERTScore	0.262	0.074	0.166	-0.022

Table 4: Correlations of metrics with human judgments on the CoNLL14 evaluation task. PT-based metrics such as BERTScore and BARTScore perform relatively worse than the overlap-based GEC metrics.

Besides, we observe that even though the PT-based metrics such as BERTScore and BARTScore have dominated in the evaluation of multiple NLG tasks (Scialom and Hill, 2021), they correlate much lower than overlap-based GEC metrics on the GEC evaluation task, whether for Pearson or Spearman, and are even negatively correlated with human judgments. So why are PT-based metrics not suitable to evaluate GEC systems? What are the differences between the PT-based metrics and overlap-based GEC metrics?

Discussion This part aims to answer the above questions. As shown in Table 3, to figure out why PT-based metrics are not suitable for GEC evaluation, we analyze a representative from the

[1] We use the reference sentences from the CoNLL14 evaluation task as the corpus to compute the IDF weight.

[2] We use the reference sentence as the input and generate the corresponding hypothesis sentence.

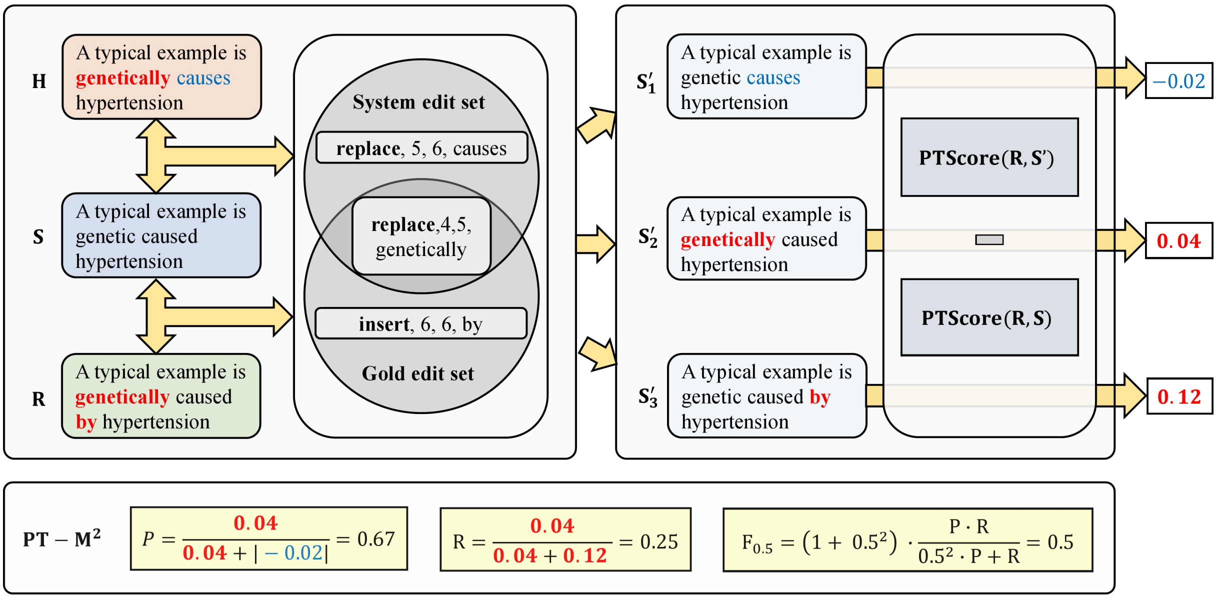
H
A typical example is genetically causes hypertension
S
A typical example is genetic caused hypertension
R
A typical example is genetically caused by hypertension
System edit set
replace, 5, 6, causes
replace,4,5, genetically
insert, 6, 6, by
Gold edit set
S'_1
A typical example is genetic causes hypertension
S'_2
A typical example is genetically caused hypertension
S'_3
A typical example is genetic caused by hypertension
PTScore(R, S')
PTScore(R, S)
−0.02
0.04
0.12
PT − M²
$P = \frac{0.04}{0.04 + |-0.02|} = 0.67$
$R = \frac{0.04}{0.04 + 0.12} = 0.25$
$F_{0.5} = (1 + 0.5^2) \cdot \frac{P \cdot R}{0.5^2 \cdot P + R} = 0.5$

Figure 1: Overview of our approach **PT-M**^{2}, **S**, **H**, and **R** denote the source sentence, the hypothesis sentence, and the reference sentence, respectively. There are three core modules in our approach: 1) Extract two edit sets from corresponding sentence pairs; 2) Compute the score for each edit in the edit set with PT-based metrics; 3) Apply edit scores as corresponding edit weights on the overlap-based GEC metrics M^2. "**Red Bold**" denotes the right corrections and their corresponding scores whereas "Blue Non-bolded" denotes the wrong corrections and their corresponding scores.

CoNLL14 evaluation task. For the two edits provided by the AMU system, even though the change of the correct edit score is larger than that of the wrong correction edit, the AMU system score provided by BERTScore is lower than that of the UFC system, which leads to the result being different with human annotation. A possible reason is that BERTScore computes scores for all the tokens in the sentence, a large percentage of the score for the unchanged part affects the trend of the overall score. When an edit is applied, not only the corresponding edit score is changed, but the scores of surrounding tokens are also affected, causing BERTScore to fail to make a correct assessment. The excessive attention to unchanged words potentially biases the final score of PT-based metrics.

4 PT-M^2: The Best of Both Worlds

4.1 Motivation

Based on the above findings, to inject pretrained knowledge into GEC evaluation metrics, we propose PT-M^2, which takes advantage of both PT-based metrics and overlap-based GEC metrics. PT-M^2 computes edit scores with PT-based metrics. Without directly using PT-based metrics to score hypothesis-reference sentence pairs, we use them at the edit-level to compute a score for each edit. As shown in Figure 1, we first extract the system edit set and gold edit set from the source-hypothesis sentence pair and source-reference sentence pair respectively. We then compute the score of each edit with PT-based metrics as the scorer. Lastly we apply edit scores as the edit weights on the M^2.

4.2 Implementation

Edit Extraction Given a source sentence S, a hypothesis sentence H and a reference sentence R, the first step we have to do is to extract the system edit set E and gold edit set G from the source-hypothesis sentence pair (S, H) and source-reference sentence pair (S, R) respectively (Dahlmeier and Ng, 2012; Bryant et al., 2017). As shown in Figure 1, each edit is composed of edit operation, start index, end index and correct tokens. We use the edit extraction module from M^2 to extract both two edit sets.[3] The intersection of the system edit set and gold edit set represents all of the correct edits provided by corresponding system.

Edit Score Without treating each edit equally, we propose a novel method to score each edit, employing PT-based metrics (e.g., BERTScore and

[3] Due to the fact that the CoNLL14 evaluation task has provided the gold edit set, we only need to extract the system edit set by M^2.

Type	Metric	PT Model	EW (Corpus)		TS (Corpus)		EW (Sentence)		TS (Sentence)	
			γ	ρ	γ	ρ	γ	ρ	γ	ρ
Overlap	GLEU	-	0.701	0.467	0.750	0.555	0.784	0.720	0.828	0.775
	ERRANT	-	0.642	0.659	0.688	0.698	0.870	0.742	0.846	0.747
	M^2	-	0.623	0.687	0.672	0.720	0.871	0.731	0.864	0.758
PT	BARTScore	BART	-	-	-	-	0.172	0.253	0.173	0.269
	BERTScore	BERT	-	-	-	-	0.262	0.074	0.166	-0.022
Ours	PT-ERRANT	BART	0.681	**0.797**	0.727	**0.841**	0.905	0.786	0.897	0.824
	PT-ERRANT	BERT	**0.705**	0.780	**0.745**	0.797	0.941	0.879	0.917	0.846
	PT-M^2	BART	0.667	0.775	0.715	0.813	0.898	0.813	0.901	0.841
	PT-M^2	BERT	0.693	0.758	0.737	0.769	**0.949**	**0.907**	**0.938**	**0.874**

Table 5: Correlations of metrics with human judgments on the CoNLL14 evaluation task. Our proposed PT-M^2 and PT-ERRANT methods correlate better with human judgments on both the corpus- and sentence-level. We highlight the **highest** score in bold and the second-highest score with underlines.

BARTScore) as the edit scorer (PTScore). As shown in Figure 1, we first build an edit set U, which is the union of the system edit set E and gold edit set G. Then, we compute the score of each edit u in the edit set U and obtain an edit score set W, in which w_u is the score of the edit u. To achieve this goal, the first step we applied is to use each edit in the edit set U to generate a partially correct version S' of the source sentence S. Then we use PTScore to compute the scores of the sentence pair (S, R) and (S', R) respectively, the more similarity between the sentence pair which input to PTScore, the higher the score that PTScore gives. The score differences are used to measure whether the edit is beneficial or not:

$$w = \text{PTScore}(S', R) - \text{PTScore}(S, R) \quad (1)$$

$w > 0$ means the edit is helpful for correction, which shows it is a correct edit, otherwise is a wrong correction edit. We use the absolute value $|w|$ as the edit score. The larger the $|w|$ is, the edit has a greater impact on the sentence, whether beneficial or harmful.

Final Score As shown in Figure 1, given the system edit set E, the gold edit set G, and the edit score set W, we treat each edit score as the corresponding edit weight and apply edit weight on each edit to compute precision, recall and $F_{0.5}$ measure respectively:

$$P = \frac{\sum_{c \in E \cap G} w_c}{\sum_{e \in E} w_e} \quad (2)$$

$$R = \frac{\sum_{c \in E \cap G} w_c}{\sum_{g \in G} w_g} \quad (3)$$

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad (4)$$

A β value of 0.5 is used in PT-M^2 follows vanilla M^2. To compute the system score, we compute $F_{0.5}$ based on the whole corpus to represent the corpus-level PT-M^2 and use the average of $F_{0.5}$ score for each sentence to represent the sentence-level PT-M^2 (Napoles et al., 2016). Besides, since our approach is transparent to the type of edits, it can be directly applied to other overlap-based GEC metrics, such as ERRANT.

5 Experiment

PT-M^2 can combine the advantages of both PT-based metrics and overlap-based GEC metrics. In PT-M^2, PT-based metrics only compute edit scores that can reduce the impact of unchanged part on the final score, paying more attention to the changes in the source sentences and editing scores are applied as corresponding edit weights on the M^2.

To demonstrate the effectiveness of our approach, we use PT-M^2 to evaluate GEC system outputs and compute the correlation with human judgments on the CoNLL14 evaluation task (Grundkiewicz et al., 2015). We choose Pearson γ and Spearman ρ to measure the correlations. We use PT-M^2 to compute both the system score at corpus- and sentence-level (Napoles et al., 2016). We adopt two well-known unsupervised PT-based metrics BERTScore and BARTScore to score edits, which can be used in multiple domains, tasks and languages (Scialom and Hill, 2021). The other metrics have been introduced in Section 3.1.

Rank	Metric				Human	
	BERTScore	M^2	SentM^2	PT-M^2	EW	TS
1	INPUT (⇑9)	CAMB (⇑1)	CUUI (⇑3)	**AMU (✓0)**	AMU	AMU
2	UFC (⇑4)	CUUI (⇑2)	**AMU (⇓1)**	CUUI (⇑2)	RAC	CAMB
3	IITB (⇑6)	AMU (⇓2)	**CAMB (✓0)**	**CAMB (✓0)**	CAMB	RAC
4	**RAC (⇓1)**	**POST (⇑1)**	**POST (⇑1)**	**RAC (⇓1)**	CUUI	CUUI
5	SJTU (⇑5)	NTHU (⇑7)	NTHU (⇑7)	**POST (✓0)**	POST	POST
6	**PKU (✓0)**	RAC (⇓3)	RAC (⇓3)	**PKU (✓0)**	UFC	PKU
7	AMU (⇓6)	**UMC (✓0)**	**PKU (✓0)**	UFC (⇓1)	PKU	UMC
8	POST (⇓3)	PKU (⇓1)	**UMC (✓0)**	SJTU (⇑2)	UMC	UFC
9	CUUI (⇓5)	SJTU (⇑1)	SJTU (⇑1)	**IITB (✓0)**	IITB	IITB
10	**UMC (⇓2)**	**UFC (⇓2)**	**UFC (⇓2)**	**NTHU (⇑2)**	SJTU	INPUT
11	IPN (⇑2)	IPN (⇑2)	IITB (⇓2)	**INPUT (✓0)**	INPUT	SJTU
12	CAMB (⇓9)	IITB (⇓3)	**INPUT (⇓1)**	UMC (⇓4)	NTHU	NTHU
13	NTHU (⇓1)	INPUT (⇓2)	**IPN (✓0)**	**IPN (✓0)**	IPN	IPN
Δ	53	27	21	**12**	-	-

Table 6: System rankings by different metrics. ⇑/⇓ denotes that the rank given by the evaluation metric is higher/lower than human judgments, and ✓ denotes that the given rank is equal to human ranking. The lowest rank difference in each rank is highlighted in **bold**. PT-M^2 successfully ranks the best system that the other metrics fail. Besides, it also shows the lowest rank difference (Δ).

5.1 Main Results

Table 5 reports the correlations of overlap-based GEC metrics, PT-based metrics, and our PT-based GEC metric PT-M^2. PT-M^2 aligns better with human judgments compared to traditional overlap-based GEC metrics, either computed at corpus- or sentence-level (Napoles et al., 2016). Besides, for the two human rankings proposed by EW and TS, PT-M^2 makes a significant improvement on both Pearson and Spearman correlations. Although BERTScore and BARTScore are computed in different ways (Sai et al., 2022), we find that our approach uses either of them as the edit scorer can get a similarly high correlation.

We also test a variant of our approach, namely PT-ERRANT, with the same operations as in PT-M^2. Compared to ERRANT, PT-ERRANT gets higher correlation with human judgments. We find that comparing PT-M^2 and PT-ERRANT, PT-M^2 correlates better at the sentence-level while PT-ERRANT aligns better at the corpus-level. We also test different values of β when computing F_β as the final score. The results show that PT-M^2 correlates more stable and is consistently better than M^2. It is worth mentioning that the sentence-level PT-M^2 with BERTScore as the scorer gets the highest correlation among all of the metrics we have experimented with. Therefore, we treat it as the main version of PT-M^2 and use the sentence-level PT-M^2 in the subsequent experiments.

Models	Size	EW		TS	
		γ	ρ	γ	ρ
BART	Small	0.883	0.743	0.869	0.740
	Base	0.898	0.813	0.901	0.841
	Large	0.899	0.813	0.906	0.841
BERT	Small	0.953	0.923	0.941	0.890
	Base	0.949	0.907	0.938	0.874
	Large	0.945	0.918	0.931	0.879

Table 7: Comparison between different PT-based models. There are no significant differences between models of different sizes.

5.2 Analysis

Effect of PT-based models To demonstrate the generality and generalizability of our approach, we use different sizes of PT-based metrics as the edit scorer to verify if the correlation of our approach is stable. As shown in Table 7, we experiment with PT-M^2 at the sentence-level, employing BERTScore and BARTScore in three different sizes, from small to large respectively. Even if we use the distilled version of BERTScore (Sanh et al., 2019), PT-M^2 also aligns much better with human

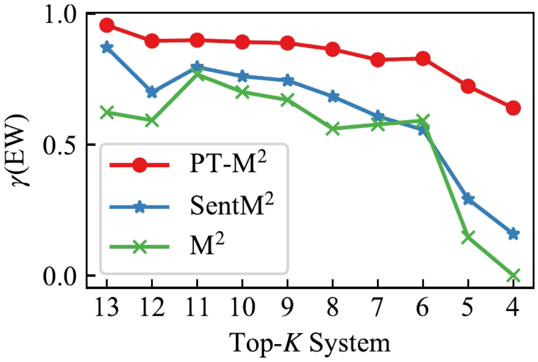

Figure 2: Pearson scores of Top-K system based on the EW ranking list. PT-M^2 is highly correlated with human judgments especially when all the systems are competitive (i.e., $K \leq 6$).

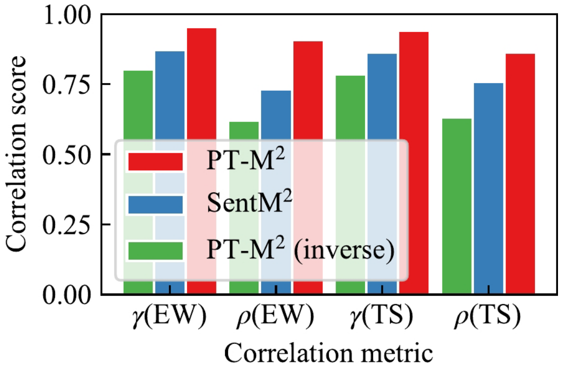

Figure 3: Effectiveness of the proposed edit score. Reversing the edit score harms the correlation.

judgments than M^2 and gets the similar correlation with the base and even the large size of models. So we can use the smaller size of PT-based metrics to accelerate the computation of GEC evaluation, with a trivial performance drop.

System Ranking Table 6 presents the system ranking results. Compared to other metrics, PT-M^2 shows the lowest rank differences with human judgments. PT-M^2 successfully ranks the best system (AMU) while the other metrics fail. Compared to BERTScore, PT-M^2 successfully ranks the INPUT system. Meanwhile, PT-M^2 gets a relatively better ranking result for the NTHU system than M^2 and SentM^2. This confirms the effectiveness of PT-M^2.

Effect of Top-K Systems To demonstrate the effectiveness of our approach, we evaluate high-performing systems with our approach to observe the change of correlations among different metrics (Zhan et al., 2021a). Figure 2 compares the Pearson correlation of the top-K systems. When K decreases, the correlations of M^2 and SentM^2 are lower than before correspondingly. Meanwhile, we find that the correlation of our approach PT-M^2 computed at the sentence-level drops much slower than M^2 and SentM^2. Further, when K is lower than 6, the Pearson correlations of M^2 and SentM^2 drop sharply while PT-M^2 does not act the same way under the circumstances. More specifically, as the system count drops from 13 to 4, the Pearson correlations of M^2 and SentM^2 down three times or even more. In contrast, the decline of PT-M^2 correlation is not obvious, which demonstrates the effectiveness of applying edit scores computed by PT-based metrics as corresponding edit weights on the overlap-based GEC metrics.

Effect of Edit Score We carry out a set of comparative experiments to demonstrate the necessity of computing edit weights. In our approach PT-M^2, we use PT-based metrics such as BERTScore and BARTScore to compute edit scores. The more important the edit, the higher score we provide. In the comparative experiments, we use the inverse of each edit score computed by PT-based metrics as the corresponding edit weight, $w = 1/|\text{PTScore}(S', R) - \text{PTScore}(S, R)|$, to demonstrate the importance of edit weights on overlap-based GEC metrics. For convenience, we use PT-M^2 (inverse) as the above comparative approach.

Figure 3 the comparison between the Pearson and Spearman correlations of the three metrics. After inversing edit weights computed by PT-based metrics, both the correlations measured by Pearson and Spearman are lower than SentM^2 for a large. Based on the above results, we demonstrate the effectiveness and advantages of employing PT-based metrics to directly compute edit weights.

A Case Study As shown in Table 8, to explain why our approach PT-M^2 correlates better with human judgments than M^2, we present an example from the CoNLL14 evaluation task (Grundkiewicz et al., 2015). The human ranking shows that the NTHU system score is higher than that of the PKU system. M^2 provides the wrong ranking result while PT-M^2 aligns the same as human judgments. In this example, PT-M^2 measures the gold edit only provided by the NTHU system as the highest score, which means it is much more important to predict this edit. Meanwhile, PT-M^2 sets the wrong correction edit supplied by the NTHU system with a lower score, which influences the source sentence a little. PT-M^2 successfully scores higher for the NTHU system.

System	**Sentence**	**Rank**	**M^2**	**PT-M^2**
SRC	It is also entire incorrect to fault social media alone for **the lack** of interpersonal skill .	-	-	
REF	It is also **entirely incorrect** to fault social media alone for **the lack** of interpersonal **skills** .	-	-	
NTHU	It is also **entirely incorrect** to fault social media alone for lack of interpersonal **skills** .	1	0.71	**0.88**
PKU	It is also entire incorrect to fault social media alone for **the lack** of interpersonal **skills** .	2	**0.83**	0.44

Table 8: Example from the CoNLL14 evaluation task. "**Red Bold**" denotes the right correction whereas "Blue Non-bolded" denotes the wrong one. The ranking of PT-M^2 is more in line with human judgments.

6 Conclusion and Future Work

In this work, we revisit GEC metrics and explore the feasibility of employing PT-based metrics to evaluate GEC systems. Compared to overlap-based GEC metrics, PT-based metrics correlate worse with human judgments. The reason we find is that PT-based metrics compute score for each token in ungrammatical sentences, however, the scores of the unchanged part account for a large proportion of the final score, leading to an inaccurate evaluation result. To leverage pretrained knowledge in GEC evaluation, we propose a novel PT-based GEC metric PT-M^2, employing PT-based metrics to score edits extracted from sentence pairs and applying edit scores as corresponding edit weights on the M^2. Experiments demonstrate that PT-M^2 gets the highest correlation on the CoNLL14 evaluation task, achieving a new state-of-the-art result of 0.949 Pearson correlation. Besides, further analysis shows that PT-M^2 is competent to evaluate high-performing GEC systems.

In the future, we would like to test the effectiveness of PT-M^2 in the GEC evaluation tasks of other languages (e.g., Chinese). It is also worthwhile to explore the benefits of PT-M^2 as a signal for reinforcement learning to train better GEC systems.

Limitations

Our PT-based GEC metric PT-M^2 has got the highest correlation with human judgments on GEC evaluation tasks. A reason that might limit the widespread use of PT-based metrics is that the calculation speed is slower than that of the traditional overlap-based GEC metrics. As PT-M^2 uses PT-based metrics to score each edit, it takes nearly 2 minutes to calculate a system score based on a single NVIDIA GTX 3080TI card, which is slower than the calculation speed of the traditional M^2 score that costs nearly 10 seconds. Therefore, there is still room for improvement in calculation speed, and we will continue developing parallel computing to speed it up in the future.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62206076) and Shenzhen College Stability Support Plan (Grant No. GXWD20220811173340003 and GXWD20220817123150002). The authors would like to thank the anonymous reviewers for their insightful suggestions.

References

Jonas Belouadi and Steffen Eger. 2022. Uscore: An effective approach to fully unsupervised evaluation metrics for machine translation. *arXiv preprint arXiv:2202.10062*.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018a. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5755–5762. AAAI Press.

Shamil Chollampatt and Hwee Tou Ng. 2018b. A reassessment of reference-based grammatical error correction metrics. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2730–2741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. Classifying syntactic errors in learner language. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107, Online. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Pro-*

ceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado. Association for Computational Linguistics.

Zorik Gekhman, Roee Aharoni, Genady Beryozkin, Markus Freitag, and Wolfgang Macherey. 2020. Kobe: Knowledge-based machine translation evaluation. *arXiv preprint arXiv:2009.11027*.

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal. Association for Computational Linguistics.

Md Asadul Islam and Enrico Magnani. 2021. Is this the end of the gold standard? a straightforward reference-less grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using pretrained encoder-decoder model. *arXiv preprint arXiv:2005.11849*.

Shaopeng Lai, Qingyu Zhou, Jiali Zeng, Zhongli Li, Chao Li, Yunbo Cao, and Jinsong Su. 2022. Type-driven multi-turn corrections for grammatical error correction. *arXiv preprint arXiv:2203.09136*.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bei Li, Quan Du, Tao Zhou, Yi Jing, Shuhan Zhou, Xin Zeng, Tong Xiao, JingBo Zhu, Xuebo Liu, and Min Zhang. 2022. ODE transformer: An ordinary differential equation-inspired model for sequence generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8335–8351, Dublin, Ireland. Association for Computational Linguistics.

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 *International Conference on Learning Representations*.

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*.

Chi-kiu Lo. 2019. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's no comparison: Reference-less evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2115, Austin, Texas. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Kevin Parnow, Zuchao Li, and Hai Zhao. 2021. Grammatical error correction as GAN-like sequence labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3284–3290, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ananya B Sai, Akash Kumar Mohankumar, and Mitesh M Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Thomas Scialom and Felix Hill. 2021. Beametrics: A benchmark for language generation evaluation evaluation. *arXiv preprint arXiv:2110.09147*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

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

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

Xin Sun and Houfeng Wang. 2022. Adjusting the precision-recall trade-off with align-and-predict decoding for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 686–693.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. *arXiv preprint arXiv:2203.13064*.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Runzhe Zhan, Xuebo Liu, Derek F. Wong, and Lidia S. Chao. 2021a. Difficulty-aware machine translation evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 26–32, Online. Association for Computational Linguistics.

Runzhe Zhan, Xuebo Liu, Derek F. Wong, and Lidia S. Chao. 2021b. Variance-aware machine translation test sets. In *Thirty-fifth Conference on Neural Information Processing Systems, Datasets and Benchmarks Track*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zheng Zhang, Liang Ding, Dazhao Cheng, Xuebo Liu, Min Zhang, and Dacheng Tao. 2022. Bliss: Robust sequence-to-sequence learning via self-supervised input representation. *ArXiv*, abs/2204.07837.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. *arXiv preprint arXiv:2005.01196*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Algorithm 1: Calculation Procedure of PT-M^2

```
Data: source sentences $\mathbf{S}$, hypothesis sentences $\mathbf{H}$, reference sentences $\mathbf{R}$, gold edit sets $\mathbf{GOLD}$
Result: F-score $F_{0.5}$
Function ComputeScore($S$, $R$, $E$):
    initialize an empty dict $W$ ;
    for $i \leftarrow 1$ to Len($E$) do
        $e \leftarrow E_i$ ; // the $i$th edit in the edit set $E$
        $S' \leftarrow$ Correct($S$, $e$) ; // apply edit $e$ to correct the source sentence $S$
        $W_e \leftarrow$ |PTScore($S'$, $R$) - PTScore($S$, $R$)| ; // compute the difference of PTScore
            between ($S'$, $R$) and ($S$, $R$)
    end
    return $W$;
$F_{0.5} \leftarrow 0$ ; // initialize $F_{0.5}$
for $i \leftarrow 1$ to Len($\mathbf{S}$) do
    $f_{max} \leftarrow -1$ ; // initialize $f_{max}$
    $S, H, R, GOLD, \leftarrow \mathbf{S}_i, \mathbf{H}_i, \mathbf{R}_i, \mathbf{GOLD}_i$;
    for $j \leftarrow 1$ to Len($R$) do
        $G \leftarrow GOLD_j$ ; // the jth gold edit set of $S$
        $E \leftarrow$ ExtractEdit($S$, $H$, $G$); // extract system edit
        $C \leftarrow E \cap G$ ; // choose the correct edit that system provides
        $W \leftarrow$ ComputeScore($S$, $R$, $E \cup G$); // compute the score of each edit
        $p \leftarrow \sum_{c \in C} W_c / \sum_{e \in E} W_e$ ; // compute the precision score
        $r \leftarrow \sum_{c \in C} W_c / \sum_{g \in G} W_g$ ; // compute the recall score
        $f \leftarrow (1 + 0.5^2) \cdot p \cdot r / (0.5^2 \cdot p + r)$ ; // compute the f score
        $f_{max} \leftarrow$ Max($f_{max}$, $f$)
    end
    $F_{0.5} \leftarrow F_{0.5} + f_{max}$
end
$F_{0.5} \leftarrow F_{0.5}$ / Len($\mathbf{S}$)
```

A Appendix

A.1 Algorithm

Algorithm 1 illustrates the calculation procedure of our PT-based GEC metric PT-M^2, showing the whole process of how to compute the sentence-level PT-M^2.

A.2 Command-line Interface

We introduce how to compute PT-M^2 and its variant PT-ERRANT in different settings with our code:

```
python evaluate.py
    --base [m2|sentm2|errant|senterrant]
    --scorer [self|bertscore|bartscore]
    --source <src_file>
    --hypothesis <hyp_file>
    --reference <ref_file>
    --output <out_file>
```