IMPARA: Impact-Based Metric for GEC Using Parallel Data

Koki Maeda and **Masahiro Kaneko** and **Naoaki Okazaki**
Tokyo Institute of Technology
{koki.maeda@nlp., masahiro.kaneko@nlp, okazaki@}.c.titech.ac.jp

Abstract

Automatic evaluation of grammatical error correction (GEC) is essential in developing useful GEC systems. Existing methods for automatic evaluation require multiple reference sentences or manual scores. However, such resources are expensive, thereby hindering automatic evaluation for various domains and correction styles. This paper proposes an **I**mpact-based **M**etric for GEC using **PARA**llel data, IMPARA, which utilizes correction impacts computed by parallel data comprising pairs of grammatical/ungrammatical sentences. As parallel data is cheaper than manually assessing evaluation scores, IMPARA can reduce the cost of data creation for automatic evaluation. Correlations between IMPARA and human scores indicate that IMPARA is comparable or better than existing evaluation methods. Furthermore, we find that IMPARA can perform evaluations that fit different domains and correction styles trained on various parallel data.

1 Introduction

Grammatical error correction (GEC) is a task of correcting grammatically incorrect sentences (Yuan and Briscoe, 2016; Chollampatt and Ng, 2018a; Junczys-Dowmunt et al., 2018; Kaneko et al., 2020, 2022; Omelianchuk et al., 2020). A GEC system is designed to be applicable in various domains, such as web text (Flachs et al., 2020) and essays written by language learners (Yannakoudakis et al., 2011) and in different correction styles, such as minimal and fluency edits (Ng et al., 2013; Napoles et al., 2017; Hotate et al., 2019). Idealy, GEC models should be evaluated by manually assessing the quality of corrections made by these models for certain target domains and styles. However, it is expensive to perform manual evaluation every time a GEC model outputs a correction; we thus need to establish an automatic evaluation method that correlates well with manual assessments.

Method	Fine-tuning data	Domain dependence
Based on language models	None	No
With manual assessments	P and M	on P and M
This work (IMPARA)	P	on P

Table 1: Comparison of reference-less methods (P: parallel data; M: manual assessment data). As discussed in Section 3, the method that is based only on language models underperforms the others. IMPARA achieves a comparable or better performance than the method trained on manual assessments, although IMPARA does not depend on manual assessment data.

Automatic evaluation methods of GEC are categorized into two. The first category is reference-based methods (Dahlmeier and Ng, 2012; Napoles et al., 2015; Bryant et al., 2017) that evaluate the closeness of output sentences from a GEC system and human-written reference sentences. In general, an ungrammatical sentence can be corrected in different ways. Thus, reference-based methods require multiple reference sentences for the accurate evaluation. However, Choshen and Abend (2018b) argued that it is unrealistic to prepare reference sentences that cover all possible corrections. In addition, they showed that low-coverage reference sets deteriorate the reliability of evaluation.

The second category is reference-less method, which uses only input sentences and system outputs (see Table 1 for comparison). Several studies applied language models for automatic evaluation (Napoles et al., 2016; Flachs et al., 2020; Islam and Magnani, 2021). However these studies had low correlations with human judgements because the perplexity of language models does not necesarily reflect the grammaticality but the frequency of words. Therefore, Asano et al. (2017) and Yoshimura et al. (2020) proposed using human assessment scores for training automatic evaluation models and hence reported performance improvements. However, applying their methods to various domains and styles is expensive because

Proceedings of the 29th International Conference on Computational Linguistics, pages 3578–3588
October 12–17, 2022.

they require a dataset of human assessment for each domain/style and because the availability of such datasets is limited. In addition, it is difficult to create a reliable dataset for human assessment (Choshen and Abend, 2018a).

We propose a novel reference-less method called **I**mpact-based **M**etric for GEC using **PARA**llel data (**IMPARA**)[1]. This method can be trained using only parallel data comprising ungrammatical and grammatical sentence pairs, which are widely available in various domains and correction styles. Furthermore, creating parallel data is less expensive than manually rating GEC outputs. Therefore, we can use IMPARA for various domains/styles with much less effort than performing manual assessments.

The simplest way to build a model for correction quality judgement is learning to discriminate between original and grammatical sentences in parallel data. However, this approach has two problems. First, an automatic evaluation method receives a pair of original sentence and *GEC output*, whereas the parallel data only includes pairs of original and *grammatical* sentences. Therefore, the discriminator may receive a pair of two ungrammatical sentences during inference, although it is trained only with pairs of ungrammatical and grammatical sentences. Second, an automatic evaluation method must handle incomplete corrections made by GEC methods. Even if an original sentence includes multiple grammatical errors, a GEC method may correct some of the errors and leave others. Hence, the supervision data for training an automatic evaluation method should include instances where grammatical errors are partially corrected.

IMPARA addresses these issues by automatic generation of supervision data with partially-corrected sentences. Decomposing corrections between original and corrected sentences into *edits*, we measure the *impact* of each edit to determine the preferences of edits. Then, we generate pairs of partially corrected sentences from parallel data and determine the preference order of generated pairs based on the impacts of the involved edits. The evaluation model is trained on the generated pairs so that it reproduces the preference order of sentence pairs.

The meta-evaluation (Section 3) demonstrates that IMPARA achieves a comparable or better evaluation performance than existing reference-less methods, even without tailored data of human assessments, but only with parallel data. Furthermore, IMPARA exhibits high capability of adapting its evaluation metric to the target domain and style given by the parallel data.

2 IMPARA

2.1 Architecture

IMPARA comprises quality estimator (QE) and similarity estimator (SE) based on BERT (Devlin et al., 2019), as illustrated in Figure 1. Given a pair of original sentence and GEC output, the QE evaluates the quailty of the GEC output. Then, the SE computes the semantic similarity of two sentences. We use a pre-trained BERT model without fine-tuning to build the SE, but fine-tune the BERT model for the QE.

Let X and Y denote an original sentence and an output of GEC system, respectively. Given a GEC output Y, the QE yields a score $\mathrm{QE}(Y) \in [0, 1]$. Given X and Y, the SE computes the similarity $\mathrm{SE}(X, Y) \in [0, 1]$ of the sentences. Equation 1 defines the overall score of the correction from X to Y,

$$\mathrm{score}(X, Y) = \begin{cases} \mathrm{QE}(Y) & (\text{if } \mathrm{SE}(X, Y) > \theta) \\ 0 & (\text{otherwise}) \end{cases}. \quad (1)$$

Here, θ is a threshold to output the score of the QE; the score is $\mathrm{SE}(X, Y)$ if the semantic similarity is higher than θ and 0 otherwise. The threshold prevents the high overall score even when output Y is deviated from original sentence X.

This study aims to build a *relative* evaluation measure that can compare two sentences in terms of the quality of grammatical correction[2]. We are not interested in building an absolute measure, unlike other metrics, e.g., recall and precision used in M^2 (Dahlmeier and Ng, 2012). Adjusting $\mathrm{score}(X, Y)$ with manual assessment scores may be possible. However, we leave this as a future work because our focus is to learn an evaluation measure only from parallel data.

2.2 Quality estimator

The QE computes a score as a dot product between a parameter vector and contextualized embeddings from the BERT model, followed by the sigmoid

[1] https://github.com/Silviase/IMPARA

[2] Therefore, $\mathrm{score}(X, Y) = 0.99$ does not mean that 99% of errors in X are corrected in Y.

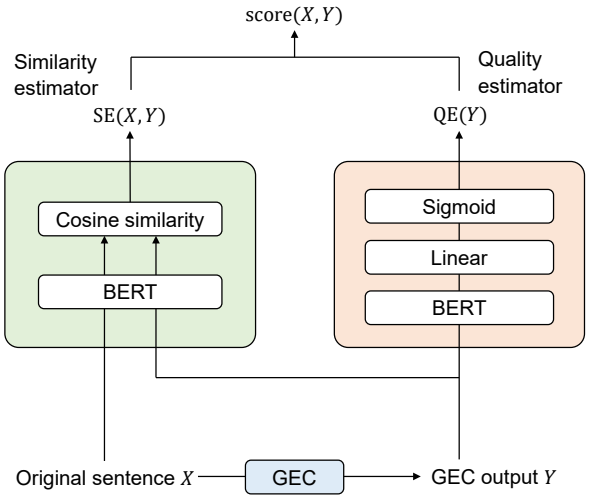

Figure 1: The IMPARA model architecture

function σ. Formally, let $\mathbf{y} \in \mathbb{R}^d$ denote the contextualized embeddings of the first token of the output sentence Y at the final layer of the BERT model (where d is the number of dimensions of embeddings). We define $q(Y) \in \mathbb{R}$ as a dot product between parameter vector $\mathbf{w} \in \mathbb{R}^d$ and embeddings $\mathbf{y}$ of sentence Y,

$$q(Y) = \mathbf{w}^\top \mathbf{y}. \tag{2}$$

Then we compute a QE score

$$\mathrm{QE}(Y) = \sigma(q(Y)). \tag{3}$$

We train the parameter vector $\mathbf{w}$ by fine-tuning the BERT model on the supervision data $\mathcal{T}$ (to be described in Section 2.4). Supervision data comprises pairs of positive S_+ and negative S_- sentences. We train the model so that it prefers positive sentences to negative ones by minimizing loss function $\mathfrak{L}$.

$$\mathfrak{L} = \frac{1}{|\mathcal{T}|} \sum_{(S_-, S_+) \in \mathcal{T}} \sigma\left(q(S_-) - q(S_+)\right) \tag{4}$$

Here, we use the sigmoid function $\sigma(.)$ to stabilize training[3].

2.3 Impact of edits

Before describing the procedure for generating the supervision data $\mathcal{T}$, we introduce the notion of an *edit* and its *impact* in grammatical error correction.

Let (S, T) be a pair of sentences from parallel data and $\mathcal{E}$ be a set of edits to obtain the corrected sentence T from the original one S. Here, edits are extracted automatically by using ERRANT (Bryant et al., 2017), a tool for aligning two sentences and extracting edits and their types.

For an edit $e \in \mathcal{E}$, T_{-e} denotes the sentence with all edits except for e, i.e., $\mathcal{E} \setminus \{e\}$, applied to S. In other words, T_{-e} presents the sentence where edit e is omitted when rewriting S into T. Hence, T is obtained by applying edit e to T_{-e}. We evaluate the impact of edit e in terms of the magnitude of the semantic change from T_{-e} to T,

$$\mathrm{impact}(T, e) = 1 - \frac{\mathrm{BERT}(T) \cdot \mathrm{BERT}(T_{-e})}{\|\mathrm{BERT}(T)\| \|\mathrm{BERT}(T_{-e})\|}. \tag{5}$$

Here, $\mathrm{BERT}(T) \in \mathbb{R}^d$ presents the mean pooling of contextualized embeddings of all tokens in sentence T at the final layer of the BERT model. Equation 5 provides a higher impact to an edit that greatly changes the meaning between T_{-e} and T. Given a set of edits E, we define its overall impact $I(T, E)$ as the sum of impacts of all edits,

$$I(T, E) = \sum_{e \in E} \mathrm{impact}(T, e). \tag{6}$$

Choshen and Abend (2018a) also proposed the concept of computing an impact of an edit. However, they define an impact as the number of applied edits without considering the impact of an individual edit. Conversely, we assume that a sentence with more semantically important corrections should receive a higher impact than others with semantically unimportant corrections.

2.4 Generating supervision data for QE

As detailed in Equation 4, the QE model requires supervision data $\mathcal{T}$ with pairs of positive and negative sentences. We cannot use parallel data of original and corrected sentences as they are because the QE model needs to measure the quality of imperfect corrections. In addition, in some parallel data with fluency corrections (e.g., JFLEG (Napoles et al., 2017)), the difference from an original to its corrected sentence is so large that the QE model may not learn the importance of each individual edit in parallel sentences.

Therefore, we generate partially corrected sentences from the sentence pairs of parallel data and determine the preference order between two sentences. Figure 2 depicts the generation process for a pair of the original S and corrected T sentences in the parallel data. First, we apply ERRANT (Bryant

[3]Preliminary experiments confirmed that the sigmoid function contirubted to improve the evaluation performance.

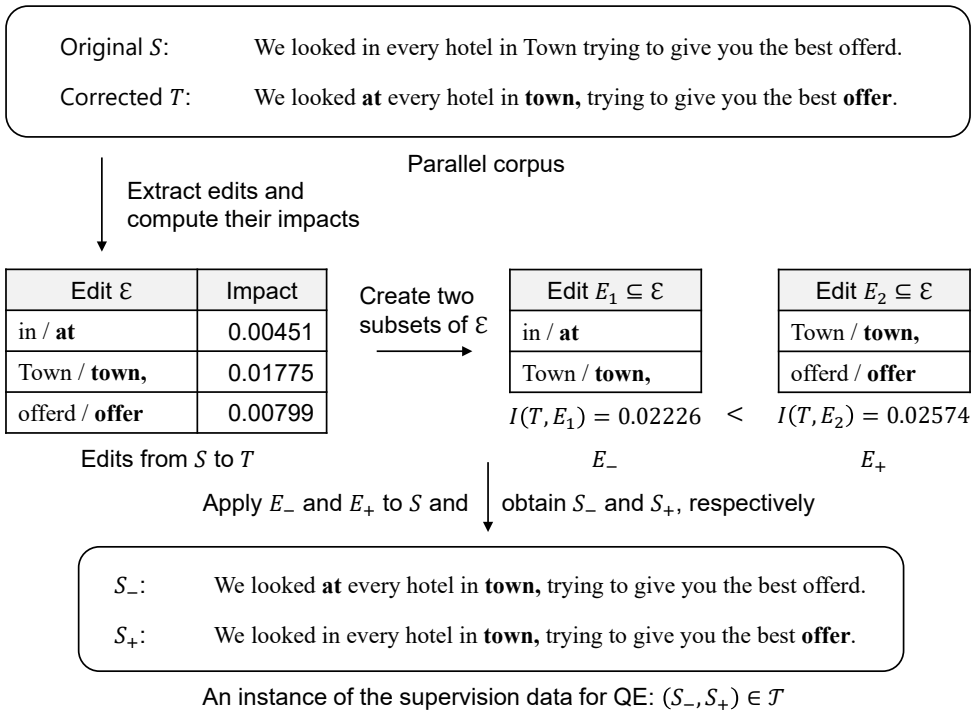

Figure 2: Procedure for generating supervision data for QE. We generate partially corrected sentences from a pair of original and corrected sentences in the parallel data, and determine their preference orders.

et al., 2017) to automatically extract a set of edits $\mathcal{E}$ between the original sentence S and corrected sentence T in the parallel data. We generate two subsets of edits $E_1, E_2 \subseteq \mathcal{E}$ using the procedure presented in the subsequent paragraph. We choose E_- and E_+ from the two subsets E_1 and E_2 such that $I(T, E_-) < I(T, E_+)$,

$$E_- = \operatorname*{argmin}_{E \in \{E_1, E_2\}} I(T, E), \tag{7}$$

$$E_+ = \operatorname*{argmax}_{E \in \{E_1, E_2\}} I(T, E). \tag{8}$$

Finally, we obtain two sentences S_- and S_+ by applying E_- and E_+, respectively, to the original sentence S. In this way, the supervision data $\mathcal{T}$ provides tuples of (S_-, S_+) where the sentence S_+ has a higher impact than S_- measured by the impacts of edits (Equation 6).

At last, we describe the procedure to obtain E_1 and E_2 from $\mathcal{E}$. We randomly create a subset $E_1 \subseteq \mathcal{E}$ with k elements, where $k \in \{1, 2, \ldots, |\mathcal{E}|\}$ is chosen from the discrete uniform distribution. Then, we modify E_1 to obtain another subset $E_2 \subseteq \mathcal{E}$. Initializing $E_2 = E_1$, we perform the following operation for each element $e \in \mathcal{E}$ with the probability $\frac{1}{|\mathcal{E}|}$:

$$E_2 \leftarrow \begin{cases} E_2 \cup \{e\} & \text{if } e \notin E_1 \wedge e \notin E_2 \\ E_2 \setminus \{e\} & \text{if } e \in E_1 \wedge e \in E_2 \end{cases}. \tag{9}$$

We discard E_1 and E_2 if $E_1 = E_2$ even after repeating the operation. Consequently, we randomly sample partial edits from $\mathcal{E}$ and enhance the diversity of partially corrected sentences.

2.5 Similarity estimator

The SE computes the semantic similarity between the original sentence X and its GEC output Y. Specifically, we compute the cosine similarity between contextual embeddings of two sentences X and Y,

$$\mathrm{SE}(X, Y) = \frac{\mathrm{BERT}(X) \cdot \mathrm{BERT}(Y)}{\|\mathrm{BERT}(X)\| \|\mathrm{BERT}(Y)\|}. \tag{10}$$

Here, the definition of $\mathrm{BERT}(.)$ is compatible with that for Equation 5.

3 Experiments

We conducted two types of experiments to examine the performance of IMPARA over existing methods. First, we compared correlations between automatic evaluation metrics and human assessments in GEC; this experiment measures the closeness of automatic and human evaluations. Second, we evaluate automatic evaluation metrics trained and tested in different text domains and correction styles; this experiment investigates the importance to train evaluation metric in a target domain/style and the

ability of IMPARA to adapt to different domains and styles.

3.1 Experimental settings

3.1.1 Hyperparameters

We used `BERT-BASE-CASED`[4] on HuggingFace as the pre-trained BERT model for the SE and QE. We kept the number of training instances $|\mathcal{T}|$ constant to 4096, regardless of test sets, to avoid the effects of the size of training data on the performance. The maximum number of training instances generated from a parallel sentence pair is 30. We set the learning rate to 10^{-5} and the batch size to 32. A dataset is split into training, validation, and test sets at a ratio of $8:1:1$. We selected the number of epochs from $1, 2, ..., 10$ that showed the best performance on the validation set. The threshold θ is set to 0.9 for the SE. This similarity value is higher than the maximum similarity value computed for any combinations of corrected sentences in the CoNLL-2014 dataset.

3.1.2 Baselines

Baseline methods include two reference-less automatic evaluation methods with different architectures: SOME (Yoshimura et al., 2020) and Scribendi Score (Islam and Magnani, 2021).

SOME employs BERT models optimized on human assessments for corrections and estimates correction quality from three perspectives: grammaticality, fluency, and meaning preservation. Models were trained on `tmu_gfm_dataset`[5], where five human raters manually assign evaluation scores for sentence pairs in the CoNLL-2013 dataset, with the hyperparameter settings of Yoshimura et al. (2020). Note that SOME was trained with additional human assessments for the CoNLL-2013 dataset, whereas IMPARA was trained only on the parallel data of the CoNLL-2013 dataset. Scribendi Score uses a language model to determine whether a correction improves the quality of a sentence. It also performs a superficial comparison of sentences to determine whether a correction is appropriate. We employed the pre-trained model `gpt-2` released by HuggingFace and `fuzzywuzzy`[6], a publicly available Python package to calculate the token sort ratio and Levenshtein distance ratio for sentence comparison.

3.1.3 Datasets and meta-evaluation metrics

The first experiment measured correlations between automatic evaluation metrics and human assessments on the dataset presented by Grundkiewicz et al. (2015). This dataset contains human rankings created for the outputs of 12 GEC systems on the CoNLL-2014 dataset. We computed Pearson's correlation coefficient (Pea) and Spearman's rank correlation coefficient (Spe) at the corpus level using the Expected Wins shown in Table 3(b) of Grundkiewicz et al. (2015). We also measured accuracy (Acc) and Kendall's rank correlation coefficient (Ken) for sentence-level comparison.

Meanwhile, Choshen and Abend (2018a) proposed MAEGE, an automatic methodology for GEC metric validation. MAEGE generates multiple partially corrected sentences from an incorrect sentence, and assigns pseudo scores to them, which are based on the number of edits applied. Then it computes correlation coefficients between the pseudo scores and the scores computed by automatic evaluation metrics. As this method do not require human assessment, it overcame difficulties such as low inter-rater agreement on human rankings. We calculated five correlations coefficients for meta-evaluation: Pearson's correlation coefficient (Pea) and Spearman's rank correlation coefficient (Spe) at corpus and sentence levels; and Kendall's rank correlation coefficient (Ken) at the chain level.

For the second experiment, we conducted meta-evalution using MAEGE on four different corpora: CoNLL-2013 dataset (Ng et al., 2013) (minimal edits), JFLEG (Napoles et al., 2017) (fluency edits), CWEB (Flachs et al., 2020) (website texts) and FCE (Yannakoudakis et al., 2011) (essay). We meta-evaluated IMPARA with different combinations of training and test sets to examine whether IMPARA can incorporate the characteristics of a dataset in the GEC evaluation. In this evaluation, we split each dataset into training and test sets at a ratio of $9:1$. In addition, we compared IMPARA with the two baseline methods in terms of domain adaptability on the four datasets.

3.2 Correlations with human assessments

Table 2 shows correlations between automatic evaluation metrics and human rankings. As we could not reproduce the scores reported in Islam and Mag-

[4] https://github.com/huggingface/transformers

[5] https://huggingface.co/datasets/tmu_gfm_dataset

[6] https://pypi.org/project/fuzzywuzzy

Method	Corpus		Sentence	
	Pea	Spe	Acc	Ken
Scribendi Score (paper)	0.951	0.940	-	-
Scribendi Score (ours)	0.303	0.729	0.414	-0.170
SOME	0.956	0.923	**0.777**	**0.555**
IMPARA	**0.974**	**0.934**	0.748	0.496
IMPARA (parallel only)	0.936	0.929	0.742	0.485

Table 2: Correlation with manual evaluation (Grundkiewicz et al., 2015) on CoNLL-2014.

Method	Corpus		Sentence		Chain
	Pea	Spe	Pea	Spe	Ken
Scribendi Score	0.884	0.981	0.374	0.421	**0.824**
SOME	**0.965**	**1.000**	0.394	0.439	0.563
IMPARA	0.951	0.990	**0.522**	**0.608**	0.692

Table 3: Meta-evaluation result using MAEGE on CoNLL-2014.

nani (2021), the first two rows presents the scores reported in their paper ("paper") and reproduced by our implementation ("ours").

IMPARA and SOME are the contenders in this evaluation; IMPARA was better than SOME at the corpus level, but inferior at the sentence level. It should be noted that IMPARA achieves the comparable performance to SOME without additional human assessments on the CoNLL-2013 dataset[7]. We also show IMPARA trained without the supervision data described in Section 2.4, but only on the parallel data ("parallel only" in Table 2). We observed the improvement by generating supervision data, comparing the last two rows in Table 2.

Table 3 reports the meta-evaluation results using MAEGE. As Choshen and Abend (2018a) suggested, we regarded this evaluation more important than the evaluation of Table 2. Again, IMPARA exhibits a comparable performance to SOME in this evaluation. Although IMPARA is slightly inferior to SOME at the corpus level, the correlation coefficients are quite high (≈ 1) to compare the two methods. Therefore, we focused on the evaluation at sentence and chain levels, which is more fine-grained than that at the sentence level. At the sentence and chain levels, IMPARA outperformed SOME with wide margins, +0.128 point (Pea), +0.169 point (Spe), and +0.129 point (Ken)[8].

[7]Again, we emphasize that SOME uses additional supervision data `tmu_gfm_dataset` for training the model.

[8]Scribendi Score overwhelmes other metrics in Ken, but this is because Scribendi Score assigns tie scores to many instances. Although such instances are *difficult* to decide preference orders, the MAEGE implementation excludes tie instances from the evaluation. In other words, MAEGE did not evaluate difficult instances for Scribendi Score. For reference, the ratios of tie instances are 42.5% (Scribendi Score), 0.04% (SOME), and 0.07% (IMPARA).

Dataset (eval)	Dataset (train)	Corpus		Sentence		Chain
		Pea	Spe	Pea	Spe	Ken
CoNLL-2013	CoNLL-2013	0.932	**1.000**	**0.411**	**0.515**	**0.688**
	CWEB	0.961	**1.000**	0.380	0.468	0.574
	JFLEG	0.959	0.990	0.344	0.408	0.568
	FCE	**0.967**	**1.000**	0.404	0.490	0.567
CWEB	CoNLL-2013	0.750	0.836	0.331	0.328	0.713
	CWEB	0.790	**0.963**	**0.472**	**0.432**	**0.780**
	JFLEG	0.757	0.818	0.353	0.354	0.775
	FCE	**0.805**	0.936	0.350	0.397	0.775
JFLEG	CoNLL-2013	0.959	0.990	0.516	0.604	0.677
	CWEB	0.952	0.972	0.524	0.572	0.644
	JFLEG	0.937	**1.000**	**0.618**	**0.685**	**0.783**
	FCE	**0.961**	0.990	0.581	0.649	0.627
FCE	CoNLL-2013	0.865	0.972	0.377	0.388	0.758
	CWEB	**0.882**	**0.990**	0.435	0.441	0.753
	JFLEG	0.852	0.972	0.390	0.429	0.739
	FCE	0.853	**0.990**	**0.541**	**0.616**	**0.848**

Table 4: Performance of IMPARA with different combinations of datasets used for training and evaluation. Using parallel data of the same domain and correction style for training and evaluation is important.

Dataset	Method	Corpus		Sentence		Chain
		Pea	Spe	Pea	Spe	Ken
CoNLL-2013	Scribendi	0.938	0.984	0.331	0.355	**0.698**
	SOME	**0.961**	**1.000**	0.370	0.419	0.502
	IMPARA	0.932	**1.000**	**0.411**	**0.515**	0.688
CWEB	Scribendi	0.637	0.451	0.177	0.194	0.616
	SOME	0.767	0.663	0.055	0.155	0.678
	IMPARA	**0.790**	**0.963**	**0.472**	**0.432**	**0.780**
JFLEG	Scribendi	0.932	0.945	0.255	0.303	0.574
	SOME	**0.955**	0.990	0.523	0.531	0.639
	IMPARA	0.937	**1.000**	**0.618**	**0.685**	**0.783**
FCE	Scribendi	**0.869**	0.933	0.342	0.449	**0.897**
	SOME	0.843	0.972	0.165	0.254	0.663
	IMPARA	0.853	**0.990**	**0.541**	**0.616**	0.848

Table 5: Performance of IMPARA and the two baseline methods on different datasets.

These results suggest that IMPARA achieves better evaluation performance than existing reference-less methods even without tailored data of human assessments, but only with parallel data.

3.3 Evaluation on other datasets

Table 4 summarizes correlation coefficients computed by MAEGE with different combinations of datasets used for training and evaluating the IMPARA model. For example, the second row of Table 4 shows the performance of IMPARA trained on CWEB and tested on CoNLL-2013. The table also indicates that IMPARA performed the best when the QE model was trained and evaluated on the same dataset. This observation is clearer when we evaluate IMPARA at the sentence and chain levels. Given that creating tailoerd evaluation data

Error type	Impact (10^{-2})	Frequency
NOUN	0.652	408
VERB:TENSE	0.649	480
VERB	0.580	557
NOUN:NUM	0.385	534
PUNCT	0.367	473
DET	0.364	1142
PREP	0.325	700

Table 6: Edit impacts for different error types (excluding OTHER) and their frequencies of occurrences in CoNLL-2014.

for each dataset with different domains and styles is expensive, IMPARA can provide a useful and practical solution, requiring only the parallel data on domains and styles.

Table 5 reports the performance of IMPARA and the two baseline methods on different datasets. IMPARA was trained on the training split of a dataset and evaluated on the test split. We did not adapt Scribendi Score to a target dataset because it is purely based on a pre-trained language model. It is impossible to adapt SOME to a target dataset because no human assessment data is available on the other datasets. SOME and Scribendi Score exhibits low performance on CWEB, FCE, and JFLEG in Table 5. In contrast, IMPARA achieved the highest correlations among all datasets. This result demonstrates the ability of IMPARA in adapting its evaluation metric to the target domain and style given by the parallel data.

3.4 Analysis

We analyzed the characteristic of edit impacts (Equation 5) computed by the BERT model. We examine edit impacts of different error types extracted by ERRANT on the CoNLL-2014 dataset. Table 6 presents the mean of edit impacts of error types, which appeared more than 400 times (excluding OTHER type)[9]. It is reasonable to find that edits for content words, such as NOUN (nouns) and VERB (verbs), have higher impacts than those for functional words, such as DET (determiner) and PREP (prepositions).

We focused more on the characteristics of impact by analyzing the highest and lowest correction impacts. Table 7 presents the corrections with the five largest edit impacts in the CoNLL-2014 dataset. In the first example, the sentence is not grammatical, and in the third, fourth, and fifth examples, the meanings of the sentences have been changed[10]. These sentences with higher impacts have relatively short length, indicating that a single correction has a large impact on the meaning of the entire sentence. In contrast, edits with the top five lowest impacts appeared in very long sentences (in 80, 235, 235, 235, and 235 words). Therefore, the longer a sentence, the smaller an impact of a correction in the sentence. These observations are consistent with the definition of Equation 6 and the importance of corrections recognized by humans.

When we generated supervision data in Section 2.4, edit impacts are not affected by the sentence length, which is constant for a given corrected sentence. Therefore, we compared edit impacts for the same corrected sentence. Table 8 shows an example of edits for content and functional words applied to the same corrected sentence. It demonstrates that the content word (i.e., VERB) has a higher impact than the functional words (i.e., DET, PREP) in the same corrected sentence, which is consistent with the results presented in Table 6.

4 Related Work

Reference-based metrics. Reference-based metrics for GEC use manually written reference sentences to evaluate a GEC system. M^2 (Dahlmeier and Ng, 2012), I-measure (Felice and Briscoe, 2015), and ERRANT (Bryant et al., 2017) are methods for evaluating GEC systems based on F-score. These methods requires explicit edit annotations to recognize the difference between the output and reference sentences to calculate precision, recall, and $F_{0.5}$. Napoles et al. (2015) proposed GLEU, a variant of BLEU metric (Papineni et al., 2002) commonly used for machine translation. GLEU does not require explicit edit annotations because it evaluates the quality of correction at the n-gram level. However, the reliability of reference-based metrics is low when reference sentences have low coverage (Choshen and Abend, 2018b).

Reference-less metrics. To address the issue of reference-based metrics, Napoles et al. (2016) introduced the reference-less metric that does not use reference sentences for the GEC evaluation. Their metric comprises a grammatical error detection tool and a language model, and showed a comparable performance to reference-based metrics. Islam and Magnani (2021) proposed Scribendi Score, which

[9]Insertion, replacement, and deletion are considered the same type.

[10]The correction of the second example in Table 7 is erroneous, but it actually appears in the CoNLL-2014 dataset.

Error Type	Sentence	Impact (10^{-2})
VERB	As a consequence , interpersonal skills _ / **are affected** .	6.57
NOUN	To some extent , this makes our life more luxurious and blundering / _ .	5.16
NOUN	One of the diseases is sickle cell trait / **anaemia** .	4.95
DET	People and friends often mock your / **their** conditions .	4.93
NOUN	They may not be able to enjoy a _ /**life** normal people can enjoy .	4.68

Table 7: Top-5 examples with the largest edit impacts in CoNLL-2014.

Error Type	Sentence	Impact (10^{-2})
Correct	Using text-messaging language as an informal way of communicating on social media networks also has a bad effect on us in the long term .	
VERB	Using text-messaging language as an informal way of communicating on social media networks also brings in / **has** a bad effect on us in the long term .	0.421
PREP	Using text-messaging language as an informal way of communicating on social media networks also has a bad effect for / **on** us in the long term .	0.236
DET	Using text-messaging language as an informal way of communicating on social media networks also has a bad effect on us in a / **the** long term .	0.164

Table 8: Examples of edits and their impacts computed for the same corrected sentence in CoNLL-2014.

is based on the perplexity of GPT-2 (Radford et al., 2019), token sort ratio, and Levenshtein distance ratio. These metrics require no GEC-specific language resource (e.g., supervision data for GEC). However, they cannot be adapted specifically to a particular domain or a correction style. Our experiments showed that they cannot evaluate GEC output robustly for a variety of domains and correction styles.

Some researchers proposed methods to directly optimize evaluation metrics on manual assessment scores for GEC outputs. Asano et al. (2017) presented a metric to evaluate GEC systems by combining sub-metrics of grammaticality, fluency, and meaning preservation. This metric is based on a regression model trained on GUG data (Heilman et al., 2014), language model, and METEOR (Denkowski and Lavie, 2014) as sub-metrics. Yoshimura et al. (2020) proposed a BERT-based metric wherein sub-metrics were optimized for human assessment scores. However, Takahashi et al. (2022) discovered that differences in learners' CEFR proficiency level[11] of a dataset affect the reliability of these metrics. Manual assessment scores are essential to these metrics to work in various domains and correction styles.

Meta evaluation methods. The performance of the metrics were judged by their closeness to human assessments (Banerjee and Lavie, 2005). A metric is generally compared using correlation coefficients between system outputs and human assessments for the outputs. Several studies have conducted meta-evaluations of automatic evaluation methods for several GEC systems (Grundkiewicz et al., 2015; Napoles et al., 2016; Asano et al., 2017). However, human annotations are known to yield poor inter-rater agreement. Therefore, Choshen and Abend (2018a) proposed MAEGE to meta-evaluate metrics without human annotations using sentence pairs ranked by the number of editing operations applied to ungrammatical sentences.

Reranking methods. Reranking methods used in GEC systems also estimate the quality of GEC outputs for choosing better corrections. Chollampatt and Ng (2018b) proposed the first neural-based reranking model that does not require any hand-crafted features for GEC. Using language models trained on large-scale corpora, such as BERT, several studies have improved the GEC performance further (Chollampatt et al., 2019; Kaneko et al., 2019). Liu et al. (2021) considered interactions of multiple outputs instead of evaluating them independently in reranking GEC outputs. The goal of these reranking models is to improve the GEG performance by selecting a better candidate from multiple candidates, not replicating a human GEC evaluation as in our study.

5 Conclusion

In this paper, we presented IMPARA, a novel reference-less metric for GEC. This method generates partially corrected sentences from parallel data comprising ungrammatical and grammatical sentence pairs. IMPARA learns their preference order of pairs of partially corrected sentences, which are

[11] https://www.cambridgeenglish.org/exams-and-tests/cefr/

determined by the impact of edits. The experiment results demonstrated that IMPARA achieved a comparable or better evaluation performance than existing reference-less methods even without tailored data of human assessments. In addition, IMPARA exhibited a high capability of adapting its metric to the target domain and style given by parallel data.

Future work includes providing an interpretable scale to scores computed by IMPARA. In addition, we plan to incorporate parallel data obtained by grammatical error generation. This may reduce the cost of building the parallel data and improve the quality of automatic evaluation metrics.

Acknowledgements

This paper is based on results obtained from a project, JPNP18002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Hiroki Asano, Tomoya Mizumoto, and Kentaro Inui. 2017. Reference-based metrics can be replaced with reference-less metrics in evaluating grammatical error correction systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–348, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

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*, pages 5755–5762, New Orleans, Louisiana. Association for the Advancement of Artificial Intelligence.

Shamil Chollampatt and Hwee Tou Ng. 2018b. Neural quality estimation of grammatical error correction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2528–2539, Brussels, Belgium. Association for Computational Linguistics.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Florence, Italy. Association for Computational Linguistics.

Leshem Choshen and Omri Abend. 2018a. Automatic metric validation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1372–1382, Melbourne, Australia. Association for Computational Linguistics.

Leshem Choshen and Omri Abend. 2018b. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Melbourne, Australia. Association for Computational Linguistics.

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

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. 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.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 8467–8478, Online. Association for 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.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180, Baltimore, Maryland. Association for Computational Linguistics.

Kengo Hotate, Masahiro Kaneko, Satoru Katsumata, and Mamoru Komachi. 2019. Controlling grammatical error correction using word edit rate. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 149–154, Florence, Italy. 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Masahiro Kaneko, Kengo Hotate, Satoru Katsumata, and Mamoru Komachi. 2019. TMU transformer system using BERT for re-ranking at BEA 2019 grammatical error correction on restricted track. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 207–212, Florence, Italy. Association for Computational Linguistics.

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.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Zhenghao Liu, Xiaoyuan Yi, Maosong Sun, Liner Yang, and Tat-Seng Chua. 2021. Neural quality estimation with multiple hypotheses for grammatical error correction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5441–5452, Online. Association for Computational Linguistics.

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.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. Jfleg: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. 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.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Yujin Takahashi, Masahiro Kaneko, Masato Mita, and Mamoru Komachi. 2022. Proficiency matters quality estimation in grammatical error correction. *arXiv preprint arXiv:2201.06199*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

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.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.