n-gram F-score for Evaluating Grammatical Error Correction

Shota Koyama[1,2], Ryo Nagata[3], Hiroya Takamura[2], Naoaki Okazaki[1,2]
[1]Tokyo Institute of Technology
[2]National Institute of Advanced Industrial Science and Technology [3]Konan University
shota.koyama@nlp.c.titech.ac.jp, nagata-inlg2024@ml.hyogo-u.ac.jp,
takamura.hiroya@aist.go.jp, okazaki@c.titech.ac.jp

Abstract

M^2 and its variants are the most widely used automatic evaluation metrics for grammatical error correction (GEC), which calculate an F-score using a phrase-based alignment between sentences. However, it is not straightforward at all to align learner sentences containing errors to their correct sentences. In addition, alignment calculations are computationally expensive. We propose *GREEN*, an alignment-free F-score for GEC evaluation. GREEN treats a sentence as a multiset of n-grams and extracts edits between sentences by set operations instead of computing an alignment. Our experiments confirm that GREEN performs better than existing methods for the corpus-level metrics and comparably for the sentence-level metrics even without computing an alignment. GREEN is available at https://github.com/shotakoyama/green.

1 Introduction

Grammatical error correction (GEC) is one of text generation tasks that aims to convert erroneous texts into error-corrected ones. Because of promising applications in second language learning, GEC has attracted widespread attention from the NLP community (Chollampatt and Ng, 2018a; Zhao et al., 2019; Sun et al., 2021; Kaneko et al., 2022; Zhou et al., 2023). Various automatic evaluation metrics for GEC have been proposed to make evaluations cheaper and faster by avoiding high-cost human evaluations.

M^2 (Dahlmeier and Ng, 2012) and its variants are the most widely used metrics in the automatic evaluation for GEC. They first compute a phrase-based alignment between sentences to extract edits of correction. They then calculate an F-score by comparing edits from the source to the reference sentences and edits from the source to the corrected sentences. The CoNLL-2014 shared task of GEC adopted M^2 as its evaluation metric, and the BEA-2019 shared task adopted ERRANT (Bryant et al., 2017), one of the variants of M^2. Currently, they are the representative metrics for GEC.

However, it is not straightforward at all to align source sentences (learner sentences containing errors) to their target sentences (correct sentences). In addition, the alignment calculation is computationally expensive and time-consuming for long sentences with many edits from the source sentence. Furthermore, M^2 requires manually annotated data with edits from the source to the reference sentences to extract edits; ERRANT needs no manually annotated data but depends on a part-of-speech tagger to perform the alignment calculation. Supposing that we could extract edits between sentences without alignments, we would design a more practical and useful alignment-free evaluation method that achieves the same level of performance as M^2 and ERRANT without depending on additional data or tools to extract the alignment.

In this paper, we propose GREEN, an **alignment-free** F-score for GEC evaluation, which treats a sentence as n-gram occurrences using a multiset (a set with repeated elements) of n-grams to compute an F-score by comparing edits between two multisets. We conducted experiments to verify the effectiveness of GREEN on the CoNLL-2014 evaluation dataset (Grundkiewicz et al., 2015) and the SEEDA dataset (Kobayashi et al., 2024). Even without computing an alignment, GREEN exhibits a higher correlation with human evaluation in terms of both Pearson and Spearman correlation coefficients for the corpus-level metrics. It also achieves comparable performance with existing methods for the sentence-level metrics.

2 Related Work

We review five existing representative reference-based metrics for GEC. M^2, ERRANT, PT-M^2, and CLEME are alignment-based F-scores. GLEU is a metric based on n-gram precision.

2.1 M^2 (Dahlmeier and Ng, 2012)

M^2 is the earliest and most representative GEC-specific automatic evaluation metric. M^2 calculates an F_β-score by comparing the system-corrected edits against human-annotated reference edits. Since the corrected sentences are not annotated with edits, M^2 automatically explores the corrected edits that have maximum overlaps with reference edits. This is the advantage of M^2 because we do not need to conduct manual annotations for system outputs once the reference annotations are provided.

One of the issues with M^2 is time complexity. M^2 finds the shortest path of a directed acyclic graph. Let the number of tokens in the source, reference, and corrected sentence be less than or equal to k. The bottleneck in the average case lies in the graph pruning algorithm to calculate the optimal alignment, which requires the $O(k^2)$ time complexity. However, in the worst case, when no nodes are pruned in this process, the numbers of nodes V and edges E are constant multiples of k^2 and k^4. Since topological sort requires $O(V+E)$ time complexity to find the shortest path, M^2 requires $O(k^4)$ in the worst case. The official implementation in the CoNLL-2014 shared task adopts the Bellman-Ford algorithm, which has a time complexity of $O(VE)$, resulting in the worst-case time complexity of $O(k^6)$. In this paper, we adopted the faster implementation[1] using topological sort.

Another issue is the inability to properly evaluate systems that generate corrupted sentences (Felice and Briscoe, 2015). M^2 gives $F = 0$ to a system that makes no changes to system-corrected sentences because M^2 calculates scores based on alignments. For this reason, M^2 may evaluate a system that generates outputs that are worse than the source text as $F \geq 0$. This is a common problem for other alignment-based F-score methods that are variants of M^2.

2.2 ERRANT (Bryant et al., 2017)

ERRANT computes an F-score by comparing the edits on the reference and corrected sentences similarly to M^2. ERRANT automatically extracts edits for both reference and corrected sentences using the linguistically enhanced alignment algorithm (Felice et al., 2016) based on the spaCy part-of-speech tagger and Damerau-Levenshtein distance, with time complexity of $O(k^2)$. The unnecessity of manually annotated reference edits is an advantage of ERRANT. We used the official implementation `v3.0.0`[2].

2.3 PT-M^2 (Gong et al., 2022)

PT-M^2 is a method that incorporates a pre-trained model into M^2. PT-M^2 calculates a score using BERT (Devlin et al., 2019) for edits extracted by M^2. M^2 gives a weight of 1 to each edit regardless of the impact of the edit, but PT-M^2 weights the edits by score, thus enabling it to give higher scores to corrected sentences containing more important corrections. We used the official implementation[3].

2.4 CLEME (Ye et al., 2023)

The original ERRANT equally evaluates edits of long and short phrases, resulting in unfair evaluations. CLEME performs edit extraction using ERRANT and evaluates the edits with length weighting. This length weighting gives larger weights to longer edits to prevent unfairness in the edit evaluation. We used the official implementation[4].

2.5 GLEU (Napoles et al., 2015, 2016a)

BLEU (Papineni et al., 2002), which is an n-gram-based metric for machine translation, shows a negative correlation on the CoNLL-2014 dataset (Grundkiewicz et al., 2015). GLEU is designed by adding a penalty term to the BLEU formula to show a positive correlation with human evaluation. GLEU is an $O(k)$ algorithm because it is an n-gram-based method. However, GLEU iterates 500 times to randomly sample one of the multiple references for each sentence, which makes the execution time of GLEU longer. In this paper, GLEU refers to the revised formula in Napoles et al. (2016a) and we explain this formula in Section 3.3. We adopted our reimplementation[5].

3 Proposed Method: GREEN

First, we describe GREEN with one reference sentence in Section 3.1. We will extend GREEN for multiple references in Section 3.2.

3.1 GREEN for Single Reference

GREEN treats a sentence as a multiset of n-grams with the maximum n-gram size N. For exam-

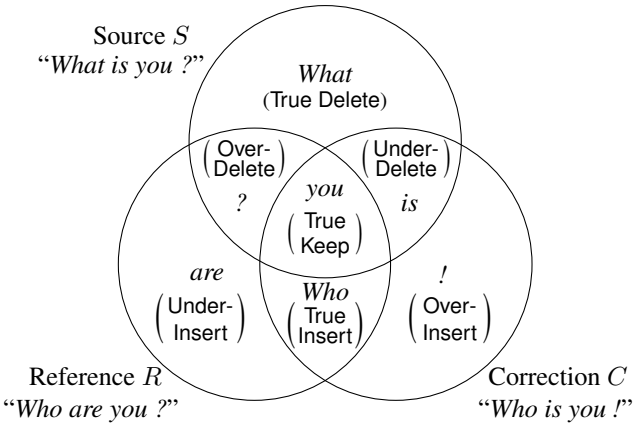
[1] https://github.com/craggy-otake/m2scorer_python3_fast

[2] https://github.com/chrisjbryant/errant

[3] https://github.com/pygongnlp/PT-M2

[4] https://github.com/THUKElab/CLEME

[5] This is because the original version is implemented in Python2.



Source S:	*What*	*is*	*you*	*?*
Reference R:	*Who*	*are*	*you*	*?*
Correction C:	*Who*	*is*	*you*	*!*

Figure 1: A three-set Venn diagram shows the occurrence of word 1-grams of S, R, C.

ple, a sentence "*a a b*" is treated as a multiset $\{a, a, b, a\text{-}a, a\text{-}b\}$[6] when we set $N = 2$[7]. GREEN considers the difference between multisets of n-grams as a correction. Corrections can be classified into deletion, insertion, and keep. For example, corrections from $\{a, c\}$ to $\{b, c\}$ involves deletion of a, which decreases the number of words, insertion of b, which increases the number of words, and keep of c, which does not change the word count[8].

GREEN compares the match between the corrections from the source sentence S to the reference sentence R and the corrections from S to the corrected sentence C. To count the match between $S \to R$ and $S \to C$, we introduce a Venn diagram illustrating the occurrences of word n-grams in S, R, C in Figure 1[9]. Table 1 shows what types of corrections are performed in $S \to R$ and $S \to C$, respectively, for all n-grams in each region of this Venn diagram. For example, the region $S \cap \overline{R} \cap \overline{C}$ contains n-grams that appear in S but not in R and C, such as "*What*". We call this region True Delete (TD) because these n-grams are correctly deleted through $S \to R$ and $S \to C$. Similarly, the region $\overline{S} \cap R \cap C$ containing n-grams inserted in both $S \to R$ and $S \to C$ is called True Insert (TI)

[6]In this paper, n-grams are represented by connecting each word with a hyphen instead of a whitespace to avoid confusing n-gram with sentence.

[7]Thus *a-a-b* is not included in this multiset.

[8]In GREEN, correction does not involve substitution. Substitution in alignment-based metrics corresponds to a combination of deletion and insertion in GREEN.

[9]We do not show n-grams of lengths two or more for simplicity in the Venn diagram.

Region	Name	$S \to R$	$S \to C$
$S \cap \overline{R} \cap \overline{C}$	True Delete	Delete	Delete
$\overline{S} \cap R \cap C$	True Insert	Insert	Insert
$S \cap R \cap C$	True Keep	Keep	Keep
$S \cap R \cap \overline{C}$	Over-Delete	Keep	Delete
$\overline{S} \cap \overline{R} \cap C$	Over-Insert	None	Insert
$S \cap \overline{R} \cap C$	Under-Delete	Delete	Keep
$\overline{S} \cap R \cap \overline{C}$	Under-Insert	Insert	None

Table 1: A table describes each region in Figure 1. Correction in which no n-gram appears in the common region involves "None".

and the region $S \cap R \cap C$ containing n-grams kept in both $S \to R$ and $S \to C$ is called True Keep (TK). TD, TI, and TK are True Positive (TP) because both $S \to R$ and $S \to C$ take the same type of corrections. The regions $S \cap R \cap \overline{C}$ and $\overline{S} \cap \overline{R} \cap C$ contain n-grams that are not deleted or inserted in $S \to R$, but are excessively deleted or inserted in $S \to C$. We call them Over-Delete (OD) and Over-Insert (OI), respectively. The elements in OD and OI are False Positive (FP) because they are mistakenly deleted or inserted in $S \to C$. The regions $S \cap \overline{R} \cap C$ and $\overline{S} \cap R \cap \overline{C}$ contain n-grams that should have been deleted or inserted in $S \to C$ as they are deleted or inserted in $S \to R$. We call them Under-Delete (UD) and Under-Insert (UI), respectively. The elements in UD and UI are False Negative (FN) because they should have been deleted or inserted in $S \to C$.

Next, we explain how to calculate the number of n-grams in each region of the Venn diagram by the operations on multisets. In this paper, we use three operations on multisets: intersection ($\cap$), union ($\cup$), and difference ($\setminus$). Each operation on multisets A and B is defined concerning the multiplicity of any element x in A and B. The multiplicity of an element x in a multiset A, which is denoted as $m_A(x)$, represents the number of times x occurs in A. For example, $m_A(a) = 2$ and $m_A(a\text{-}a) = 1$ when $A = \{\boldsymbol{a}, \boldsymbol{a}, b, \boldsymbol{a\text{-}a}, a\text{-}b\}$. In this paper, we define the three operations above as follows:

$$m_{A \cap B}(x) = \min(m_A(x), m_B(x)),$$
$$m_{A \cup B}(x) = \max(m_A(x), m_B(x)),$$
$$m_{A \setminus B}(x) = \max(m_A(x) - m_B(x), 0).$$

Hence, the number of n-gram x included in each region of the Venn diagram in Figure 1 is represented as follows:

$$\mathsf{TD}_{S,R,C}(x) = m_{S \cap \overline{R} \cap \overline{C}}(x) = m_{S \setminus (R \cup C)}(x)$$

$$= \max\{m_S(x) - \max(m_R(x), m_C(x)), 0\}, \quad (1)$$

$$\mathsf{TI}_{S,R,C}(x) = m_{\overline{S}\cap R\cap C}(x) = m_{(R\cap C)\setminus S}(x)$$
$$= \max\{\min(m_R(x), m_C(x)) - m_S(x), 0\}, \quad (2)$$

$$\mathsf{TK}_{S,R,C}(x) = m_{S\cap R\cap C}(x)$$
$$= \min(m_S(x), m_R(x), m_C(x)), \quad (3)$$

$$\mathsf{OD}_{S,R,C}(x) = m_{S\cap R\cap \overline{C}}(x) = m_{(S\cap R)\setminus C}(x)$$
$$= \max\{\min(m_S(x), m_R(x)) - m_C(x), 0\}, \quad (4)$$

$$\mathsf{OI}_{S,R,C}(x) = m_{\overline{S}\cap \overline{R}\cap C}(x) = m_{C\setminus(S\cup R)}(x)$$
$$= \max\{m_C(x) - \max(m_S(x), m_R(x)), 0\}, \quad (5)$$

$$\mathsf{UD}_{S,R,C}(x) = m_{S\cap \overline{R}\cap C}(x) = m_{(S\cap C)\setminus R}(x)$$
$$= \max\{\min(m_S(x), m_C(x)) - m_R(x), 0\}, \quad (6)$$

$$\mathsf{UI}_{S,R,C}(x) = m_{\overline{S}\cap R\cap \overline{C}}(x) = m_{R\setminus(S\cup C)}(x)$$
$$= \max\{m_R(x) - \max(m_S(x), m_C(x)), 0\}. \quad (7)$$

GREEN calculates TP, FP, and FN for each n-gram size. The TP, FP, and FN of n-grams for S, R, C are calculated as follows:

$$\mathsf{TP}_{n,S,R,C} = \sum_{\forall n\text{-gram } x} \left(\mathsf{TD}_{S,R,C}(x) + \mathsf{TI}_{S,R,C}(x) + \mathsf{TK}_{S,R,C}(x)\right),$$

$$\mathsf{FP}_{n,S,R,C} = \sum_{\forall n\text{-gram } x} \left(\mathsf{OD}_{S,R,C}(x) + \mathsf{OI}_{S,R,C}(x)\right),$$

$$\mathsf{FN}_{n,S,R,C} = \sum_{\forall n\text{-gram } x} \left(\mathsf{UD}_{S,R,C}(x) + \mathsf{UI}_{S,R,C}(x)\right).$$

Finally, GREEN accumulates TP, FP, and FN for corpus-level to obtain an F score. $\mathbb{S} = (S_1, \ldots, S_D), \mathbb{R} = (R_1, \ldots, R_D), \mathbb{C} = (C_1, \ldots, C_D)$ denote a set of D source, reference, and corrected sentences respectively. GREEN calculates precision and recall for n-gram lengths from 1 to N and the geometric mean of these precisions and recalls as BLEU (Papineni et al., 2002) does.

$$\mathrm{prec}(N, \mathbb{S}, \mathbb{R}, \mathbb{C}) = \left(\prod_{n=1}^{N} \frac{\sum_{i=1}^{D} \mathsf{TP}_{n,S_i,R_i,C_i}}{\sum_{i=1}^{D} \left(\mathsf{TP}_{n,S_i,R_i,C_i} + \mathsf{FP}_{n,S_i,R_i,C_i}\right)}\right)^{\frac{1}{N}},$$

$$\mathrm{recall}(N, \mathbb{S}, \mathbb{R}, \mathbb{C}) = \left(\prod_{n=1}^{N} \frac{\sum_{i=1}^{D} \mathsf{TP}_{n,S_i,R_i,C_i}}{\sum_{i=1}^{D} \left(\mathsf{TP}_{n,S_i,R_i,C_i} + \mathsf{FN}_{n,S_i,R_i,C_i}\right)}\right)^{\frac{1}{N}}.$$

At last, we calculate an F_β score as follows:

$$F_\beta(N, \mathbb{S}, \mathbb{R}, \mathbb{C}) = \frac{(1+\beta^2)\mathrm{prec}(N, \mathbb{S}, \mathbb{R}, \mathbb{C})\mathrm{recall}(N, \mathbb{S}, \mathbb{R}, \mathbb{C})}{\beta^2 \mathrm{prec}(N, \mathbb{S}, \mathbb{R}, \mathbb{C}) + \mathrm{recall}(N, \mathbb{S}, \mathbb{R}, \mathbb{C})}$$

where β is a factor denoting how important recall is in comparison to precision. In this paper, we call this F_β score GREEN_β.

3.2 GREEN for Multiple References

When we use multiple references, i.e., when m reference sentences $R_{i_1}, \ldots, R_{i_m}$ are given for the i-th source sentence S_i, GREEN selects the reference sentence $\hat{R}_i$ that maximizes the sentence-level GREEN for the corrected sentence C_i as follows:

$$\hat{R}_i = \underset{R\in\{R_{i_1},\ldots,R_{i_m}\}}{\mathrm{argmax}} \mathrm{GREEN}_\beta(N, (S_i), (R), (C_i)). \quad (8)$$

We compute $\mathrm{GREEN}_\beta(\mathbb{S}, \hat{\mathbb{R}}, \mathbb{C})$ using D reference sentences $\hat{\mathbb{R}} = \{\hat{R}_1, \ldots \hat{R}_D\}$ selected by Equation (8). This practice of selecting the reference that maximizes the sentence-level F-score is also adopted in M^2 and ERRANT.

3.3 Reformulation of GLEU

To compare GREEN with GLEU, we transform GLEU into a form using the representations in Equations (1) through (7). Equation (9) is a multiset-based representation of the original GLEU formula. The transformation in Figure 2 results in Equation (10). We can see that GLEU is calculated by subtracting UD as penalty term from the numerator of n-gram precision $\sum m_{R\cap C}(x) / \sum m_C(x)$. GLEU uses only TI, TK, OI, and UD from Equations (1) through (7), while GREEN uses all of them. GLEU has FNs in the penalty term but no FPs, which could lead to underestimating FPs and unreasonably giving high scores to systems that make aggressively incorrect edits.

4 Experiments

4.1 Settings

To demonstrate the effectiveness of GREEN, we computed its correlation with human judgments on the CoNLL-2014 evaluation dataset (Grundkiewicz et al., 2015) and the SEEDA dataset (Kobayashi et al., 2024). The CoNLL-2014 dataset is based on the test dataset of the CoNLL-2014 shared task (Ng et al., 2014), which utilizes student essays and consists of 1,312 source sentences. In this dataset, each instance has two reference sentences. This evaluation dataset consists of the rankings for each instance from 13 GEC system outputs (12 submissions of the shared task participants and the source

$$p_n = \frac{\sum_{\forall n\text{-gram } x \in R \cap C} m_{R\cap C}(x) - \sum_{\forall n\text{-gram } x \in S \cap C} \max\{0, m_{S\cap C}(x) - m_{R\cap C}(x)\}}{\sum_{\forall n\text{-gram } x \in C} m_C(x)} \tag{9}$$

$$= \frac{\sum_{\forall n\text{-gram } x \in R \cap C} m_{R\cap C}(x) - \sum_{\forall n\text{-gram } x \in S \cap C} \max\{0, \min(m_S(x), m_C(x)) - \min(m_R(x), m_C(x))\}}{\sum_{\forall n\text{-gram } x \in C} m_C(x)}$$

$$= \frac{\sum_{\forall n\text{-gram } x \in R \cap C} m_{R\cap C}(x) - \sum_{\forall n\text{-gram } x \in S \cap C} \max\{0, \min(m_S(x), m_C(x)) - m_R(x)\}}{\sum_{\forall n\text{-gram } x \in C} m_C(x)}$$

$$= \frac{\sum_{\forall n\text{-gram } x} m_{R\cap C}(x) - \sum_{\forall n\text{-gram } x} m_{(S\cap C)\setminus R}(x)}{\sum_{\forall n\text{-gram } x} m_C(x)} = \frac{\sum_{\forall n\text{-gram } x} \mathsf{TI}(x) + \mathsf{TK}(x) - \mathsf{UD}(x)}{\sum_{\forall n\text{-gram } x} \mathsf{TI}(x) + \mathsf{TK}(x) + \mathsf{OI}(x) + \mathsf{UD}(x)} \tag{10}$$

Figure 2: Reformulation of GLEU.

text). The SEEDA dataset shares the source and reference sentences with the CoNLL-2014 dataset. This dataset consists of the rankings for 15 corrected texts, including source text and two human-written texts. To follow modern trends in GEC, SEEDA employs the modern neural systems, while the CoNLL-2014 dataset consists of classical systems. The default setting of the SEEDA evaluation excludes two fluency texts (GPT-3.5 corrected text and human-written text) from 15 texts, and we followed this. SEEDA has two system rankings with different annotation methods: SEEDA-S for the sentence-based human evaluation and SEEDA-E for the edit-based human evaluation.

Following Grundkiewicz et al. (2015), we measure Pearson r and Spearman ρ correlation coefficients between the evaluation metric scores and human rankings. We must convert them into corpus-level system scores because the human judgment dataset consists of sentence-level rankings. We use the Expected Wins (EW) score (Bojar et al., 2013) employed in the WMT13 task of the evaluation metric as the corpus-level system score because Grundkiewicz et al. (2015) validated that we can obtain high accuracy by EW with the human judgment dataset for GEC.

In our experiments, for n-gram-based metrics, we use a maximum n-gram length of $N = 4$ for word-level tokenization following the setting of GLEU, and $N = 6$ for character-level following the setting of CHRF (Popović, 2015), which is a character-level metric for machine translation. The difference in tokenization is denoted as "wordGREEN" (word-level) or "charGREEN" (character-level).

Napoles et al. (2016b) reported that the average of sentence-level scores is better for evaluating the GEC systems than the corpus-level score when using M^2 and GLEU. However, corpus-level metric is adopted to measure the system performance in the CoNLL-2014 shared task (Ng et al., 2014) and the BEA-2019 shared task (Bryant et al., 2019). Because it is important for an evaluation measure to perform well at both the corpus-level and sentence-level metrics, we conduct experiments at both levels in this paper.

After the CoNLL-2014 shared task first adopted $\beta = 0.5$ for M^2, it has been the standard practice to use $F_{0.5}$ for alignment-based F-scores. Since it is more important for a GEC system to be precise than to correct as many errors as possible, it is considered better to weigh precision twice more than recall for M^2 and its variants. However, weighing precision more in n-gram-based F-score results that the metric most highly evaluates the unedited source sentence because precision is 100 for the source sentence, which contains no FPs. Therefore, we should not weigh precision more than recall in

	Corpus-Level Metrics						Sentence-Level Metrics					
	CoNLL		SEEDA-S		SEEDA-E		CoNLL		SEEDA-S		SEEDA-E	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Alignment-based F-score												
M^2	0.623	0.687	0.616	0.517	0.736	0.776	0.872	0.731	0.797	0.762	0.869	**0.951**
ERRANT	0.644	0.687	0.529	0.364	0.690	0.699	0.871	0.775	0.764	0.727	0.855	0.930
PT-M^2	0.686	0.786	0.737	0.720	0.798	0.916	**0.934**	**0.890**	0.831	0.804	0.878	0.930
CLEME	0.648	0.709	0.573	0.427	0.702	0.727	0.877	0.824	0.818	0.804	0.872	0.930
n-gram-based precision												
wordGLEU	0.696	0.445	0.870	0.811	0.891	0.895	0.779	0.720	0.926	**0.923**	0.915	0.916
charGLEU	0.606	0.593	0.807	0.706	0.843	0.867	0.655	0.665	0.880	0.853	0.905	0.937
n-gram-based F-score												
wordGREEN	0.741	0.698	**0.920**	**0.909**	**0.911**	**0.930**	0.835	0.731	0.922	0.902	0.920	0.937
charGREEN	**0.786**	**0.813**	0.913	0.881	**0.911**	0.909	0.834	0.852	**0.928**	0.881	**0.930**	0.916

Table 2: Pearson (r) and Spearman (ρ) correlation coefficients between each metric and the human score of the CoNLL-2014 evaluation dataset and the SEEDA dataset.

Metric	AMU	AMU-S
M^2	4.34	196.60
ERRANT	12.35	14.34
PT-M^2	109.82	$>$ 1 hour
CLEME	10.15	12.10
wordGLEU	2.69	2.80
wordGREEN	0.55	0.56

Table 3: The average execution time in seconds to evaluate the AMU system output in the CoNLL-2014 dataset and the slow AMU (AMU-S) in which one sentence in AMU is replaced by an example making M^2 slow.

n-gram-based F-score. Furthermore, we should rather weigh recall more than precision because the effect of individual annotator bias (Bryant and Ng, 2015) may unreasonably reduce precision due to the system corrections such that they are correct but not edited by the annotator. To alleviate this annotator bias, we employ $\beta = 2.0$, which weighs recall twice more than precision, for GREEN in our experiments.

4.2 Results of Corpus-Level Metrics

The correlation coefficients between the reference-based corpus-level GEC metrics and the EW scores on the CoNLL and SEEDA datasets are shown in the left half of Table 2. We confirmed that wordGREEN or charGREEN performs the best in these corpus-level metrics. We confirmed that wordGREEN and charGREEN perform the best on the CoNLL-2014 and SEEDA datasets, respectively, in corpus-level metrics. The three alignment-based F-scores of M^2, ERRANT, and CLEME show similar performance, while PT-M^2 is better than these metrics, which implies that the impact of incorporating the pre-trained model is significant. GLEU shows a relatively worse performance with Spearman ρ in CoNLL-2014 as shown in Chollampatt and Ng (2018b), while GLEU shows a relatively better performance in SEEDA as shown in Kobayashi et al. (2024). We can confirm that GREEN, in contrast to GLEU, performs consistently well in both classical and neural system evaluations.

4.3 Results of Sentence-Level Metrics

The correlation coefficients between the reference-based sentence-level GEC metrics and the EW scores on the CoNLL and SEEDA datasets are shown in the right half of Table 2. We can confirm that wordGREEN and charGREEN show comparable performance to the existing sentence-level metrics. In particular, charGREEN shows the best Pearson correlation coefficients r on the SEEDA-S and SEEDA-E datasets. On CoNLL-2014, PT-M^2 shows the highest correlation using a pre-trained model BERT. All the sentence-level metrics show higher correlations than their corpus-level counterparts, as shown in Napoles et al. (2016b). The GEC field needs to investigate why sentence-level metrics are good in future work.

4.4 Efficiency of GREEN

We measured the average execution time of 10 runs to calculate the score for evaluating the output of the AMU system that shows the highest score with human evaluation in the CoNLL-2014 shared task. As mentioned in Section 2.1, the worst-case time

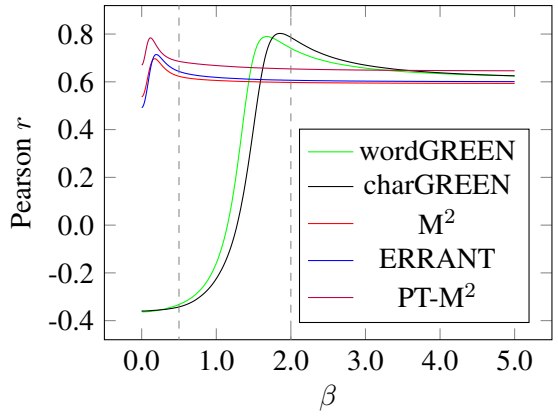


Figure 3: Pearson correlation coefficient on the CoNLL-2014 dataset varying β.

complexity of M^2 is quite high. We also measure the average execution time of AMU-S, which replaces one sentence of AMU with an example[10] making M^2 slow because it corresponds to the worst-case scenario. We show the execution times in seconds in Table 3. GREEN has the advantage of being faster than other methods in execution time, although its performance is better than or comparable to others. M^2 and PT-M^2 are not practical in the worst-case scenario. The advantage of GREEN is that it does not require linguistic resources to compute alignments or pre-trained models, which enables even non-English GEC to perform the evaluation immediately and efficiently in linear time, without the preparation of annotated data required in M^2 and PT-M^2 or linguistic resources required in ERRANT and CLEME. Despite an n-gram frequency-based method, GLEU takes a longer execution time than GREEN because GLEU samples random references 500 times when using multiple references.

5 Analysis

5.1 Impact of β for F-score

In Section 4, we confirmed the effectiveness of GREEN in terms of performance and efficiency. In our experiments, we employed $\beta = 2.0$. We investigate the impact of β on the performance of GREEN and other F-score-based metrics. We show the change of Pearson r for F-based corpus-level metrics on the CoNLL-2014 dataset when changing the β from 0.00 to 5.00 in 0.01 increments in Figure 3. ERRANT and PT-M^2, which are variants of M^2, show a similar trend to M^2 in

[10]We included this in Appendix A.

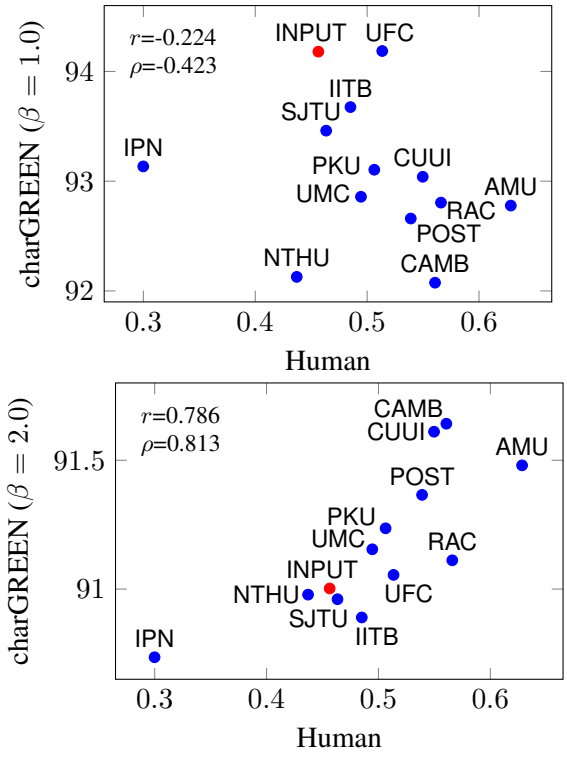


Figure 4: Scatter plots of corpus-level charGREEN scores with $\beta = 1.0$ and that with $\beta = 2.0$ on the CoNLL-2014 submissions.

that they correlate better for $0 \leq \beta \leq 0.5$. We can see that these alignment-based methods and the n-gram-based method GREEN show different trends in changing β. GREEN performs better than M^2 and its variants when we set the appropriate β such as 2.0. However, if β is too small, the performance degrades, resulting in negative correlations.

To investigate this cause, we show the corpus-level charGREEN and EW scores at $\beta = 1.0, 2.0$ in Figure 4. CharGREEN with $\beta = 1.0$ gives unreasonably high scores to IITB, INPUT, SJTU, and UFC. INPUT is the source text without any corrections, and IITB, SJTU, and UFC are the three system outputs with the fewest corrections from the source among all outputs. Because these outputs obtain the high precision, GREEN gives unreasonably high scores to them with a smaller β. CharGREEN with $\beta = 2.0$ gives higher scores to systems that actively make correct corrections (AMU) and lower scores to systems that are excessively conservative (IITB) or make many incorrect corrections (IPN), resulting in a high correlation on the CoNLL-2014 evaluation dataset.

5.2 Evaluating Source and Degradation

Felice and Briscoe (2015) pointed out that M^2 suf-

	AMU	INPUT	IPN	NULL
Alignment-based F-score				
M^2	35.01	0.00	7.09	28.01
ERRANT	31.97	0.00	5.95	0.20
PT-M^2	35.94	0.00	5.72	2.44
CLEME	25.14	0.00	4.41	33.44
n-gram-based precision				
wordGLEU	58.08	56.34	55.08	0.00
charGLEU	81.68	81.75	81.06	0.00
n-gram-based F-score				
wordGREEN	79.26	76.93	76.31	43.46
charGREEN	91.48	91.00	90.74	31.28
human	0.628	0.456	0.300	-

Table 4: Scores for AMU, INPUT, IPN, and NULL by GEC metrics.

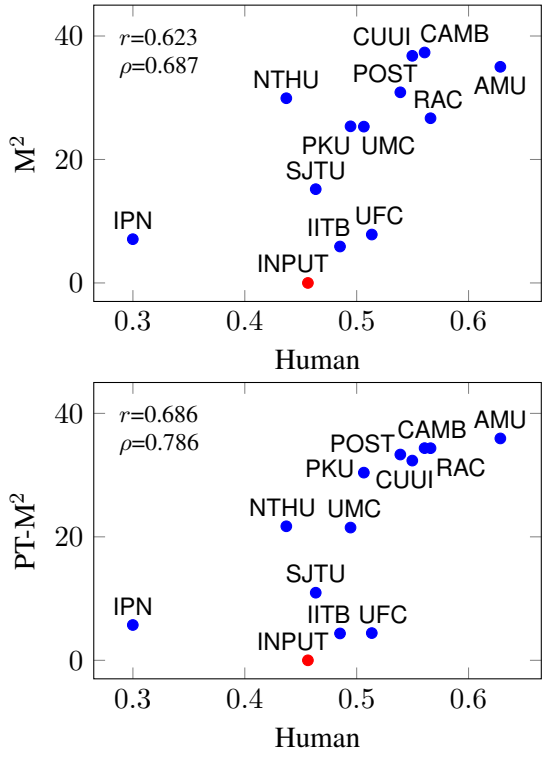


Figure 5: Scatter plots of corpus-level M^2, and PT-M^2 scores on the CoNLL-2014 submissions.

fers from the issue that it cannot evaluate the degraded output text as worse than the source text. Napoles et al. (2015) indicated that its cause is that M^2 maximally matches the wrong phrase deletions to the reference edits. In fact, given a system that always outputs an empty sentence for each input sentence (we refer to this system as NULL), this system would rank sixth out of 13 systems (12 actual task participants and NULL) if it had participated in the CoNLL-2014 shared task. This indicates the insensitivity of M^2 to corrupted text, such as that generated by NULL. The reason is that M^2 matches the long phrase deletions by NULL to the correct edits in reference and M^2 gives NULL a higher score than it actually is. Table 4 shows the scores of the CoNLL-2014 dataset by GEC metrics for AMU (the best system in the human judgment), INPUT (the source), IPN (the worst system) and NULL (empty text). Alignment-based F-scores (M^2, ERRANT, PT-M^2, CLEME) gives 0.00 to INPUT containing no edits to evaluate. M^2 wrongly evaluates NULL as a relatively better output because it maximally matches phrase deletions. Although PT-M^2 faces the same problem as M^2, it can avoid giving a high score to NULL by its model-based weighted score. CLEME also wrongly gives a high score to NULL because it excludes empty output sentences from the target of evaluation. Since three of 1312 sentences are deleted completely in the CoNLL-2014 reference dataset, CLEME calculates the score of NULL by only evaluating these three sentences. Since ERRANT uses the linguistically enhanced alignment, it does not match whole-sentence deletions with the correct reference edits while giving a score of 0.20 for the three deleted sentences.

Figure 5 shows the scores of M^2 and PT-M^2 and the EW scores. These two methods give scores highly correlated with the human evaluation to the systems with human scores between 0.5 and 0.6. However, they give inconsistent values to the systems with EW scores between 0.4 and 0.5. We can see that the alignment-based F-score has problems in evaluating the source and degradation.

Both wordGREEN and charGREEN can evaluate the systems in Table 4 in the correct order (AMU > INPUT > IPN > NULL). WordGLEU can evaluate as GREEN does, however, charGLEU fails to evaluate AMU better than INPUT. GLEU cannot evaluate TD, as shown in Equation (10), which results in rating NULL to be 0. On the other hand, GREEN can also evaluate TDs in NULL.

5.3 Difference between Corpus-level Metric and Sentence-level Metric

To investigate why sentence-level metrics perform better than their corpus-level counterparts, we show the score of sentence-level charGREEN and M^2 in Figure 6. We did not find enough differences between corpus-level charGREEN (shown in Figure 4) and sentence-level charGREEN worth mentioning. On the other hand, sentence-level M^2 gives

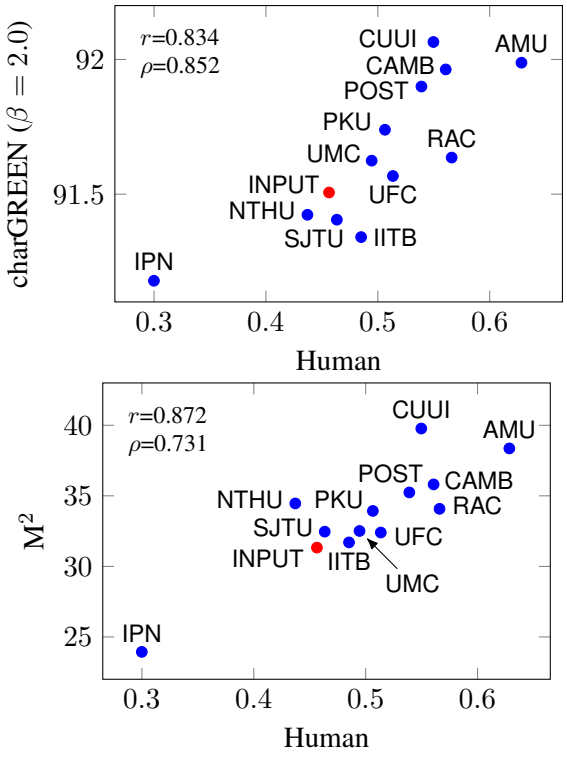


Figure 6: Scatter plots of sentence-level charGREEN and M^2 scores on the CoNLL-2014 submissions.

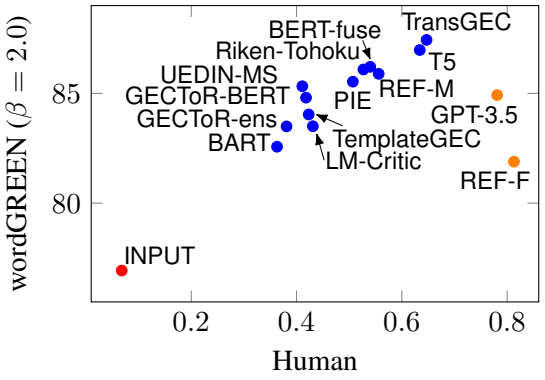


Figure 7: Scatter plots of corpus-level wordGREEN with $\beta = 2.0$ on the SEEDA-S dataset.

scores correlated with the human evaluation to the systems with EW scores between 0.4 and 0.5 while corpus-level M^2 fails (shown in Figure 5). This is because sentence-level M^2 gives $F = 1.0$ to cases where $S = R = C$, resulting in alleviating the bias to give lower scores to cases closer to INPUT.

5.4 Incorporating Pre-trained Model

We can see that M^2 and PT-M^2 show similar tendencies as a whole, but locally PT-M^2 behaves more similarly to human evaluation. For example, in Figure 5, the plotted points in the range of 0.5 to 0.6 of the human score are straightly aligned in PT-M^2, but scattered in M^2. This implies the effectiveness of incorporating the pre-trained model in GEC evaluation. Incorporating the pre-trained model into GREEN may realize the state-of-the-art GEC evaluation. We leave this for future work.

5.5 Evaluating Fluency Edit

We follow the default setting of the SEEDA evaluation in which we exclude the two fluency-editing systems (GPT-3.5 and REF-F) from the calculation of correlation coefficients. To observe the behavior of evaluating fluent texts by GREEN, we show the score of corpus-level wordGREEN and EW of the SEEDA-S dataset in Figure 7. We can cofirm that INPUT (shown by a red dot) and the systems in the default setting (shown by blue dots) show a high correlation with GREEN. On the other hand, two fluency-editing systems (shown by orange dots) stand out as outliers. This result is obvious because the reference texts used in the SEEDA evaluation are not fluency-edited texts. However, we need further study on how to properly evaluate fluency-edited texts such as LLM-generated texts, using reference-based evaluation metrics.

6 Conclusions

We proposed an alignment-free GEC evaluation metric, GREEN, which computes F-score by comparing edits between multisets. GREEN shows a higher correlation for both Pearson and Spearman correlation coefficients for the corpus-level metrics and comparable performance with existing evaluation metrics for the sentence-level metrics while it runs faster than existing methods and does not require the alignment calculation. We also analyzed the effect on β for F-score-based methods. We confirmed that alignment-based methods and GREEN have different tendencies on β. We investigated the problem that alignment-based F-score is difficult to evaluate the source text and degraded text correctly. We confirmed that corpus-level GREEN properly evaluates systems in contrast to existing corpus-level metrics, and sentence-level metrics alleviate the bias of alignment-based F-score on the source and degraded texts. Further challenges include incorporating pre-trained models and evaluating fluency-edited texts.

Acknowledgments

We thank all anonymous reviewers for their careful reading and constructive comments. This work

was supported by JSPS KAKENHI Grant Number JP23KJ0930 and a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. 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.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 697–707, Beijing, China. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018a. 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 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.

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.

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.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. 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.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. Revisiting Meta-evaluation for Grammatical Error Correction. *Transactions of the Association for Computational Linguistics*, 12:837–855.

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, Matt Post, and Joel Tetreault. 2016a. GLEU without tuning. *Preprint*, arXiv:1605.02592.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016b. 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.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. 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.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Xin Sun, Tao Ge, Furu Wei, and Houfeng Wang. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5937–5947, Online. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, Qingyu Zhou, Yangning Li, Shirong Ma, Hai-Tao Zheng, and Ying Shen. 2023. CLEME: Debiasing multi-reference evaluation for grammatical error correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6174–6189, Singapore. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. 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 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving Seq2Seq grammatical error correction via decoding interventions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

A Example that M^2 Takes a Long Time to Calculate

In an issue of the official M^2 GitHub repository[11], an example is given in which M^2 takes a long time to calculate. Here is the example in this issue:

> As it is a genetic risk , the patient force might have a high chance of carrying the risk , hence the need to inform their relatives is important . Hence , you are suffering from a genetic disease that the genetic trait might be passed on to your next generation if you have a child . Hence , there is no legal obligation to disclose to their family members , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there is no legal obligation . Hence , there

Such a degeneration of repetition sometimes occurs in neural text generation (Holtzman et al., 2020). In AMU-S, the 333rd sentence in AMU is replaced by this sentence.

[11] https://github.com/nusnlp/m2scorer/issues/8