CLEME2.0: Towards More Interpretable Evaluation by Disentangling Edits for Grammatical Error Correction

Anonymous ACL submission

Abstract

The paper focuses on improving the interpretability of Grammatical Error Correction (GEC) metrics, which receives little attention in previous studies. To bridge the gap, we propose **CLEME2.0**, a reference-based evaluation strategy that can describe four elementary dimensions of GEC systems, namely hit-correction, error-correction, under-correction, and over-correction. They collectively contribute to revealing the critical characteristics and locating drawbacks of GEC systems. Evaluating systems by Combining these dimensions leads to high human consistency over other reference-based and reference-less metrics. Extensive experiments on 2 human judgement datasets and 6 reference datasets demonstrate the effectiveness and robustness of our method.[1]

1 Introduction

Grammatical Error Correction (GEC) is the task of automatically detecting and correcting all grammatical errors in a given text (Bryant et al., 2023; Ma et al., 2022; Ye et al., 2022). A core component of any NLP tasks is the development of automatic metrics that can objectively measure model performance (Bryant et al., 2023). However, proposing appropriate evaluation of GEC has long been a challenging task (Madnani et al., 2011), due to the subjectivity (Bryant and Ng, 2015), complexity (Mita et al., 2019) and subtlety (Choshen and Abend, 2018) of GEC (Napoles et al., 2015).

Recent studies have been trying to develop GEC metrics that can achieve high correlations with human judgements (Yoshimura et al., 2020a), with less attention paid to the interpretability of the automatic metrics. We define the interpretability of metrics as their ability to reveal the concerned characteristics of systems, which is vital in locating the drawbacks of a certain system. It is well-acknowledged that excellent GEC systems, which usually conform to the principle of minimal editing, should adhere to two gold principles, namely grammaticality and faithfulness. Grammaticality necessitates that all grammatical errors should be accurately corrected, while faithfulness requires that the corrections maintain the original textual meaning and syntactic structure. However, the widely-adopted mainstream GEC metrics (Bryant et al., 2017; Ye et al., 2023) indicate the GEC performance by precision, recall, and F scores, which can hardly characterize these critical dimensions of GEC systems, thus hindering the development.

Figure 1: An example of edit disentanglement. We highlight TP, $\mathrm{FP}_{\mathrm{ne}}$, $\mathrm{FP}_{\mathrm{un}}$, and FN in different colors.

Therefore, we propose **CLEME2.0**, a more interpretable reference-based evaluation strategy that can describe four fundamental aspects of GEC systems: hit-correction, error-correction, under-correction, and over-correction. The first three aspects are responsible for describing grammaticality, while the last one is for faithfulness since the over-correction edits tend to change the original semantics, especially for LLMs (Coyne et al., 2023). To achieve this, CLEME2.0 distinguishes between necessary and unnecessary corrections and disentangles edits into four main types: true positive (TP), necessary false positive ($\mathrm{FP}_{\mathrm{ne}}$), unnecessary false positive ($\mathrm{FP}_{\mathrm{un}}$), false negative (FN) edits.[2] For example in Figure 1, the Hyp.1 makes three necessary edits on the right positions, where [*the* $\rightarrow \epsilon$] is a TP edit but two of others ([*were* $\rightarrow$ *was*] and [*for* $\rightarrow$ *in*]) are $\mathrm{FP}_{\mathrm{ne}}$ edits since they are not covered in the reference. So Hyp.1 tends to mistakenly correct grammatically errors. On the other hand, Hyp.2

[1] All the codes will be released after the peer review.

[2] True negative edits are not considered in our method.

makes extra two $\mathrm{FP_{un}}$ edits ([$\epsilon \rightarrow$ *of*] and ([*century* $\rightarrow$ *centuries*]) since the reference does *not* correct the right positions, indicating the occurrence of two under-correction phenomena. Additionally, [*for* $\rightarrow$ *for*] of the Hyp.2 is considered as an FN edit, which means the occurrence of an under-correction phenomenon. Since the edit disentanglement is based on the chunk partition technique proposed by CLEME, so we dub this strategy as CLEME2.0.

Disentangling edits enables us to investigate concrete dimensions of GEC systems by computing upon an evaluation dataset four disentangled scores: hit-correction, error-correction, under-correction, and over-correction scores. In contrast to mainstream GEC metrics like ERRANT (Bryant et al., 2017) and MaxMatch (Dahlmeier and Ng, 2012a) that reveal the system performance by P/R/$F_{0.5}$, this disentanglement can provide an interpretable insight into fine-grained dimensions responsible for describing critical characteristics of GEC systems. Then, we integrate these disentangled scores into a comprehensive score using linear weighted summation, placing different emphases on disentangled scores. We leverage the comprehensive score to indicate the system performance from a global perspective. Similar to CLEME (Ye et al., 2023), CLEME2.0 also supports the evaluation based on either correction dependence or correction independence assumptions, providing a flexible option.

Besides, we assume that edits with various extents of modification should affect distinctively the evaluation results. Therefore, we incorporate two edit weighting techniques into CLEME2.0, namely similarity-based weighting (Gong et al., 2022) and LLM-based weighting. Specifically, the techniques compute an important weight for each edit using a language model rather than treating each edit equally, thus equipping CLEME2.0 with abilities to capture context semantics and overcome the defect of traditional measures relying on superficial form similarity (Kobayashi et al., 2024a).

To verify the effectiveness of CLEME2.0, we conduct extensive experiments on 2 human judgment datasets (GJG15 (Grundkiewicz et al., 2015) and SEEDA (Kobayashi et al., 2024b)), where our method consistently achieves high correlations. We also demonstrate the robustness of CLEME2.0 by computing the evaluation results based on 6 reference datasets with disparate annotation styles. In summary, our contributions are three folds:

(1) We propose CLEME2.0, a more interpretable evaluation strategy, which is beneficial to reveal crucial characteristics of GEC systems.

(2) We boost CLEME2.0 with two edit weighting techniques, including similarity-based and LLM-based weighting, to overcome the inability of traditional reference-based metrics.

(3) Extensive experiments and analyses are conducted to confirm the effectiveness and robustness of our proposed method.

2 Related Work

Reference-based metrics. Reference-based metrics evaluate GEC systems by referencing manually written materials. The M^2 scorer (Dahlmeier and Ng, 2012b) identifies optimal edit sequences between source sentences and system hypotheses, using the F0.5 score. However, this method can inflate scores by manipulating edit boundaries. Bryant et al. (2017) proposed ERRANT, which improves edit extraction with a linguistically-informed alignment algorithm, but it remains language-dependent and biased in multi-reference evaluation. Napoles et al. (2015) introduced GLEU, an n-gram-based metric inspired by BLEU for GEC evaluation. Ye et al. (2023) proposed CLEME to eliminate bias in multi-reference evaluation by transforming the source, hypothesis, and references into chunk sequences with consistent boundaries, providing unbiased $F_{0.5}$ scores. Gong et al. (2022) introduce PT-M^2, focusing on scoring changed words extracted by the M^2 metric.

Reference-less metrics. To overcome the limitations of reference-based metrics, recent research focus on reference-less scoring. Inspired by quality estimation in NMT, Napoles et al. (2016a) propose Grammaticality-Based Metrics (GBMs) using an existing GEC system or a pre-trained ridge regression model. Asano et al. (2017) enhance GBMs by adding criteria like grammaticality, fluency, and meaning preservation. Yoshimura et al. (2020b) introduce SOME, which uses sub-metrics optimized for manual assessment with regression models. Scribendi Score (Islam and Magnani, 2021) combines language perplexity and token/Levenshtein distance ratios. IMPARA (Maeda et al., 2022) incorporates a Quality Estimator and a Semantic Estimator based on BERT to evaluate GEC output quality and semantic similarity. While reference-less metrics align well with human judgments, they lack interpretability due to the heavy dependence on trained models, thus posing latent risks.

3 Method

Our CLEME2.0 can be generally divided into three main steps, with the overview shown in Figure 2. Additionally, we incorporate two distinct edit weighting techniques to enhance performance.

3.1 Edit Extraction

The first step is edit extraction. Given a source sentence X and a target (either hypothesis or reference) sentence Y, this step is to extract the edits describing the modification from X to Y. Here, we utilize the chunk partition technique from CLEME (Ye et al., 2023) to execute the process of edit extraction. Unlike the traditional metrics like ERRANT (Bryant et al., 2017) and MaxMatch (Dahlmeier and Ng, 2012a), CLEME concurrently aligns all sentences, including the source, the hypothesis, and all the references. This facilitates segmentation of them all into chunk sequences with an equal number of chunks, irrespective of the varying token counts in different sentences, as delineated in Figure 2. It is worth noting that a chunk is a basic edit unit, which can be unchanged, corrected, or dummy (empty) (Ye et al., 2023).

3.2 Disentangled Scores

For the purpose of computing disentangled scores, we initially disentangle edits into four core types. 1) **TP edits** refer to the corrected/dummy hypothesis chunks that share the same tokens as the corresponding reference chunks. 2) **FP$_{\text{ne}}$ edits** are the corrected/dummy hypothesis chunks that have different tokens from those in the corresponding reference chunks wherein the reference chunks are also corrected/dummy ones. 3) **FP$_{\text{un}}$ edits** are the corrected hypothesis chunks but their corresponding reference chunks remain unchanged. 4) **FN edits** indicate the unchanged hypothesis chunks but the corresponding reference chunks are corrected/dummy. It is highlighted that traditional metrics (Dahlmeier and Ng, 2012a; Bryant et al., 2017) do not distinguish between FP$_{\text{ne}}$ and FP$_{\text{un}}$, treating both as FP, thereby resulting in confusion between error-correction and over-correction. Actually, we have $FP = FP_{\text{ne}} + FP_{\text{un}}$.

Furthermore, we can differentiate between necessary and unnecessary edits. TP, FP$_{\text{ne}}$, and FN edits are all *necessary* edits, since their corresponding reference chunks are also corrected/dummy, implying the existence of grammatical errors in the related parts of X. On the contrary, FP$_{\text{un}}$ edit are *unnecessary* edits because the systems propose corrections not represented in references. Consequently, we can define four disentangled scores.

Hit-correction score. This paper defines the hit-correction score as the ratio of TP edits to all necessary reference edits. Its purpose is to quantify the accuracy with which systems offer correct corrections. The formula is as follows:

$$Hit = \frac{TP}{necessity} = \frac{TP}{TP + FP_{\text{ne}} + FN} \quad (1)$$

Error-correction score. Conversely, the error-correction score is defined as the ratio of FP$_{\text{ne}}$ edits to all necessary reference edits. This score seeks to evaluate the degree to which systems generate erroneous corrections for grammatical errors. The formula for this score is as follows:

$$Error = \frac{FP_{\text{ne}}}{necessity} = \frac{FP_{\text{ne}}}{TP + FP_{\text{ne}} + FN} \quad (2)$$

Under-correction score. Similarly, the under-correction score is proposed to measure the degree to which systems omit to correct grammatical errors, which is computed as follow:

$$Under = \frac{FN}{necessity} = \frac{FN}{TP + FP_{\text{ne}} + FN} \quad (3)$$

Over-correction score. The score is introduced in response to frequent observations that LLMs are prone to over-correcting texts. This score is determined by the proportion of FP$_{\text{un}}$ edits to all hypothesis corrected/dummy edits, aiming to gauge the level to which systems offer excessive corrections:

$$Over = \frac{FP_{\text{un}}}{TP + FP} \quad (4)$$

3.3 Comprehensive Score

Once the four disentangled scores have been computed, they need to be merged into a comprehensive score that encapsulates the global performance of the systems. We employ a weighted summation approach to organize these four scores for interpretability and simplification. By definition, systems with higher hit-correction scores are usually preferable, a tendency that inversely applies to the

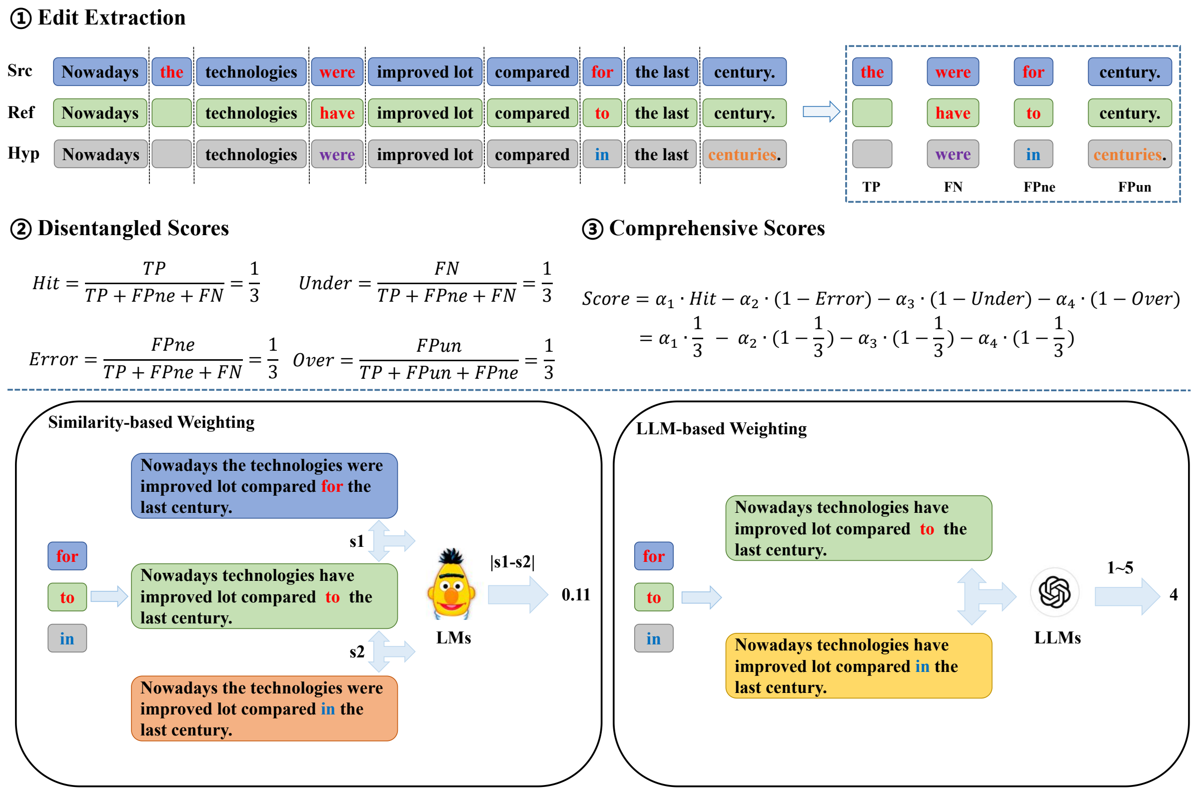

Figure 2: Overview of our approach CLEME2.0. First, we extract the hypothesis edits and reference edits and divide them into TP, FP_{ne}, FP_{un}, and FN edits. Second, we calculate four disentangled scores. Third, we combine them into a comprehensive score. Additionally, we leverage two edit weighting techniques.

remaining scores. Thus, the comprehensive score can be calculated using the following formula:

$$\begin{aligned} Score &= \alpha_1 \cdot Hit + \alpha_2 \cdot (1 - Error) \\ &+ \alpha_3 \cdot (1 - Under) + \alpha_4 \cdot (1 - Over) \end{aligned} \tag{5}$$

where α_i is the trade-off factor for each disentangled score, and we constrain that $0 < \alpha_i < 1$ and $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$.

3.4 Edit Weighting

Existing reference-based metrics, such as ERRANT and CLEME, depend heavily on superficial literal similarity. This means that, regardless of length or modification, all types of edits have equal weighting in the evaluation scores. This aspect fails to acknowledge that human evaluators might semantically consider the edits' varying importance levels. Therefore, we introduce two distinct edit weighting techniques to compute the importance weights of edits. These weights are then incorporated into the calculation of the aforementioned disentangled scores as depicted in Equation (1) $\sim$ (4). Take the hit-correction score as a typical example, we reformulate the Equation (1) as follow:

$$Hit = \frac{w_{TP}}{w_{TP} + w_{FP_{\text{ne}}} + w_{FN}} \tag{6}$$

Similarity-based weighting. We use PTScore from Gong et al. (2022) to provide edit weights. Through simulating a partially accurate version X' of the source sentence X, PTScore can assign individual weights to edits, in spite of multiple edits in a sentence. Since it performs based on BERTScore (Zhang et al., 2019) designed to compute similarity scores for text generation, we call this technique as similarity-based weighting. The computation process is as follows:

$$X' = \text{replace}(X, e_{\text{hyp}}) \tag{7}$$

$$w = |\,\text{PTScore}(X', R) - \text{PTScore}(X, R)| \tag{8}$$

where the function $\text{replace}()$ is intended to replace a specific chunk of the source X with the corrected/dummy hypothesis chunk e_{hyp}. Here, R denotes the reference sentence. Comprehensive details can be found in Gong et al. (2022).

LLM-based weighting. In light of the impressive semantic understanding capabilities of LLMs,

Metric		CoNLL-2014		BN-10GEC		E-Minimal		E-Fluency		NE-Minimal		NE-Fluency		Avg.
		EW	TS	EW	TS	EW	TS	EW	TS	EW	TS	EW	TS	
M^2	γ	0.623	0.672	0.547	0.610	0.597	0.650	0.590	0.659	0.575	0.634	0.582	0.649	0.616
	ρ	0.687	0.720	0.648	0.692	0.654	0.703	0.654	0.709	0.577	0.648	0.648	0.703	0.670
GLEU	γ	0.701	0.750	0.678	0.761	0.533	0.513	0.693	0.771	-0.044	-0.113	0.674	0.767	0.557
	ρ	0.467	0.555	0.754	0.806	0.577	0.511	0.710	0.757	-0.005	-0.055	0.725	0.819	0.551
ERRANT	γ	0.642	0.688	0.586	0.644	0.578	0.631	0.594	0.663	0.585	0.637	0.597	0.659	0.625
	ρ	0.659	0.698	0.637	0.698	0.742	0.786	0.720	0.775	0.747	0.797	0.753	0.797	0.734
PT-M^2	γ	0.693	0.737	0.650	0.706	0.626	0.667	0.621	0.681	0.630	0.675	0.620	0.682	0.666
	ρ	0.758	0.769	0.690	0.824	0.709	0.736	0.758	0.802	0.736	0.758	0.758	0.802	0.758
CLEME-dep	γ	0.648	0.691	0.602	0.656	0.594	0.644	0.589	0.654	0.595	0.643	0.612	0.673	0.633
	ρ	0.709	0.742	0.692	0.747	0.797	0.813	0.714	0.775	0.786	0.835	0.720	0.791	0.760
CLEME-ind	γ	0.649	0.691	0.609	0.659	0.593	0.643	0.587	0.653	0.601	0.647	0.611	0.672	0.635
	ρ	0.709	0.731	0.692	0.747	0.791	0.802	0.731	0.791	0.797	0.841	0.714	0.786	0.761
CLEME2.0-dep (Ours)	γ	0.700	0.765	0.675	0.745	0.690	0.768	0.695	0.788	0.702	0.778	0.704	0.800	0.734
	ρ	0.665	0.736	0.626	0.692	0.736	0.808	0.742	0.830	0.775	0.846	0.599	0.714	0.730
CLEME2.0-ind (Ours)	γ	0.718	0.777	0.731	0.793	0.708	0.784	0.736	0.824	0.757	0.826	0.801	0.848	0.775
	ρ	0.665	0.736	0.698	0.758	0.736	0.808	0.742	0.830	0.775	0.846	0.670	0.769	0.753
CLEME2.0-sim-dep (Ours)	γ	0.783	0.853	0.721	0.801	0.765	0.834	0.737	0.827	0.761	0.824	0.741	0.834	0.790
	ρ	0.819	0.890	0.802	0.863	0.791	0.868	0.758	0.852	0.830	0.896	0.786	0.857	0.834
CLEME2.0-sim-ind (Ours)	γ	**0.806**	**0.871**	**0.772**	**0.839**	**0.780**	**0.841**	**0.761**	**0.844**	**0.782**	**0.834**	**0.798**	**0.877**	**0.817**
	ρ	**0.846**	**0.901**	**0.835**	**0.885**	**0.819**	**0.885**	**0.758**	**0.852**	**0.846**	**0.896**	**0.863**	**0.923**	**0.859**
SentM^2	γ	0.871	0.864	0.567	0.646	0.805♣	0.836♣	0.655	0.732	0.729♣	0.785♣	0.621	0.699	0.734
	ρ	0.731	0.758	0.593	0.648	0.806♣	0.845♣	0.731	0.764	0.797♣	0.846♣	0.632	0.687	0.737
SentGLEU	γ	0.784	0.828	0.756	0.826	0.742♣	0.773♣	0.785	0.846	0.723♣	0.762♣	0.778	0.848	0.788
	ρ	0.720	0.775	0.769	0.824	0.764♣	0.797♣	0.791	0.846	0.764♣	0.830♣	0.768	0.846	0.791
SentERRANT	γ	0.870	0.846	0.885	0.896	0.768♣	0.803♣	0.806	0.732	0.710♣	0.765♣	0.793	0.847	0.810
	ρ	0.742	0.747	0.786	0.830	0.775♣	0.819♣	0.813	0.764	0.780♣	0.841♣	0.830	0.857	0.799
SentPT-M^2	γ	**0.949**	**0.938**	0.602♣	0.682♣	0.831♣	0.855♣	0.689	0.763	0.770♣	0.822♣	0.648	0.725	0.772
	ρ	0.907	0.874	0.626♣	0.670♣	0.808♣	0.819♣	0.797	0.841	0.813♣	0.857♣	0.742	0.786	0.795
SentCLEME-dep	γ	0.876	0.844	**0.915**	**0.913**	0.806♣	0.838♣	0.849	0.886	0.742♣	0.795♣	0.876	0.921	0.855
	ρ	0.824	0.808	**0.835**	**0.874**	0.775♣	0.819♣	0.824	0.863	0.797♣	0.846♣	0.791	0.846	0.825
SentCLEME-ind	γ	0.868	0.857	0.855♣	0.876♣	0.821♣	0.856♣	0.841	0.877	0.782♣	0.831♣	0.852	0.896	0.851
	ρ	0.725	0.758	0.659♣	0.714♣	0.775♣	0.819♣	0.808	0.846	0.819♣	0.874♣	0.762	0.825	0.782
SentCLEME2.0-dep (Ours)	γ	0.870	0.881	0.766	0.830	0.941♣	0.954♣	0.892	0.938	0.913♣	**0.918♣**	0.916	0.949	0.897
	ρ	0.714	0.725	0.681	0.747	0.857♣	0.885♣	0.824	0.901	**0.857♣**	**0.912♣**	0.720	0.791	0.801
SentCLEME2.0-ind (Ours)	γ	0.866	0.881	0.799	0.853	0.941♣	0.956♣	0.915	0.952	**0.915♣**	0.917♣	0.883	0.904	0.899
	ρ	0.709	0.720	0.681	0.747	**0.879♣**	**0.912♣**	0.857	0.923	0.824♣	0.885♣	0.654	0.720	0.793
SentCLEME2.0-sim-dep (Ours)	γ	0.926	0.937	0.797	0.861	0.939♣	0.948♣	0.908	0.952	0.871♣	0.872♣	0.918	0.947	0.906
	ρ	**0.907**	**0.912**	0.808	0.863	0.852♣	0.879♣	**0.885**	0.945	0.753♣	0.780♣	**0.896**	**0.940**	**0.868**
SentCLEME2.0-sim-ind (Ours)	γ	0.915	0.936	0.808	0.866	**0.945♣**	**0.956♣**	**0.923**	**0.963**	0.885♣	0.887♣	**0.931**	**0.961**	**0.915**
	ρ	0.868	0.879	0.753	0.824	0.863♣	0.901♣	0.879	**0.956**	0.775♣	0.802♣	0.835	0.923	0.855

Table 1: Correlation results on GJG15 Ranking. We highlight the **highest** score in bold and the second-highest score with underlines. ♣ We remove unchanged reference sentences for higher correlations due to low-quality annotations. Otherwise, negative correlations are possible.

some recent work has sought their use in evaluating assorted NLP tasks (Pavlovic and Poesio, 2024; Chen et al., 2024), GEC evaluations included (Sottana et al., 2023). Drawing inspiration from this trend, we employ Llama-2-7B (Touvron et al., 2023) as a scorer to acquire the importance score for each edit. These scores range from 1 to 5, with higher scores indicating increased edit importance. We provide the implementation details and the prompting template in Appendix D.

4 Experiments

4.1 Experimental Settings

Human ranking datasets. We conduct comprehensive experiments across two human judgment datasets with disparate annotation protocols.

- **GJG15** (Grundkiewicz et al., 2015) is constructed to manually evaluate classical systems (Junczys-Dowmunt and Grundkiewicz, 2014; Rozovskaya et al., 2014) in the CoNLL-2014 shared task (Ng et al., 2014).
- **SEEDA**. Kobayashi et al. (2024b) reveal several shortcomings in GJS15 and subsequently propose SEEDA, an alternative dataset featuring human judgments across two levels of granularity. To align with the contemporary trend in GEC, SEEDA is primarily focused on mainstream neural-based systems.

Both of human judgment datasets derive the overall human rankings for all GEC systems by employing Expected Wins (EW) (Bojar et al., 2013) and TrueSkill (TS) (Sakaguchi et al., 2014) methods. Following the previous approaches (Ye et al., 2023; Kobayashi et al., 2024b), we compute the Pearson (γ) and Spearman (ρ) correlations between metrics and human judgments, in order to ascertain the effectiveness and robustness of GEC metrics within the context of *system-level ranking*.

Reference datasets. Reference-based metrics rely on a reference set to establish a system ranking list, the properties of which may significantly influence the performance of the metrics. To investigate the impact of variable reference sets, we assess human consistency across 6 reference datasets. These datasets encompass a range of annotation styles, and a number of human annotators, including CoNLL-2014 (Grundkiewicz et al., 2015), BN-10GEC (Bryant and Ng, 2015) and SN-8GEC (Sakaguchi et al., 2016). Notably, SN-8GEC is partitioned into 4 sub-sets, namely E-Minimal, E-Fluency, NE-Minimal, and NE-Fluency. A more thorough breakdown of these datasets and the statistics are provided in Appendix A.

Corpus and sentence levels. GEC evaluation metrics can compute an overall system-level score for a given system in two settings (Gong et al., 2022). Given the metric M, source sentences $\mathbf{S}$, hypothesis sentences $\mathbf{H}$ and reference sentences $\mathbf{R}$, 1) **corpus-level** metrics compute the system score based on the whole corpus $M(\mathbf{S}, \mathbf{H}, \mathbf{R})$, and 2) **sentence-level** metrics use the average of the sentence-level scores $\sum_i^I M(\mathbf{S}_i, \mathbf{H}_i, \mathbf{R}_i)/I$.

Trade-off factors. We employ a cross-evaluation approach to determine two optimal configurations for trade-off factors applicable at the corpus and sentence levels, respectively. At the corpus level, we assign the factors as $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.45, 0.35, 0.15, 0.05$. Conversely, at the sentence level, these are adjusted to $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.35, 0.25, 0.20, 0.20$. Additionally, CLEME2.0 metrics named with "sim" are based on similarity-based edit weighting, and we leave the analysis of LLM-based edit weighting to Section 5.2.

Evaluation Assumptions. Ye et al. (2023) propose evaluating systems based on one of two assumptions, namely correction dependence and correction independence. In short, the correction independence assumption offers a more relaxed edit matching process, implying that systems are more inclined to yield higher scores when multiple

Metric	SEEDA-S		SEEDA-E		Avg.
	γ	ρ	γ	ρ	
M^2	0.658	0.487	0.791	0.764	0.675
PT-M^2	0.845	0.769	0.896	0.909	0.855
ERRANT	0.557	0.406	0.697	0.671	0.583
PT-ERRANT	0.818	0.720	0.888	0.888	0.829
GoToScorer	0.929	0.881	0.901	0.937	0.912
GLEU	0.847	0.886	0.911	0.897	0.885
Scribendi Score	0.631	0.641	0.830	0.848	0.738
SOME	0.892	0.867	0.901	0.951	0.903
IMPARA	0.911	0.874	0.889	0.944	0.903
CLEME-dep	0.633	0.501	0.755	0.757	0.662
CLEME-ind	0.616	0.466	0.736	0.708	0.632
CLEME2.0-dep (Ours)	**0.937**	0.865	0.945	0.939	0.922
CLEME2.0-ind (Ours)	0.908	0.844	**0.961**	0.946	0.915
CLEME2.0-sim-dep (Ours)	0.923	**0.914**	0.948	0.974	0.940
CLEME2.0-sim-ind (Ours)	0.921	0.907	0.953	**0.981**	**0.941**
Sent-M^2	0.802	0.692	0.887	0.846	0.807
SentERRANT	0.758	0.643	0.860	0.825	0.772
SentCLEME-dep	0.866	0.809	0.944	0.939	0.890
SentCLEME-ind	0.864	0.858	0.935	0.911	0.892
SentCLEME2.0-dep (Ours)	0.905	0.844	0.955	0.946	0.913
SentCLEME2.0-ind (Ours)	0.875	0.837	0.953	0.953	0.905
SentCLEME2.0-sim-dep (Ours)	**0.924**	0.858	0.923	0.953	0.915
SentCLEME2.0-sim-ind (Ours)	0.921	**0.886**	**0.957**	**0.960**	**0.931**

Table 2: Results of human correlations on SEEDA Ranking based on TrueSkill (TS).

Metric	EW		TS		Avg.
	γ	ρ	γ	ρ	
CLEME2.0-dep-Hit	0.599	0.593	0.673	0.648	0.628
CLEME2.0-dep-Error	-0.444	-0.533	-0.526	-0.593	-0.524
CLEME2.0-dep-Under	0.496	0.599	0.576	0.659	0.583
CLEME2.0-dep-Over	0.118	0.269	0.073	0.275	0.253
SentCLEME2.0-dep-Hit	0.594	0.593	0.672	0.648	0.627
SentCLEME2.0-dep-Error	-0.405	-0.429	-0.489	-0.500	-0.456
SentCLEME2.0-dep-Under	0.489	0.511	0.572	0.582	0.539
SentCLEME2.0-dep-Over	-0.247	-0.363	-0.346	-0.440	-0.349

Table 3: Correlation results of each disentangled score on GJG15 Ranking.

references are accessible. Inspired by this work, CLEME2.0 also supports both assumptions, and we will study their impact on our method.

4.2 Results of GJG15 Ranking

The correlations between the GEC metrics and human judgments on the GJG15 rankings are shown in Table 1, and we have the following insights.

CLEME2.0 outperforms other metrics at both corpus and sentence levels. For corpus-level, CLEME2.0-sim-ind achieves the highest average correlations, closely followed by CLEME2.0-sim-dep. CLEME2.0-ind and CLEME2.0-dep can also gain comparable correlations with other metrics, in spite of the fact that they do not utilize any

Metric	EW		TS		Avg.
	γ	ρ	γ	ρ	
CLEME2.0-dep	0.461	0.423	0.483	0.457	0.456
CLEME2.0-ind	0.468	0.421	0.489	0.453	0.458
CLEME2.0-sim-dep	0.559	0.592	0.581	**0.624**	0.589
CLEME2.0-sim-ind	**0.566**	**0.593**	**0.588**	0.622	**0.592**
SentCLEME2.0-dep	0.374	0.305	0.362	0.290	0.333
SentCLEME2.0-ind	0.372	0.302	0.356	0.283	0.328
SentCLEME2.0-sim-dep	0.410	**0.361**	**0.400**	**0.345**	**0.379**
SentCLEME2.0-sim-ind	**0.412**	0.360	0.399	0.338	0.377

Table 4: Average correlations of (Sent)CLEME2.0 and (Sent)CLEME2.0-sim on CoNLL-2014.

Dataset	Corpus-EW		Corpus-TS		Sentence-EW		Sentence-TS	
	γ	ρ	γ	ρ	γ	ρ	γ	ρ
CoNLL-2014	0.697	0.659	0.759	0.720	0.626	0.654	0.696	0.698
BN-10GEC	0.732	0.764	0.796	0.813	0.638	0.637	0.708	0.698
E-Minimal	0.709	0.786	0.779	0.819	0.642	0.692	0.715	0.747
E-Fluency	0.760	0.786	0.831	0.841	0.642	0.665	0.720	0.714
NE-Minimal	0.777	0.823	0.839	0.861	0.654	0.747	0.723	0.791
NE-Fluency	0.823	0.692	0.849	0.709	0.664	0.791	0.742	0.830

Table 5: Correlation results of LLM-based weighting on GJG15 Ranking.

edit weighting techniques. On the other hand, sentence-level metrics exhibit a similar pattern. SentCLEME2.0-sim-dep and SentCLEME2.0-sim-ind achieve the highest Pearson and Spearson correlations, respectively. These results significantly demonstrate the effectiveness and robustness of our proposed method across different settings.

Sentence-level metrics outperform their corpus-level counterparts. This observation aligns with recent studies (Gong et al., 2022; Ye et al., 2023). This is because system-level rankings treat each sample equally, which is consistent with the evaluation process of sentence-level metrics. In contrast, corpus-level metrics allow samples with more edits to significantly influence the results. SentPT-M^2 exhibits the best performance on CoNLL-2014, but its results on BN-10GEC, E-Minimal, and NE-Fluency are inferior compared to our method.

In general, our method aligns more consistently with human judgments than existing mainstream metrics. Particularly, most weighted outcomes outshine the unweighted ones, attributable to the incorporation of semantic considerations. However, on E-Minimal and NE-Minimal, the unweighted and weighted results exhibit comparability. We conjecture that this could be due to the annotations in these datasets being minimal yet decisive, reducing the possibility of generating diverse weights.

4.3 Results of SEEDA Ranking

We conduct a supplementary experiment on the SEEDA-Sentence and SEEDA-Edit datasets, comparing our method with a wide range of GEC metrics. Table 2 demonstrates again that our approach obtains the optimal results on both datasets. Kobayashi et al. (2024b) claim that the correlations of most metrics tend to decline when shifting from classical to neural systems in evaluation. This suggests that traditional metrics may struggle when assessing more extensively edited and fluent corrections generated by neural systems. However, our method is still able to address these challenges and obtain even better results. The results on SEEDA-Edit surpass those on SEEDA-Sentence due to the finer granularity of SEED-Edit, which is more consistent with the functioning of CLEME2.0.

It is crucial to mention that reference-less metrics such as SOME and IMPARA yield high outcomes, in part, because these are fine-tuned on GEC data. Although fine-tuned metrics generally perform better, they are not without their limitations. Firstly, the incorporation of fine-tuning in SOME and IMPARA makes these reference-less metrics more costly. Second, these reference-less metrics may suffer from poor robustness since the assessment process is not guided by human-annotated references. For example, the authors of Scribendi Score claim that it can achieve high correlations on the human judgment dataset from Napoles et al. (2016b). However, only moderate correlations are observable on SEEDA-Edit.

5 Analysis

5.1 Ablation Studies

Isolated effect of each disentangled score. We perform ablation experiments on (Sent)CLEME2.0-dep to observe how each disentangled score performs. Since a system exhibiting lower error-correction, under-correction, and over-correction is more desirable, these scores are subtracted from 1. The results are presented in Table 3. Both hit-correction and under-correction scores display moderate correlations. Over-correction scores exhibit slight positive correlations at the corpus-level, but negligible negative correlations at the sentence level. Interestingly, error-correction scores manifest negative correlations. However, this does not imply that error-correction scores have no influence on the comprehensive score. In fact, the trade-off factor of error-correction scores is relatively sig-

	Chunk 1	Chunk 2	Chunk 3	Chunk 4	Chunk 5	Chunk 6	Chunk 7	Chunk 8	Chunk 9
Source	When we are	diagonosed out	with certain genetic	disease	, are we suppose to disclose	this result	to	our	relatives ?
Ref.	When we are	diagnosed	with certain genetic	diseases	, are we suppose to disclose	this result	to	our	relatives ?
Hyp.	When we are	diagnosed out (0.056)	with certain genetic	diseases (0.006)	, are we suppose to disclose	the results (0.019)	to	their (0.021)	relatives ?

	Chunk 1	Chunk 2	Chunk 3	Chunk 4	Chunk 5	Chunk 6
Source	Do one	who	suffered	from this disease keep it a secret	of infrom	their relatives ?
Ref	Does one	who	suffers	from this disease keep it a secret	or inform	their relatives ?
Hyp.	Do one (0.028)	who	suffer (0.011)	from this disease keep it a secret	to inform (0.094)	their relatives ?

Table 6: Cases of CLEME2.0. We highlight TP chunks, FP_{ne} chunks, FP_{un} chunks, and FN chunks in different colors. Fractions in brackets in Hyp. are similarity-based weighting scores.

nificant. It is hypothesized that evaluations based solely on error-correction scores could unduly encourage systems that produce only highly confident edits, resulting in potential evaluation bias.

Average correlations. To further compare different metrics from a global perspective, we report the average correlations obtained through the exhaustive enumeration of various parameter configurations. Specifically, all possible parameter combinations are attempted, with a step increment of 0.05. From Table 4, we observe that all the correlations are positive, regardless of the applied correction assumptions, evaluation levels, and weighting techniques. Comparing the unweighted and similarity-based weighted results, we conclude that similarity-based weighting significantly promotes human correlations, even on a global scale. Furthermore, corpus-level metrics tend to attain higher average results than sentence-level metrics; nonetheless, sentence-level metrics with optimal parameters surpass their corpus-level counterparts. This suggests that corpus-level metrics may exhibit enhanced robustness concerning parameter selection.

5.2 Exploration of LLM

The results of LLM-based edit weighting are shown in Table 5. The corpus-level results are quite satisfying and are comparable to those of most metrics such as PT-M2 and CLEME. However, the sentence-level outcomes are less satisfactory. This could be due to the fact that LLM assigns error-editing scores on a scale of 1 to 5, which is notably coarser when contrasted with the 0 to 1 continuous scoring scale. Sentence-level scores depend on the mean of the editing scores within a particular sentence. Consequently, even the slightest bias in the scores assigned by the LLM might lead to significant deviations in the sentence-level outcomes.

Although the LLM has had some success, its performance still falls short when compared to the similarity-based weighting. This might be due to the scoring granularity of the 1 to 5 scale provided by the LLM not being sufficiently fine-tuned. In addition, the score heavily relies on the functionality of the LLM, which proves rather unstable.

5.3 Case Study

Table 6 presents examples of CLEME2.0. In the top group, chunk 2 achieves the highest score (0.056), highlighting its significant impact on the sentence. Although "diagnosed" was correctly amended, the omission of "out" persists, rendering the sentence still incorrect. Chunk 4 represents a hit-correction relating to the singular and plural forms in the source sentence and its low score indicates a minimal impact. Chunks 6 and 8 are types of overcorrection. Chunk 6 does not change the original meaning, whereas chunk 8 introduces a more severe error due to an incorrect personal pronoun. In the second group, both chunks 3 and 5 exhibit error-corrections, with chunk 5 scoring higher than chunk 3. Chunk 3 involves issues of tense and singular-plural, while chunk 5 presents a more serious error that completely alters the meaning of the sentence. The weighting scores reflect the superiority of the method. For metrics that do not apply weightings, sorts of edits are uniformly assigned, which does not reflect the actual semantics.

6 Conclusion

This paper proposes CLEME2.0, an interpretable evaluation strategy, which are beneficial to reveal crucial characteristics of GEC systems. To address the limitations of traditional reference-based metrics in capturing semantic nuances, we enhance CLEME2.0 using two innovative edit weighting techniques: similarity-based and LLM-based weighting. Through comprehensive experiments and analyses, we validate the efficacy and reliability of our approach. We believe CLEME2.0 will provide a promising perspective on the task of grammatical error correction.

Limitation

Although CLEME2.0 can be extended to other languages, its effectiveness has not been tested in any language other than English. Furthermore, all reference sets utilized in our experiments are derived from the CoNLL-2014 shared task, which is a second language dataset. To demonstrate the robustness of our approaches, further experiments on evaluation datasets encompassing multiple languages and text domains are required. Finally, we strongly advocate for the construction of new GEC evaluation datasets to advance the development of NLP.

Ethics Statement

In this paper, we validate the effectiveness and robustness of our proposed approach using the CoNLL-2014, BN-10GEC, and SN-8GEC reference datasets. These datasets are sourced from publicly available resources on legitimate websites and do not contain any sensitive data. Additionally, all the baselines employed in our experiments are publicly accessible GEC metrics, and we have duly cited the respective authors. We confirm that all datasets and baselines utilized in our experiments are consistent with their intended purposes.

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.

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, 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.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3):643–701.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or llms as the judge? a study on judgement biases. *arXiv preprint arXiv:2402.10669*.

Leshem Choshen and Omri Abend. 2018. 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.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction. *arXiv preprint arXiv:2303.14342*.

Daniel Dahlmeier and Hwee Tou Ng. 2012a. 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.

Daniel Dahlmeier and Hwee Tou Ng. 2012b. 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.

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.

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

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.

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.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. The amu system in the conll-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024a. Large language models are state-of-the-art evaluator for grammatical error correction. *arXiv preprint arXiv:2403.17540*.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024b. Revisiting meta-evaluation for grammatical error correction. *arXiv preprint arXiv:2403.02674*.

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, et al. 2022. Linguistic rules-based corpus generation for native chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589.

Nitin Madnani, Martin Chodorow, Joel Tetreault, and Alla Rozovskaya. 2011. They can help: Using crowdsourcing to improve the evaluation of grammatical error detection systems. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 508–513.

Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. 2022. Impara: Impact-based metric for gec using parallel data. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3578–3588.

Masato Mita, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, and Kentaro Inui. 2019. Cross-corpora evaluation and analysis of grammatical error correction models—is single-corpus evaluation enough? In *Proceedings of NAACL-HLT*, pages 1309–1314.

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.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016a. There's no comparison: Reference-less evaluation metrics in grammatical error correction. *arXiv preprint arXiv:1610.02124*.

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.

Maja Pavlovic and Massimo Poesio. 2024. The effectiveness of llms as annotators: A comparative overview and empirical analysis of direct representation. *arXiv preprint arXiv:2405.01299*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, and Dario Amodei. 2019. Gpt 2; language models are unsupervised multitask learners. In *2019 by OpenAI*.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The illinois-columbia system in the conll-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 34–42.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of gpt-4: reliably evaluating large language models on sequence to sequence tasks. *arXiv preprint arXiv:2310.13800*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jingheng Ye, Yinghui Li, Shirong Ma, Rui Xie, Wei Wu, and Hai-Tao Zheng. 2022. Focus is what you need for chinese grammatical error correction. *arXiv preprint arXiv:2210.12692*.

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.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020a. 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.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020b. 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.

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

A Details about GEC Meta-Evaluation

A.1 Human Rankings

GJG15 ranking. Grundkiewicz et al. (2015) propose the first large-scale human judgement dataset towards 12 participating systems of the CoNLL-2014 shared task. In this assessment, 8 native speaker are asked to rank the outputs of all the systems from best to worst. Two system ranking lists are generated using Expected Wins (EW) and TrueSkill (TS) respectively. Since all the involved systems are mostly classical systems such as statistical machine translation approaches (Junczys-Dowmunt and Grundkiewicz, 2014) and classifier-based approaches (Rozovskaya et al., 2014), the dataset may not be a ideal test bed for meta-evaluation in the current time.

SEEDA ranking. Kobayashi et al. (2024b) identify several limitations of GJG15 ranking dataset, and propose a new human ranking dataset called SEEDA. SEEDA consists of corrections with human ratings along two different granularities: edit-based and sentence-based, covering 12 state-of-the-art systems including large language models (LLMs), and two human corrections with different focuses. Three native English speakers participate in the annotation process. Similar to Grundkiewicz et al. (2015), the overall human rankings are derived from TrueSkill (TS) and Expected Wins (EW) based on pairwise judgments.

B Statistics of Reference Datasets

Table 7 presents the statistics of all the reference sets involved in our experiments.

B.1 Baseline Metrics

In our evaluation, we compare our method with the following reference-based baseline metrics, including corpus and sentence-level variants:

- **M^2** and **SentM^2** (Dahlmeier and Ng, 2012a) dynamically extract the hypothesis edits with the maximum overlap of gold annotations by utilizing the Levenshtein algorithm.

- **GLEU** and **SentGLEU** (Napoles et al., 2015) are BLEU-like GEC metrics based on n-gram matching, rewarding hypothesis n-grams that align with the reference but not the source, while penalizing those aligning solely with the source. GLEU is the main metric in JFLEG, an English GEC dataset that highlights holistic fluency edits.

- **ERRANT** and **SentERRANT** (Bryant et al., 2017) are the most mainstream GEC metrics, which are based on edit matching. They are able to extract edits more accurately, by utilizing the linguistically enhanced Damerau-Levenshtein algorithm.

- **PT-M^2** and **SentPT-M^2** (Gong et al., 2022) leverage pre-trained language model (PLM) to evaluate GEC systems. The main idea is similar to M^2 and ERRANT, but they can leverage the knowledge of pre-trained language models to score edits effectively.

- **CLEME** and **SentCLEME** (Ye et al., 2023) are proposed to provide unbiased scores for multi-reference evaluation. Besides, the authors introduce the correction independence assumption, so CLEME can perform based on either traditional correction dependence or correction independence assumptions.

In addition, for the evaluation of SEEDA, we have additionally added the following evaluation methods in accordance with the evaluation methods reported in Kobayashi et al. (2024b):

- **GoToScorer** (Gotou et al., 2020): takes into account the difficulty of error correction when calculating the evaluation score. The difficulty is calculated based on the number of systems that can correct errors.

- **Scribendi Score** (Islam and Magnani, 2021): evaluates in conjunction with the complexity

Item	CoNLL-2014	BN-10GEC	E-Minimal	E-Fluency	NE-Minimal	NE-Fluency
# Sents (Length)	1,312 (23.0)	1,312 (23.0)	1,312 (23.0)	1,312 (23.0)	1,312 (23.0)	1,312 (23.0)
# Refs (Length)	2,624 (22.8)	13,120 (22.9)	2,624 (23.2)	2,624 (22.8)	2,624 (23.0)	2,624 (22.2)
# Edits (Length)	5,937 (1.0)	36,677 (1.0)	4,500 (1.0)	8,373 (1.1)	4,964 (0.9)	11,033 (1.2)
# Unchanged Chunks (Length)	11,174 (4.8)	93,496 (2.5)	8,887 (6.3)	12,823 (3.8)	10,748 (5.1)	14,086 (2.9)
# Corrected/Dummy Chunks (Length)	4,994 (1.3)	26,948 (2.4)	3,963 (1.2)	6,305 (1.7)	4,221 (1.2)	6,892 (2.6)

Table 7: Statistics of CoNLL-2014 (Ng et al., 2014), BN-10GEC (Bryant and Ng, 2015) and SN-8GEC (Sakaguchi et al., 2016) reference sets. We leverage ERRANT (Bryant et al., 2017) for edit extraction, and CLEME (Ye et al., 2023) for chunk extraction.

calculated by GPT-2 (Radford et al., 2019), the labeled ranking ratio and the Levenstein distance ratio.

- **SOME** (Yoshimura et al., 2020b): optimizes human evaluation by fine-tuning BERT separately for criteria such as grammaticality, fluency, and meaning preservation.
- **IMPARA** (Maeda et al., 2022): incorporates a quality assessment model fine-tuned using BERT parallel data and a similarity model that takes into account the effects of editing.

C Detailed Results of Evaluation

We list detailed evaluation results of CLEME2.0 on CoNLL-2014 in Table 8.

D Experimental Details of LLM-based Edit Weighting

Due to the strong semantic understanding capabilities of large language models (LLMs), recent work (Sottana et al., 2023) has sparked interest in using LLMs for text evaluation, including the evaluation of grammatical error correction. Inspried by this, we utilize LLMs as weighted scorers to assess the importance of each edit. The template for the LLM is shown in Figure 3. For each edit, the constructed sentence contains only one grammatical error, while the other positions are correct. The second line shows the modification of that edit. The LLM is required to determine the necessity of the modified edit and output a score from 1 to 5. A higher score indicates a greater necessity for the edit modification. We do not inform the LLM of the specific types of edits; instead, we let the larger model evaluate the necessity of the modified edits.

D.1 Hit-Correction Edits

Scenario: The hypothesis and reference sentence are consistent.
Focus: The significance of the transition from the source to the hypothesis sentence.
Scoring: A higher score indicates that the edit from source to reference sentence carries substantial importance. Conversely, a lower score suggests that this transition is less crucial.

D.2 Error-Correction Edits

Scenario: The hypothesis and reference sentence are inconsistent.
Focus: The significance of the transition from the hypothesis to the reference sentence.
Scoring: A high score indicates a critical edit, suggesting significant inaccuracies in the hypothesis sentence. A low score implies that the modification is of minimal importance, indicating the hypothesis sentence is either correct or not substantially incorrect.

D.3 Under-Correction Edits

Scenario: The source and hypothesis sentence remain unchanged.
Focus: The importance of modifications from the source to the reference sentence.
Scoring: A high score implies a critical need for the edit, pointing to a severe under-correction. Conversely, a low score indicates that the edit is of lesser importance, suggesting a mild under-correction.

D.4 Over-Correction Edits

Scenario: The source is equivalent to the reference sentence, leading to two distinct situations:

1. The reference is not an ideal sentence, and the hypothesis sentence is corrected but deemed overcorrected.
2. The reference is optimal, necessitating no amendments, yet the hypothesis sentence introduces corrections.

Evaluation:

Prompt:
As a grammar correction evaluator, you are required to score the corrected editors for each grammatical error. We will give three lines, the first line is the original sentence given, the second line is the modification made to the editor, and the third line is the output form.
The scoring range is 1-5. The larger the score, the more important the editor's correction is. Correspondingly, the smaller the score, the less important the editor's correction is.
1 point indicates that this editor's modification has almost no impact on the original sentence and is dispensable.
2 points indicates that this editorial change has a slight impact.
3 points indicates that this editor's changes have a certain impact.
4 points indicates that this editorial change is necessary.
5 points indicates that this editing modification is very necessary and of high importance.
The output format is a score of 1 to 5 points.
Next, I will give you a sentence only with an edit. You need to rate each edit in sequence. The desired output is just a score, without any redundant explanation.
Example Input:
Sentence: Nowadays the technologies were improved a lot compared to the last century.
Edit: were => have
Output (1-5):
Example Output:
5
Note that the output must be a number between 1 and 5. Here is the sample:

Figure 3: The prompting of LLM-based weighting.

- **First Situation**: Assess the importance (W_1) of the edit from the source to the hypothesis sentence. A higher W1 score indicates that the edit is crucial, suggesting imperfections in the reference sentence. Conversely, a lower score suggests that the edit is of minimal importance, rendering the hypothesis's correction unnecessary.

- **Second Situation**: Examine the significance (W_2) of the edit from the hypothesis to the reference sentence. A higher score indicates that the edit is critical, denoting that the hypothesis's correction was overly aggressive. A lower score implies the edit was unneeded, making the correction by the hypothesis irrelevant.

Formula: The computation of over-correction score is defined as follow:

$$\text{over-correction score} = W_2 - W_1$$

This score can be either positive or negative. A higher over-correction score signals a less effective performance by the correction system.

By systematically assessing the necessity and importance of different types of edits, we can better understand and improve the performance of grammatical error correction systems.

Metric		AMU	CAMB	CUUI	IITB	INPUT	IPN	NTHU	PKU	POST	RAC	SJTU	UFC	UMC
CLEME2.0-dependent	**TP**	380	584	471	22	0	39	330	246	412	254	85	32	260
	sim	9.20	12.66	7.58	0.39	0.00	0.77	5.79	6.69	8.80	6.68	1.50	0.42	5.29
	FP	817	1307	964	67	0	488	905	709	1145	782	272	18	789
	sim	16.03	30.92	16.06	1.80	0.00	11.93	24.56	14.36	19.25	11.98	6.49	0.25	18.26
	FP$_{ne}$	276	418	311	34	0	149	302	254	316	259	76	12	245
	sim	4.08	6.55	3.68	0.75	0.00	4.61	5.89	4.06	4.60	3.80	2.30	0.17	3.83
	FP$_{un}$	541	889	653	33	0	339	603	455	829	523	196	6	544
	sim	11.95	24.36	12.38	1.05	0.00	7.33	18.67	10.30	14.64	8.18	4.19	0.08	14.43
	FN	1360	1150	1357	2057	1782	2886	1388	1454	1354	1487	1668	2087	1461
	sim	34.25	28.45	36.21	78.39	48.24	83.10	46.53	36.15	34.48	38.00	56.28	51.27	39.60
	TN	6298	6329	6224	6007	6308	5160	6274	6286	6428	6276	6313	5965	6355
	sim	6237	6301	6190	5373	6226	5241	5973	6214	6382	6194	5902	6092	6273
	Hit	0.188	0.271	0.220	0.010	0.00	0.013	0.163	0.126	0.198	0.127	0.046	0.015	0.132
	sim	0.194	0.266	0.160	0.005	0.00	0.009	0.100	0.143	0.184	0.138	0.025	0.008	0.109
	Error	0.137	0.194	0.145	0.016	0.00	0.048	0.150	0.130	0.152	0.130	0.042	0.006	0.125
	sim	0.086	0.138	0.078	0.009	0.00	0.052	0.101	0.0866	0.096	0.078	0.038	0.003	0.079
	Under	0.675	0.534	0.634	0.973	1.00	0.939	0.687	0.744	0.650	0.744	0.912	0.979	0.743
	sim	0.721	0.597	0.763	0.986	1.00	0.939	0.799	0.771	0.720	0.784	0.937	0.989	0.813
	Over	0.452	0.470	0.455	0.371	0.00	0.643	0.488	0.476	0.532	0.505	0.549	0.12	0.519
	sim	0.474	0.559	0.524	0.478	0.00	0.577	0.615	0.490	0.522	0.438	0.524	0.116	0.613
	Score	0.483	0.508	0.497	0.431	0.45	0.408	0.463	0.450	0.479	0.505	0.434	0.450	0.453
	sim	0.503	0.520	0.484	0.425	0.45	0.408	0.439	0.474	0.491	0.438	0.424	0.448	0.452
SentCLEME2.0-dependent	**TP**	376	580	467	22	0	39	327	244	409	251	84	32	259
	sim	9.14	12.63	7.52	0.39	0.00	0.76	5.72	6.65	8.75	6.59	1.48	0.42	5.23
	FP	821	1311	968	67	0	488	908	711	1148	785	273	18	790
	sim	16.49	31.25	16.50	1.85	0.00	13.00	24.83	14.38	19.36	12.34	7.13	0.26	18.47
	FP$_{ne}$	286	431	320	22	0	132	310	262	326	271	81	10	255
	sim	4.60	7.51	4.27	0.44	0.00	2.62	6.58	4.58	5.06	4.02	1.28	0.15	4.39
	FP$_{un}$	535	880	648	45	0	356	598	449	822	514	192	8	535
	sim	11.89	23.74	12.23	1.42	0.00	10.39	18.24	9.80	14.30	8.32	5.85	0.12	14.07
	FN	1600	1374	1577	1972	1982	1940	1660	1712	1587	1744	1900	1980	1714
	sim	43.65	35.92	45.22	57.46	58.31	54.69	46.92	46.02	43.09	46.05	55.32	58.35	48.02
	TN	6058	6105	6004	6092	6108	6106	6002	6028	6195	6019	6081	6072	6102
	sim	6052	6095	6009	6093	6106	6115	5995	6012	6203	6027	6079	6070	6115
	Hit	0.136	0.210	0.163	0.008	0.00	0.013	0.119	0.088	0.142	0.089	0.032	0.012	0.091
	sim	0.131	0.205	0.142	0.007	0.00	0.011	0.104	0.088	0.129	0.086	0.027	0.008	0.087
	Error	0.080	0.129	0.090	0.005	0.00	0.038	0.095	0.076	0.088	0.071	0.023	0.002	0.070
	sim	0.063	0.102	0.066	0.004	0.00	0.033	0.079	0.059	0.070	0.051	0.020	0.001	0.059
	Under	0.500	0.392	0.479	0.675	0.687	0.639	0.496	0.538	0.486	0.551	0.637	0.678	0.546
	sim	0.519	0.419	0.517	0.673	0.684	0.645	0.524	0.553	0.509	0.567	0.641	0.680	0.557
	Over	0.248	0.419	0.293	0.031	0.00	0.242	0.304	0.235	0.342	0.232	0.121	0.006	0.267
	sim	0.241	0.421	0.294	0.030	0.00	0.224	0.302	0.224	0.331	0.203	0.119	0.005	0.267
	Score	0.498	0.513	0.507	0.467	0.466	0.447	0.481	0.475	0.495	0.477	0.469	0.471	0.476
	sim	0.502	0.520	0.504	0.467	0.466	0.449	0.479	0.481	0.494	0.484	0.467	0.469	0.479
CLEME2.0-independent	**TP**	388	596	487	22	0	39	338	248	420	255	85	32	262
	sim	9.47	13.11	7.99	0.40	0.00	0.81	6.13	6.80	9.07	6.91	1.54	0.47	5.49
	FP	809	1295	948	67	0	488	897	707	1137	781	272	18	787
	sim	14.74	28.11	14.42	1.91	0.00	11.82	22.93	13.03	17.62	11.23	6.46	0.25	16.99
	FP$_{ne}$	408	627	449	34	0	234	447	388	487	406	134	12	366
	sim	6.32	10.62	5.51	0.86	0.00	4.79	9.50	7.30	7.12	5.56	2.41	0.17	6.14
	FP$_{un}$	401	668	499	33	0	254	450	319	650	375	138	6	421
	sim	8.42	17.49	8.91	1.05	0.00	7.03	13.43	5.73	10.50	5.67	4.05	0.08	10.85
	FN	1029	778	984	1497	1530	1382	1045	1129	989	1135	1398	1506	1136
	sim	26.88	20.31	27.94	53.23	41.31	50.21	36.83	28.40	26.59	29.30	40.63	41.49	31.88
	TN	6629	6701	6597	6567	6560	6664	6617	6611	6793	6628	6583	6546	6680
	Hit	0.213	0.298	0.254	0.014	0.000	0.024	0.185	0.141	0.222	0.142	0.053	0.021	0.149
	sim	0.222	0.298	0.193	0.007	0.000	0.015	0.117	0.160	0.212	0.165	0.035	0.011	0.126
	Error	0.224	0.313	0.234	0.022	0.000	0.141	0.244	0.220	0.257	0.226	0.083	0.008	0.207
	sim	0.148	0.241	0.133	0.016	0.000	0.086	0.181	0.172	0.166	0.133	0.054	0.004	0.141
	Under	0.564	0.389	0.513	0.964	1.000	0.835	0.571	0.640	0.522	0.632	0.865	0.972	0.644
	sim	0.630	0.461	0.674	0.977	1.000	0.900	0.702	0.668	0.622	0.701	0.911	0.985	0.733
	Over	0.335	0.353	0.348	0.371	0.000	0.482	0.364	0.334	0.417	0.362	0.387	0.12	0.401
	sim	0.348	0.424	0.397	0.454	0.000	0.557	0.462	0.289	0.393	0.313	0.506	0.11	0.483
	Score	0.472	0.486	0.490	0.432	0.450	0.389	0.448	0.434	0.461	0.431	0.431	0.453	0.439
	sim	0.503	0.508	0.490	0.426	0.450	0.400	0.428	0.463	0.489	0.479	0.425	0.449	0.446
SentCLEME2.0-independent	**TP-sim**	9.16	12.59	7.73	0.40	0.00	0.75	5.93	6.67	8.77	6.67	1.50	0.47	5.21
	FP-sim	15.83	29.93	15.62	1.76	0.00	12.58	24.30	14.17	18.94	12.00	6.84	0.27	17.76
	FP$_{ne}$**-sim**	7.20	12.38	6.58	0.70	0.00	5.27	10.94	8.38	8.37	6.25	2.70	0.19	6.81
	FP$_{un}$**-sim**	8.63	17.54	9.03	1.07	0.00	7.31	13.36	5.80	10.57	5.75	4.14	0.08	10.95
	FN-sim	31.54	22.55	32.06	47.73	48.90	43.66	33.43	33.87	30.37	33.61	45.12	48.29	36.24
	Hit	0.155	0.239	0.189	0.010	0.000	0.016	0.137	0.100	0.165	0.106	0.036	0.015	0.105
	sim	0.154	0.240	0.174	0.009	0.000	0.014	0.125	0.100	0.155	0.103	0.033	0.012	0.102
	Error	0.159	0.261	0.178	0.015	0.000	0.110	0.192	0.165	0.192	0.162	0.059	0.005	0.147
	sim	0.134	0.229	0.147	0.013	0.000	0.094	0.170	0.144	0.164	0.129	0.051	0.004	0.127
	Under	0.403	0.268	0.373	0.627	0.647	0.563	0.390	0.447	0.375	0.450	0.574	0.635	0.449
	sim	0.429	0.299	0.415	0.629	0.647	0.580	0.425	0.467	0.407	0.475	0.586	0.639	0.471
	Over	0.183	0.315	0.227	0.023	0.000	0.171	0.224	0.163	0.266	0.165	0.086	0.004	0.206
	sim	0.183	0.320	0.230	0.023	0.000	0.169	0.229	0.159	0.264	0.150	0.089	0.005	0.211
	Score	0.485	0.486	0.493	0.466	0.468	0.428	0.461	0.453	0.474	0.458	0.461	0.474	0.461
	sim	0.493	0.498	0.496	0.466	0.468	0.432	0.462	0.461	0.478	0.469	0.462	0.473	0.466

Table 8: Detailed evaluation results across 13 GEC systems on CoNLL-2014 on GJG15.