CLEME2.0: Towards 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 received little attention in previous studies. To bridge the gap, we introduce **CLEME2.0**, a reference-based metric describing four fundamental aspects of GEC systems: hit-correction, wrong-correction, under-correction, and over-correction. They collectively contribute to exposing critical qualities and locating drawbacks of GEC systems. Evaluating systems by combining these aspects also leads to superior human consistency over other reference-based and reference-less metrics. Extensive experiments on two human judgment datasets and six reference datasets demonstrate the effectiveness and robustness of our method, achieving a new state-of-the-art result.[1]

1 Introduction

The task of *Grammatical Error Correction* (GEC) automatically detects and corrects grammatical errors in a given text (Bryant et al., 2023). A core component of GEC is the development of automatic metrics that can objectively measure model performance (Kobayashi et al., 2024b; Ye et al., 2023c). 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.

Recent research efforts have focused on developing GEC metrics that closely align with human judgements (Koyama et al., 2024), whereas the interpretability of these metrics has received less emphasis. We define the *interpretability* of metrics as their *capacity to disclose concerned characteristics of systems*, which is crucial for identifying weaknesses in a given GEC system. It is generally recognized that excellent systems should adhere to the principles of grammaticality and faithfulness (Bryant et al., 2023). Grammaticality demands that all grammatical errors be accurately corrected, while faithfulness ensures that corrections retain the original meaning and syntactic structure. Nevertheless, the commonly utilized GEC metrics (Bryant et al., 2017; Dahlmeier and Ng, 2012a) are PRF-based (**P**recision, **R**ecall, and **F** scores). We claim that PRF-based metrics fail to effectively capture subtle dimensions of GEC systems, consequently hindering progress. As illustrated in Figure 1, the edits [*were*→*was*] and [*for*→*in*] in Hyp. 1 are regarded as 2 FP + FN edits by ERRANT (Bryant et al., 2017) or 2 FP edits by CLEME (Ye et al., 2023c). Meanwhile, the edit [ϵ→*of*] in Hyp. 2 is categorized as an FP edit for both ERRANT and CLEME. Despite this, these two categories of FP edits carry different implications. The former type is correctly placed but wrongly modified, whereas the latter is incorrectly positioned. The inability to differentiate between these FP edits results in ambiguous interpretations of P/R/$F_{0.5}$ scores, as they fail to quantify grammaticality and faithfulness.

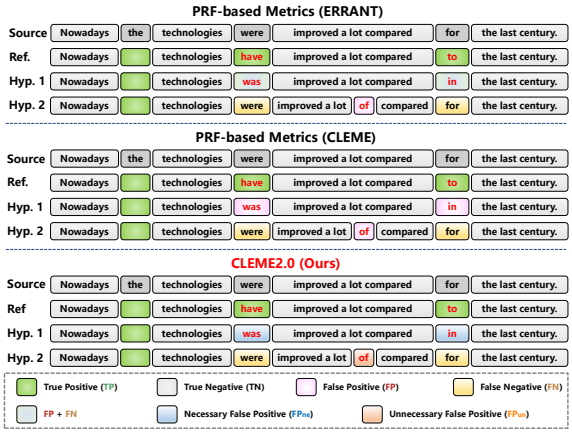

Figure 1: An example of CLEME2.0.

Thus, we introduce **CLEME2.0**, an interpretable reference-based metric describing four fundamen-

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

tal aspects of GEC systems: 1) the *hit-correction* score reflects the degree to which a system accurately corrects grammatical errors, 2) the *wrong-correction* score denotes the degree of incorrect corrections made, 3) the *under-correction* score reveals the degree of missing corrections, and 4) the *over-correction* score measures the degree of excessive corrections. An excellent GEC system should gain a higher hit-correction score and lower wrong-correction, under-correction, and over-correction scores. The initial three aspects assess grammaticality, whereas the over-correction score pertains to faithfulness, given that it often alters the original meaning, a challenge notably observed with LLMs (Coyne et al., 2023; Li et al., 2023a). To achieve this, CLEME2.0 first distinguishes between necessary and unnecessary *false positive* (FP) edits. The idea is that necessary FP edits indicate the system's wrong-correction degree, while unnecessary FP edits reveal the system's over-correction degree. As shown in the bottom block of Figure 1, [*were*→*was*] and [*for*→*in*] in Hyp. 1 are regarded as FP_{ne} edits, while [ϵ→*of*] in Hyp. 2 is considered as an FP_{un} edit.

As a result, CLEME2.0 establishes a one-to-one relationship between four distinct system aspects and four types of edits: hit-correction v.s. TP, wrong-correction v.s. FP_{ne}, under-correction v.s. FN, and over-correction v.s. FP_{un}. Unlike conventional GEC metrics like ERRANT (Bryant et al., 2017) and MaxMatch (Dahlmeier and Ng, 2012a) that evaluate using P/R/$\text{F}_{0.5}$ scores, it offers a nuanced view into the detailed aspects necessary for characterizing critical features of GEC systems. These separated scores are then consolidated into an overall score via linear weighted summation, giving varying importance to these distinct scores. This aggregate score provides a holistic measure of system performance. Similar to CLEME, our method adopts the chunk partition technique and supports evaluations based on either correction dependence or correction independence assumptions, so we dub the metric as CLEME2.0.

Moreover, we propose that edits of varying modification levels should uniquely influence the evaluation outcomes. For example, corrections involving punctuation are often less significant than corrections of content words. Therefore, we integrate two edit weighting techniques into CLEME2.0, similarity-based weighting (Gong et al., 2022) and LLM-based weighting. In particular, these methods compute a specific weight for each edit through a language model rather than assigning equal weight to all edits, thereby enabling CLEME2.0 to grasp contextual semantics and address the limitations of conventional metrics that depend on surface-level form similarity (Kobayashi et al., 2024a).

To verify the effectiveness of CLEME2.0, we conduct extensive experiments on two 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 six reference datasets with disparate annotation styles. In summary, our contributions are three folds:

(1) We introduce CLEME2.0, an interpretable reference-based metric, which is beneficial to reveal crucial aspects of GEC systems.

(2) We enhance CLEME2.0 by incorporating two edit weighting techniques, addressing the limitations of conventional reference-based metrics in capturing semantics.

(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 comparing their outputs to manually written references (Ye et al., 2023a,b; Huang et al., 2023; Li et al., 2024; Napoles et al., 2015). The M^2 scorer (Dahlmeier and Ng, 2012b) identifies optimal edit sequences between source sentences and system hypotheses, using $\text{F}_{0.5}$ scores. However, this method can inflate scores by manipulating edit boundaries. To mitigate this problem, ERRANT (Bryant et al., 2017) improves edit extraction through a linguistically-informed alignment algorithm, but it remains language-dependent and biased in multi-reference evaluation. CLEME (Ye et al., 2023c) further provides unbiased $\text{F}_{0.5}$ scores and introduces an extra correction assumption for multi-reference evaluation. PT-M^2 (Gong et al., 2022) combines PT-based and existing GEC metrics for higher correlations with human judgments.

Reference-less metrics. To overcome the limitations of reference-based metrics, recent research focus on reference-less scoring. Inspired by quality estimation in NMT (Liu et al., 2022; Dong

et al., 2023), 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. (2020) 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., 2023c) to execute the process of edit extraction. Unlike the traditional metrics like ERRANT (Bryant et al., 2017) and Max-Match (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., 2023c).

3.2 Disentangled Scores

To compute disentangled scores, we initially disentangle edits into four elementary types. 1) **TP edits** refer to the corrected/dummy hypothesis chunks that share the same tokens as the corresponding reference chunks. 2) **FP_{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_{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; Li et al., 2023b) do not distinguish between FP_{ne} and FP_{un}, treating both as FP, thereby resulting in confusion between wrong-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_{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_{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:

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

Wrong-correction score. Conversely, the wrong-correction score is defined as the ratio of FP_{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:

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

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 follows:

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

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_{un} edits to all hypothesis corrected/dummy edits, aiming to gauge the level to which systems offer excessive corrections:

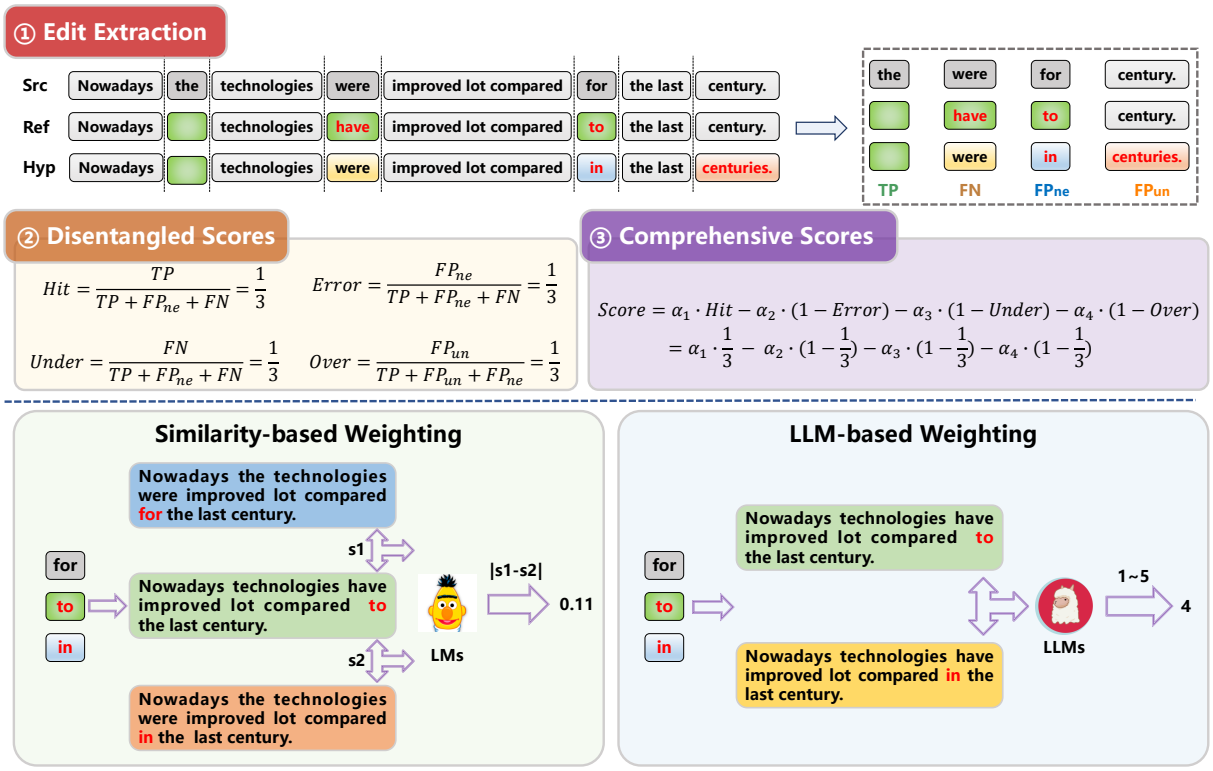

Figure 2: Overview of CLEME2.0. Initially, we extract edits and categorize hypothesis edits as TP, FN, FP_{ne}, and FP_{un}. Next, we compute four distinct scores. Finally, we integrate these scores into an overall score utilizing one of the edit weighting techniques.

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

With the disentangled scores indicating disparate aspects of GEC systems, researchers can identify specific weaknesses and implement targeted improvements without expensive human labor.

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

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

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) ∼ (4). Take the hit-correction score as a typical example, we reformulate the Equation (1) as follows:

$$Hit = \frac{w_{TP}}{w_{TP} + w_{FP_{\text{ne}}} + w_{FN}} \quad (6)$$

Similarity-based weighting. We use PTScore to assign edit weights (Gong et al., 2022). Since it performs based on BERTScore (Zhang et al., 2019), a tool for evaluating text generation through similarity scores, we refer to this technique as similarity-based weighting. The rationale is to prioritize edits with a more significant modification of the meaning and quality of the text.

By simulating a partially accurate version X' of the source sentence X, PTScore can associate specific weights to edits within a sentence. 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 R is the reference sentence, while the function $\text{replace}()$ is used to replace a specific chunk of the source X with the corrected/dummy hypothesis chunk e_{hyp}. A positive weight $w > 0$ indicates a beneficial correction, whereas a negative value suggests a wrong correction. The absolute value $|w|$ is utilized as the edit weight following (Gong et al., 2022), and the significance[2] of an edit in a sentence grows with a larger $|w|$.

LLM-based weighting. Recent studies have begun investigating the effectiveness of LLM-based evaluation, known for their advanced semantic comprehension (Qin et al., 2024b), in assessing various NLP tasks (Pavlovic and Poesio, 2024; Sottana et al., 2023). Building on this trend, we prompt Llama-2-7B (Touvron et al., 2023) to assign edit weights from 1 to 5, where a higher value signifies more critical edits. This methodology is rooted in the idea that LLMs, due to their extensive training on diverse data, are adept at grasping intricate language patterns and text structure. Detailed implementation instructions and the prompting framework are available in Appendix A.

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., 2023c; 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 B.

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 utilize a cross-evaluation search method to identify two optimal sets of trade-off factors for both the corpus and sentence levels. At the corpus level, the factors are assigned as $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.45, 0.35, 0.15, 0.05$, while for the sentence level, they are adjusted to $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.35, 0.25, 0.20, 0.20$.

Evaluation assumptions. CLEME can evaluate GEC systems based on correction *dependence* (-dep) or *independence* (-ind) assumptions. The correction independence assumption offers a more relaxed edit-matching process, implying that systems might yield better scores when multiple references

[2]For more detailed analysis, refer to our case study in Section 5.1 and PT-M^2 (Gong et al., 2022).

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. CLEME2.0-sim is based on similarity-based weighting. We highlight the **highest** scores in bold and the second-highest scores with underlines. ♣ We exclude unchanged references for higher correlations due to low-quality annotations in some reference sets. Results without excluding references are presented in Appendix C.1.

are available. Inspired by this work, CLEME2.0 also supports both assumptions, and we will study their effects 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, even though they do not utilize any 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 is consistent with recent studies (Gong et al., 2022; Ye et al., 2023c). This is because system-level rankings treat each sample equally regardless of edit numbers, mirroring how sentence-level metrics are evaluated. On the other hand, corpus-level metrics emphasize samples with more edits, thus causing the

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).

gap between automatic metrics and human evaluation. SentPT-M^2 shows superior performance on CoNLL-2014 but performs worse on BN-10GEC, E-Minimal, and NE-Fluency compared to our approach, revealing a lack of robustness of the metric.

Generally, our method aligns more closely with human assessments than existing popular metrics. Notably, our method with similarity-based weighting surpasses unweighted ones, thanks to the integration of semantic factors. However, on E-Minimal and NE-Minimal, weighted and unweighted results are comparable. We suspect this is because these datasets have minimal yet crucial annotations, reducing the possibility of varying weights and the efficacy of edit weighting.

Furthermore, we present comprehensive results of CLEME2.0 on CoNLL-2014 and offer insights into our method for analyzing and identifying weaknesses in GEC systems in Appendix C.2.

4.3 Results of SEEDA Ranking

We carry out an additional experiment on the SEEDA-Sentence and SEEDA-Edit datasets, where we compare our method against a variety of GEC metrics. As presented in Table 2, our approach consistently achieves the best outcomes across both datasets. According to Kobayashi et al. (2024b), the correlations of most metrics tend to decrease when transitioning from classical to neural evaluation systems. This implies that conventional metrics might face difficulties in evaluating the more extensively edited and fluent corrections produced by neural systems. Nevertheless, our method effectively tackles these challenges, delivering even improved performance. The results for SEEDA-Edit exceed those for SEEDA-Sentence, due to the greater detail in SEEDA-Edit, aligning more closely with the operation 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.

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

4.4 Results of LLM-based Weighting

Table 3 presents the outcomes of LLM-based weighting, noting that its effectiveness is not as favorable as similarity-based weighting. A likely reason is the coarse grading method of LLMs, which allocates edit weights from 1 to 5, unlike the finer continuous scale $[0, 1]$. Although Kobayashi et al. (2024a) argue that LLMs serve as effective evaluators for GEC, their research pertains to huge closed-source LLMs (GPT-4 and GPT-3.5) and involves specific prompt engineering. They also identify the importance of the LLM scale since GPT-3.5 may even obtain negative correlations with human judgments. In contrast, we employ a more straightforward approach with open-source LLama-2-7B.

	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 ?
	$Hit = 0.00$, $Wrong = 0.79$, $Under = 0.21$, $Over = 0.00$					

	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	, should we disclose	this result	to	our	relatives ?
Ref.	When we are	diagnosed	with certain genetic	diseases	, should we disclose	this result	to	our	relatives ?
Hyp.	When we are	**diagnosed out** (0.056)	with certain genetic	**diseases** (0.006)	, should we disclose	**the results** (0.019)	to	**their** (0.021)	relatives ?
	$Hit = 0.10$, $Wrong = 0.90$, $Under = 0.0$, $Over = 0.39$								

Table 4: Study cases of CLEME2.0 with similarity-based weighting. We highlight TP, FP_{ne}, FP_{un}, and FN chunks in different colors. Values in brackets are similarity-based weighting scores.

5 Analysis

5.1 Case Study

Table 4 demonstrates instances of CLEME2.0. In the first set, Chunks 3 and 5 are FP_{ne} edits contributing to the wrong-correction score, with a higher edit weight of Chunk 5 than Chunk 3 since Chunk 5 introduces an error that entirely alters the sentence's meaning. In the second set, Chunk 2 obtains the highest edit weight of 0.056, underscoring its substantial influence on the evaluation. Despite the correct modification of "diagnosed", the misuse of "out" remains, keeping the correction wrong. Chunk 4 illustrates a singular-to-plural correction in the source sentence, with a low weight indicating a minor impact. Chunks 6 and 8 showcase over-corrections. Chunk 6 leaves the original meaning unchanged, whereas Chunk 8 introduces a significant error by misusing a personal pronoun.

The cases highlight the effectiveness of the weighting technique. Otherwise, all edits are given equal weight, failing to distinguish hypothesis edits with varying correction levels. We display the cases of LLM-based weighting in Appendix C.3.

5.2 Ablation Study

We conduct ablation studies on (Sent)CLEME2.0-dep to analyze the performance of individual disentangled scores. A preferable system has reduced wrong-correction, under-correction, and over-correction scores, so we report corrections between 1 - x with human judgments where x is one of the scores. The outcomes are detailed in Table 5. Hit-correction and under-correction show moderate correlations. Over-correction scores have small positive correlations at the corpus level, with minimal negative correlations at the sentence level. Notably, wrong-correction scores display negative correlations, but this does not mean they do not impact the overall score. In reality, the trade-off factor for wrong-correction scores is relatively substantial. The hypothesis is that focusing evaluations only on wrong-correction scores might prefer systems that make only highly confident edits, potentially leading to assessment bias.

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-Wrong	-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-Wrong	-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 5: Correlation results of each disentangled score on GJG15 Ranking.

Additionally, we utilize the similarity-weighting approach on CLEME to evaluate its efficacy, with the outcomes detailed in Appendix C.4. To examine our method on a broad scale, we also provide the average correlations obtained from a comprehensive analysis of all potential parameter settings. The results are found in Appendix C.5.

6 Conclusion

This paper introduces CLEME2.0, an interpretable evaluation metric for GEC that effectively highlights four key aspects of systems. By incorporating edit weighting techniques, we overcome the challenges traditional reference-based metrics face in recognizing semantic subtleties. Extensive experiments and analyses confirm the effectiveness and robustness of our method. We anticipate that CLEME2.0 will offer a valuable perspective in the GEC community.

Limitation

Limitation in languages and datasets. While CLEME2.0 is adaptable to various languages, its efficiency beyond English remains unverified. Additionally, the reference sets employed in our experiments stem from the CoNLL-2014 shared task, which involves a second language dataset. To confirm the robustness of our methods, it's necessary to conduct further experiments using evaluation datasets that cover a range of languages and text domains. Finally, we highly encourage the creation of new GEC evaluation datasets to foster progress.

Lack of further human evaluation for interpretability. The experiments discussed in the paper are primarily concerned with assessing the correlation between automatic metrics and human judgments. However, they fall short of providing a thorough analysis of the method's interpretability. Although we showcase the strong correlation performance of CLEME2.0, its interpretability is still unverified. In future work, we will conduct human evaluation experiments to showcase the interpretability of our method.

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.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2023. A survey of natural language generation. *ACM Comput. Surv.*, 55(8):173:1–173:38.

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.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024. Are llm-based evaluators confusing nlg quality criteria? *arXiv preprint arXiv:2402.12055*.

Haojing Huang, Jingheng Ye, Qingyu Zhou, Yinghui Li, Yangning Li, Feng Zhou, and Hai-Tao Zheng. 2023. A frustratingly easy plug-and-play detection-and-reasoning module for chinese spelling check. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 11514–11525. Association for Computational Linguistics.

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

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*.

Shota Koyama, Ryo Nagata, Hiroya Takamura, and Naoaki Okazaki. 2024. n-gram F-score for evaluating grammatical error correction. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 303–313, Tokyo, Japan. Association for Computational Linguistics.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023a. On the (in)effectiveness of large language models for chinese text correction. *CoRR*, abs/2307.09007.

Yinghui Li, Shang Qin, Jingheng Ye, Shirong Ma, Yangning Li, Libo Qin, Xuming Hu, Wenhao Jiang, Hai-Tao Zheng, and Philip S. Yu. 2024. Rethinking the roles of large language models in chinese grammatical error correction. *CoRR*, abs/2402.11420.

Yinghui Li, Zishan Xu, Shaoshen Chen, Haojing Huang, Yangning Li, Yong Jiang, Zhongli Li, Qingyu Zhou, Hai-Tao Zheng, and Ying Shen. 2023b. Towards real-world writing assistance: A chinese character checking benchmark with faked and misspelled characters. *CoRR*, abs/2311.11268.

Ruiyang Liu, Yinghui Li, Linmi Tao, Dun Liang, and Hai-Tao Zheng. 2022. Are we ready for a new paradigm shift? A survey on visual deep MLP. *Patterns*, 3(7):100520.

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*.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024a. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. 2024b. Multilingual large language model: A survey of resources, taxonomy and frontiers. *CoRR*, abs/2404.04925.

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*.

Zhen Tan, Alimohammad Beigi, Song Wang, Ruocheng Guo, Amrita Bhattacharjee, Bohan Jiang, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*.

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*.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, and Sujian Li. 2024. InstructEval: Instruction-tuned text evaluator from human preference. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13462–13474, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023a. Mixedit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10161–10175. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, and Haitao Zheng. 2023b. System report for ccl23-eval task 7: Thu kelab (sz)-exploring data augmentation and denoising for chinese grammatical error correction. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pages 262–270.

Jingheng Ye, Yinghui Li, Qingyu Zhou, Yangning Li, Shirong Ma, Hai-Tao Zheng, and Ying Shen. 2023c. 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. 2020. Some: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522.

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 LLM-based Edit Weighting

Because of the powerful semantic comprehension abilities of large language models (LLMs) (Qin et al., 2024a; Tan et al., 2024), recent studies (Hu et al., 2024; Chen et al., 2024; Wu et al., 2024) have generated interest in employing LLMs for text assessment on various NLP tasks. Building on this idea, we use Llama-2-7B as a scorer to determine edit weights. The prompt for edit weighting is presented in Figure 3. We set the temperature to 0.1 in order to ensure consistent and certain results. We instruct the LLM to evaluate each edit individually to prevent interference from other grammatical errors. Edit weights vary from 1 to 5, with higher values representing a greater need for correction. We do not specify the types of edits to the LLM; instead, we allow the LLM to directly evaluate the importance of edits through inherent language understanding abilities. An input to our LLM-based weighting technique is composed of an uncorrected sentence and a certain edit.

Prompt:
As an evaluator for grammatical error correction, you are tasked with assessing the importance of each error. You will be provided with two lines: the first is an uncorrected sentence, the second shows the edit. Then you output the importance score of the given edit.
The scores range from 1 to 5, where a higher score reflects the greater significance of the correction, while a lower score indicates minor importance.
- A score of 1 means the correction is almost negligible and unnecessary.
- A score of 2 means the correction has slight influence.
- A score of 3 signifies some impact by the correction.
- A score of 4 means the edit is essential.
- A score of 5 indicates the modification is highly important and necessary.

Next, I'll provide you a sentence with an edit. You should score each edit accordingly. The output should only be the score, with no additional explanation.

Example Input:
Uncorrected sentence: Nowadays the technologies were improved a lot compared to the last century.
Edit: were → have
Example Output (1-5): 5

Note that the output must be a number between 1 and 5. Here is the formal input:
Uncorrected sentence: {uncorrected sentence}
Edit: {edit}
Example Output (1-5):

Figure 3: Prompt of LLM-based weighting.

B Details about GEC Meta-Evaluation

B.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.2 Statistics of Reference Datasets

Table 6 presents the statistics of all the reference sets involved in our experiments.

B.3 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

Item	CoNLL-2014	BN-10GEC	E-Minimal	E-Fluency	NE-Minimal	NE-Fluency
# Sentence (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)
# Reference (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)
# Edit (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 Chunk (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 Chunk (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 6: 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., 2023c) for chunk extraction.

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., 2023c) 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 following 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 calculated by GPT-2 (Radford et al., 2019), the labeled ranking ratio and the Levenstein distance ratio.

- **SOME** (Yoshimura et al., 2020): 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 Extra Results

C.1 Results of Full References

The results without excluding unchanged reference sentences are presented in Table 7. We observe an obvious performance reduction in traditional metrics especially in NE-Minimal, which contains numerous under-corrections due to annotation by non-experts under the minimal editing guideline. We remove 470 unchanged references in E-Minimal and 612 unchanged references in NE-Minimal. In particular, SentERRANT, SentCLEME-dep, and SentCLEME-ind exhibit negative correlations in NE-Minimal revealing their lack of robustness. Many metrics also undergo a significant decrease in E-Minimal except CLEME2.0. In the case of E-Minimal, many metrics also show a marked decline, except for CLEME2.0. Our approach achieves the highest or comparable correlations across all reference sets, underscoring the method's robustness.

C.2 Details Results on CoNLL-2014

Table 11 presents a comprehensive evaluation of CLEME2.0 on CoNLL-2014 across all GEC systems. Our method offers a clear and quantitative examination of detailed features of GEC systems, which other automatic metrics cannot provide. For instance, the CAMB system attains the

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	
SentGLEU	γ	0.784	0.828	0.756	0.826	0.624	0.581	0.785	0.846	0.218	0.142	0.778	0.848	0.668 (⇓ 0.120)
	ρ	0.720	0.775	0.769	0.824	0.599	0.593	0.791	0.846	0.220	0.170	0.768	0.846	0.660 (⇓ 0.131)
SentERRANT	γ	0.870	0.846	0.885	0.896	0.760	0.692	0.806	0.732	0.104	-0.066	0.793	0.847	0.680 (⇓ 0.130)
	ρ	0.742	0.747	0.786	0.830	0.626	0.588	0.813	0.764	-0.003	-0.137	0.830	0.857	0.620 (⇓ 0.179)
SentCLEME-dep	γ	0.876	0.844	**0.915**	**0.913**	0.602	0.507	0.849	0.886	-0.021	-0.127	0.876	0.921	0.670 (⇓ 0.185)
	ρ	0.824	0.808	**0.835**	**0.874**	0.451	0.412	0.824	0.863	-0.181	-0.247	0.791	0.846	0.592 (⇓ 0.233)
SentCLEME-ind	γ	0.868	0.857	0.539	0.453	0.513	0.410	0.841	0.877	-0.061	-0.181	0.852	0.896	0.572 (⇓ 0.279)
	ρ	0.725	0.758	0.209	0.143	0.368	0.335	0.808	0.846	-0.167	-0.247	0.762	0.825	0.447 (⇓ 0.335)
SentCLEME2.0-dep (Ours)	γ	0.870	0.881	0.766	0.830	0.937	0.928	0.892	0.938	0.634	0.571	0.916	0.949	0.843 (⇓ 0.054)
	ρ	0.714	0.725	0.681	0.747	**0.846**	**0.852**	0.824	0.901	0.368	0.352	0.720	0.791	0.710 (⇓ 0.091)
SentCLEME2.0-ind (Ours)	γ	0.866	0.881	0.799	0.853	**0.940**	**0.933**	0.915	0.952	0.693	0.631	0.883	0.904	0.854 (⇓ 0.045)
	ρ	0.709	0.720	0.681	0.747	0.819	0.835	0.857	0.923	0.423	0.401	0.654	0.720	0.707 (⇓ 0.086)
SentCLEME2.0-sim-dep (Ours)	γ	0.926	0.937	0.797	0.861	0.914	0.902	0.908	0.952	0.607	0.550	0.918	0.947	0.852 (⇓ 0.054)
	ρ	**0.907**	**0.912**	0.808	0.863	0.808	0.813	**0.885**	0.945	0.527	0.505	**0.896**	**0.940**	**0.817 (⇓ 0.051)**
SentCLEME2.0-sim-ind (Ours)	γ	0.915	0.936	0.808	0.866	0.922	0.916	**0.923**	**0.963**	**0.720**	**0.669**	**0.931**	**0.961**	**0.877 (⇓ 0.038)**
	ρ	0.868	0.879	0.753	0.824	0.808	0.841	0.879	**0.956**	**0.544**	**0.527**	0.835	0.923	0.803 (⇓ 0.052)

Table 7: Correlation results on GJG15 Ranking. We report the results without excluding unchanged reference sentences and the reduction compared with Table 1. We highlight the **highest** scores in bold and the second-highest scores with underlines.

top hit-correction score of 0.271 for CLEME2.0-dep, which shows that about 27.1% of edits by the system are accurate. The wrong-correction score of 0.194 indicates that 19.4% of edits are correctly placed but incorrect, the under-correction score of 0.534 indicates that 53.4% of grammatical errors are overlooked by the system, and the over-correction score of 0.470 suggests that 47.0% of the edits are unnecessary.

As a result, developers and researchers can pinpoint the aspects of their systems that require enhancement. Furthermore, users can select GEC systems that best meet their requirements. For instance, users might opt for a system with a minimal under-correction score in high-stakes situations, as they expect to detect every possible grammatical mistake even though the system might make some unnecessary edits.

C.3 Case Study of LLM-based Weighting

In Table 9, we report instances of CLEME2.0 using LLM-based weighting. We notice distinct preferences when comparing similarity-based and LLM-based weighting methods. In the first example, Llama-2 attributes significant weights to Chunks 1 and 2, highlighting key grammatical mistakes. Conversely, it assigns a minor weight to Chunk 5 due to its imperfect modification. The second example shows Llama-2 attributing substantial weights to all chunks. Specifically, for Chunk 2, the hypothesis fails to remove the redundant "out," emphasizing the under-correction issue. Chunks 6 and 8 display excessive corrections, altering the sentence's original intent and thus indicating considerable over-correction. Generally, Llama-2 tends to ascribe either very high or very low weights to modifications. We speculate it is due to the small scale of the LLM we adopt, impairing its ability to distinguish grammatical errors with varying levels.

Metric	EW		TS	
	γ	ρ	γ	ρ
CLEME-dep-unw	0.638	0.654	0.681	0.709
CLEME-ind-unw	0.640	0.648	0.680	0.698
CLEME-dep-len	**0.700**	0.665	0.691	0.742
CLEME-ind-len	0.649	0.709	0.691	0.731
CLEME-dep-sim	0.655	**0.764**	**0.698**	**0.797**
CLEME-ind-sim	0.641	0.720	0.687	0.747
SentCLEME-dep-unw	0.853	0.687	0.805	0.604
SentCLEME-ind-unw	0.790	0.275	0.722	0.181
SentCLEME-dep-len	0.876	**0.824**	0.844	**0.808**
SentCLEME-ind-len	0.868	0.725	**0.857**	0.758
SentCLEME-dep-sim	**0.888**	0.692	0.844	0.648
SentCLEME-ind-sim	0.843	0.500	0.786	0.434

Table 8: Extra results of CLEME with different edit weighting techniques: unweighting (unw), length-based weighting (len), and similarity-based weighting (sim).

C.4 Extra Results of CLEME with Similarity Weighting

We additionally investigate the application of similarity-based weighting to CLEME (Ye et al., 2023c) and present the results on CoNLL2014 in Table 8. We find that similarity-based weighting is superior to length-based weighting for corpus-level CLEME while the trend is reversed for Sent-

	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** (5)	who	**suffer** (5)	from this disease keep it a secret	**to inform** (1)	their relatives ?
$Hit = 0.00, \quad Wrong = 0.55, \quad Under = 0.45, \quad Over = 0.00$						

	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	, should we disclose	this result	to	our	relatives ?
Ref.	When we are	diagnosed	with certain genetic	diseases	, should we disclose	this result	to	our	relatives ?
Hyp.	When we are	**diagnosed out** (5)	with certain genetic	**diseases** (5)	, should we disclose	**the results** (4)	to	**their** (5)	relatives ?
$Hit = 0.50, \quad Wrong = 0.50, \quad Under = 0.00, \quad Over = 0.47$									

Table 9: Study cases of CLEME2.0 with LLM-based weighting. We highlight TP, FP_{ne}, FP_{un}, and FN chunks in different colors. Values in brackets are LLM-based weighting scores.

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

CLEME, and both are better than the unweighting setting. Moreover, it should be noted that no matter the weighting strategy employed, CLEME consistently underperforms compared to CLEME2.0. This is attributed to the fundamental disparities in design and scoring frameworks between the versions. CLEME2.0 was crafted to incorporate these sophisticated weighting techniques, allowing it to better distinguish between diverse error types and deliver a more thorough and refined performance assessment.

C.5 Average Correlations.

To analyze our method from a global viewpoint, we present the average correlations derived from the exhaustive enumeration of possible parameter configurations. We explore all potential parameter combinations with increments of 0.05. Table 10 shows that all correlations are positive, regardless of the correction assumptions, levels of evaluation, or weighting techniques used. By comparing results from unweighted and similarity-based weighted metrics, we determine that similarity-based weighting substantially enhances human correlation on a global level. Additionally, corpus-level metrics generally achieve higher average values compared to sentence-level metrics. However, sentence-level metrics with optimal parameters can outperform their corpus-level equivalents. This implies that corpus-level metrics might demonstrate greater robustness concerning parameter selection.

Metric		AMU	CAMB	CUUI	IITB	INPUT	IPN	NTHU	PKU	POST	RAC	SJTU	UFC	UMC
CLEME2.0-dep	**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
	Wrong	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-dep	**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
	Wrong	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-ind	**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
	Wrong	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-ind	**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
	Wrong	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 11: Detailed evaluation results across 13 GEC systems on CoNLL-2014.