IMPARA-GED: Grammatical Error Detection is Boosting Reference-free Grammatical Error Quality Estimator

Yusuke Sakai[†], **Takumi Goto**[†], **Taro Watanabe**
Nara Institute of Science and Technology (NAIST), Japan
{sakai.yusuke.sr9, goto.takumi.gv7, taro}@is.naist.jp
†Equal Contribution

Abstract

We propose IMPARA-GED, a novel reference-free automatic grammatical error correction (GEC) evaluation method with grammatical error detection (GED) capabilities. We focus on the quality estimator of IMPARA, an existing automatic GEC evaluation method, and construct that of IMPARA-GED using a pre-trained language model with enhanced GED capabilities. Experimental results on SEEDA, a meta-evaluation dataset for automatic GEC evaluation methods, demonstrate that IMPARA-GED achieves the highest correlation with human sentence-level evaluations.

1 Introduction

Grammatical error correction (GEC) is the task of automatically correcting grammatical or superficial errors in input sentences. While GEC systems should manually assess the quality of their corrections, human evaluation for a wide range of arbitrary inputs is strenuous, making it impractical. Therefore, it is necessary to establish automatic evaluation methods for GEC that correlate highly with human evaluation. The automatic GEC evaluation methods can be categorized into reference-based evaluation methods (Bryant et al., 2017; Gotou et al., 2020; Koyama et al., 2024) and reference-free evaluation methods (Yoshimura et al., 2020; Islam and Magnani, 2021; Maeda et al., 2022).

Reference-based evaluation methods measure the closeness between the outputs of GEC systems and human-written references. However, since incorrect sentences can be corrected in multiple ways, accurate evaluation requires multiple reference sentences. Yet, constructing comprehensive human references is impractical, and low-coverage reference sets often deteriorate evaluation reliability (Choshen and Abend, 2018a,b). Therefore, reference-free evaluation methods, which rely only on input sentences and system outputs, have the potential to overcome these limitations.

Most reference-free evaluation methods employ pre-trained language models (PLMs). For instance, Scribendi Score (Islam and Magnani, 2021) uses the perplexity of a PLM to compute evaluation scores for corrected outputs. SOME (Yoshimura et al., 2020) trains PLMs separately on human assessment scores for fluency, grammaticality, and meaning preservation. IMPARA (Maeda et al., 2022) combines a similarity estimator between inputs and system outputs with a quality estimator for system outputs, which rely on PLMs. The quality estimator is trained without requiring human-annotated evaluation results, using only parallel data of erroneous and corrected sentences constructed for GEC systems. However, although the quality estimator is trained on a vanilla PLM, its pre-trained knowledge alone is insufficient to capture grammatical errors accurately.

In this paper, we propose IMPARA-GED, a novel reference-free automatic evaluation method for GEC. Inspired by the insight that additional training on the Grammatical Error Detection (GED) task enhances GEC systems (Yuan et al., 2021; Kaneko et al., 2020), the quality estimator of IMPARA-GED is constructed by first fine-tuning a PLM on the GED task and then applying quality estimator construction method of IMPARA. Moreover, we remove the similarity estimator used in IMPARA, as empirical observations indicate that it fails to effectively capture grammatical errors in the outputs of modern GEC systems. Our experimental results on SEEDA, a meta-evaluation benchmark (Kobayashi et al., 2024b), show that IMPARA-GED achieves the highest correlation with human sentence-level evaluations.

2 Background: IMPARA

IMPARA consists of two models: a **quality estimator** (QE), which assesses the quality of GEC systems outputs, and a **similarity estimator** (SE),

Findings of the Association for Computational Linguistics: ACL 2025, pages 25647–25654
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

which evaluates the semantic preservation between inputs and system outputs. While the similarity estimator utilizes a vanilla PLM, the quality estimator is constructed by fine-tuning an encoder-based PLM such as BERT (Devlin et al., 2019).

Construction of Quality Estimator. The quality estimator is constructed by learning the pairwise quality ranking order (S_+, S_-), where S_+ has higher quality and S_- has lower quality. These pairs can be automatically generated from parallel data of incorrect and correct sentences constructed for GEC systems. Specifically, an edit set for transforming an incorrect sentence into a correct sentence is extracted, and the impact of each edit is quantified based on the degree of semantic change when the edit is removed from the correct sentence. Two different subsets of the edit set are then sampled and applied to the incorrect sentence, generating two different corrected sentences. Since the impact of each edit is already computed, the total impact can be calculated for each subset as a quality. We regard the higher-quality correction as S_+ and the lower-quality correction as S_-, forming the training data $\mathcal{T}$. Finally, the quality estimator R is trained using the training data $\mathcal{T}$ by minimizing the loss function $\mathcal{L}^{\text{QE}}$ as follows:

$$\mathcal{L}^{\text{QE}} = \frac{1}{|\mathcal{T}|} \sum_{(S_+, S_-) \in \mathcal{T}} \sigma(R(S_-) - R(S_+)), \quad (1)$$

where $\sigma(\cdot)$ is a sigmoid function. The quality estimator R linearly transforms the final layer's first token embedding representation of the incorrect sentence into a real-valued score.

Scoring in IMPARA. IMPARA calculates the evaluation score $S(I, O) \in [0, 1]$ for a pair of an input sentence I and its corrected output O from a GEC system using the quality estimator R and the similarity estimator $\text{sim}(\cdot)$ as follows:

$$S(I, O) = \begin{cases} \sigma(R(O)) & \text{sim}(I, O; \text{PLM}) > \theta \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where θ is the threshold of the similarity. This threshold is used to filter out sentences that are irrelevant to the input sentences but receive high quality estimation scores.

3 Proposed Method: IMPARA-GED

We propose IMPARA-GED by removing the similarity estimator from IMPARA and incorporating additional training on the GED task before quality estimator construction.

	System-level				Sentence-level			
	SEEDA-S		SEEDA-E		SEEDA-S		SEEDA-E	
IMPARA-SE	r	ρ	r	ρ	Acc.	τ	Acc.	τ
QE only	.916	.902	.902	.965	.753	.506	.752	.504
†**BERT$_{\textbf{Base}}$**	.916	.902	.902	.965	.753	.506	.752	.504
BERT$_{\text{Large}}$	.889	.867	.909	.916	.731	.463	.737	.474
BERT$_{\text{Base-uncased}}$	.922	.909	.903	.944	.746	.493	.745	.491
BERT$_{\text{Large-uncased}}$	.902	.895	.904	.951	.738	.476	.743	.487
ELECTRA$_{\text{Base}}$	.920	.902	.904	.965	.752	.505	.751	.503
†**ELECTRA$_{\textbf{Large}}$**	.916	.902	.902	.965	.753	.506	.752	.504
DeBERTa-v3$_{\text{Base}}$	.906	.916	.891	.958	.750	.500	.749	.498
DeBERTa-v3$_{\text{Large}}$	.915	.916	.900	.958	.749	.498	.749	.499
†**ModernBERT$_{\textbf{Base}}$**	.916	.902	.902	.965	.753	.506	.752	.504
ModernBERT$_{\text{Large}}$	.917	.903	.903	.965	.753	.505	.752	.503

Table 1: The score differences on the SEEDA arise from variations in the PLMs used as the SE. The QE is fixed using the reproduced IMPARA QE weights. Employing BERT$_{\text{Base}}$ as the SE makes it equivalent to the IMPARA. The threshold θ is 0.9, as in IMPARA. PLMs marked with † produced similarity scores above the threshold for all instances in SEEDA, rendering the SE meaningless.

3.1 Rethinking of the Similarity Estimator

We observed that, in some cases, filtering by the similarity threshold did not work properly. This is due to PLMs struggling to effectively capture grammatical errors. Table 1 shows the meta-evaluation results of IMPARA using SEEDA (Kobayashi et al., 2024b) (detailed in §4), where various PLMs are employed as the similarity estimator. These results indicate that the choice of PLM affects performance, and the similarity estimator either does not work or negatively impacts the results.

This reason is that similarity estimation with a vanilla PLM results in incorrect filtering. For instance, the sentence pair *"I think the family will stay mentally healty as it is , without having emtional stress."* and *"I think the family will stay mentally healthy without having emotional stress."* is assigned a similarity score of 0.787 by BERT$_{\text{Large-uncased}}$. Here, the corrected sentence is simply a revision of the errors in the incorrect sentence, meaning it should not be filtered by the similarity threshold. However, with IMPARA's default threshold of 0.9, it is incorrectly filtered. Moreover, removing the correction of "healthy" increases the similarity score to 0.926, suggesting that BERT$_{\text{Large-uncased}}$ struggles to understand the semantic impact of spelling errors. In contrast, there are cases where incorrect corrections that should be filtered are instead accepted. For instance, the pair *"I like cats."* and *"I dislike cats."* is assigned a high similarity score of 0.980 with $\text{BERT}_{\text{Base}}$. Since negation is not a valid correction in GEC, this correction should be filtered. However, due to the high similarity, it is mistakenly accepted.

These observations suggest that the similarity estimator fails in its intended role. Furthermore, in the outputs of modern GEC systems included in the SEEDA dataset, corrections that significantly deviate from the original erroneous sentence are rarely encountered in practice. This issue of adversarial corrections is not unique to IMPARA but is a general problem observed in other evaluation metrics, such as SOME (Islam and Magnani, 2021). Given this, filtering adversarial corrections should be explored as a separate research direction. Therefore, we focus on only the quality estimation performance of IMPARA and eliminate the similarity threshold as follows:

$$S(I, O) = \sigma(R(O)). \tag{3}$$

3.2 Additional training on the GED task

The sentence pairs used for constructing IMPARA's quality estimator are created based on the impact of each correction. However, as discussed in §3.1, a vanilla PLM may not sufficiently capture errors. To address this, IMPARA-GED introduces additional training on the GED task to build a quality estimator that more accurately captures token-level error information. Then, IMPARA's training method is applied to construct the final quality estimator.

Following Yuan et al. (2021), the GED model classifies errors at the token level, using four variations of error labels: (1) **2-class** setting that binarizes tokens as correct or incorrect, (2) **4-class** setting that categorizes tokens into correct, insertion, deletion, and substitution, (3) **25-class** setting based on POS categories as defined by ERRANT (Bryant et al., 2017), and (4) **55-class** setting that combines these classifications. These token-level labels are automatically assigned based on existing parallel data of incorrect and corrected sentences and the alignment results from ERRANT.

Formally, given an erroneous sentence $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ consisting of N tokens and its corresponding error labels $\boldsymbol{t} = [t_1, t_2, \ldots, t_N]$, the model is trained by minimizing the loss function:

$$\mathcal{L}^{\text{GED}}(\boldsymbol{x}, \boldsymbol{t}) = -\frac{1}{N} \sum_{i=1}^{N} \log p(t_i | x_i, \boldsymbol{x}). \tag{4}$$

Next, IMPARA-GED fine-tunes the GED model for IMPARA's quality estimator following Equation 1 and performs inference according to Equation 3. The impact calculation is also done using the same GED model. Note that, instead of embedding the first token as in Equation 2, mean pooling over all token embeddings is applied to make more effective use of token-level error detection information.

4 Experimental Settings

Construction of IMPARA-GED. We used CoNLL-2013 (Ng et al., 2013) and FCE (Yannakoudakis et al., 2011) for model construction. CoNLL-2013 was split into train, dev, and devtest sets in an 8:1:1 ratio, while FCE was used with its predefined splits. First, we construct the GED model following the settings of Yuan et al. (2021). The PLM is trained for five epochs using a combined train set of FCE and CoNLL-2013, and the checkpoint that achieves the highest score on the FCE dev set is selected as the final GED model. Next, the GED model is used to build the quality estimator following the procedure described in § 2 and the settings of Maeda et al. (2022), using the CoNLL-2013 train set. The GED model is then trained for ten epochs following Equation 1, and the checkpoint achieving the best performance on the CoNLL-2013 dev set is selected. We trained the model using five different random seeds, and the one that performs best on the CoNLL-2013 devtest set is selected as the final model. We report results for all combinations of the following label granularities: 2-class, 4-class, 25-class, and 55-class, and the following PLMs: $\text{BERT}_{\text{Base}}$ (Devlin et al., 2019), DeBERTa-v3_{Large} (He et al., 2023), and $\text{ModernBERT}_{\text{Large}}$ (Warner et al., 2024).

Evaluations. We conduct meta-evaluations using SEEDA (Kobayashi et al., 2024b), using both edit-level annotations (`SEEDA-E`) and sentence-level one (`SEEDA-S`). We follow the TrueSkill-based system ranking and the `Base` system setting, which includes outputs from 12 modern GEC systems. We report Pearson's correlation coefficient r and Spearman's rank correlation coefficient ρ as correlation metrics for system-level evaluation and Accuracy (Acc.) and Kendall's rank correlation coefficient τ for sentence-level evaluation. As baselines, we include reference-based evaluation methods: ERRANT (Felice et al., 2016; Bryant et al., 2017), PT-ERRANT (Gong et al., 2022), GREEN (Koyama et al., 2024), and GLEU (Napoles et al., 2015).

For reference-free evaluation methods, we report Scribendi Score (Islam and Magnani, 2021), SOME (Yoshimura et al., 2020), and the original IMPARA (Maeda et al., 2022). Additionally, we include GPT-4-S (Kobayashi et al., 2024a) and its three derivative systems as large language model-based GEC evaluation methods. Our implementation uses `gec-metrics`[1] (Goto et al., 2025a) with each system's default settings. For GPT-4-S, we cite the reported values from Kobayashi et al. (2024a). Following Goto et al. (2025b), all system evaluations are conducted using TrueSkill (Herbrich et al., 2006), aligning with the aggregation method used in human evaluation. We also conducted significance testing following Yoshimura et al. (2020), using Williams significance tests (Graham and Baldwin, 2014) for system-level evaluation and bootstrap resampling (Graham et al., 2014) for sentence-level evaluation.

5 Experimental Results and Discussions

Main results. Table 2 shows the SEEDA evaluation results for each automatic GEC evaluation method. We observe a general improvement in sentence-level evaluation when additional training on the GED task is applied. Notably, using ModernBERT$_{\text{Large}}$ with the 2-class classification GED task achieves the highest performance among all methods in the sentence-level `SEEDA-S`. Additionally, in sentence-level `SEEDA-E`, the model outperforms all models except GPT4-S. In system-level `SEEDA-E`, the benefits of GED training were not consistently observed. This may be attributed to the fact that system-level correlations with human evaluation are already high (above 0.9), leaving limited room for further improvement. Indeed, global trends reflected in human judgments are already well captured by automatic evaluation metrics. Therefore, further improvement in correlation would require the ability to capture more subtle and fine-grained evaluations at the sentence level. However, for sentence-level evaluation, the IMPARA-GED series consistently outperformed the base model in most cases, and the impact of GED training was more pronounced. Since IMPARA evaluates at the sentence level, this outcome is reasonable, and GED training has proven effective in enhancing sentence-level evaluation.

	System-level				Sentence-level			
	`SEEDA-S`		`SEEDA-E`		`SEEDA-S`		`SEEDA-E`	
Methods	r	ρ	r	ρ	Acc.	τ	Acc.	τ
ERRANT	.763	.706	.881	.895	.594	.189	.608	.217
PT-ERRANT	.870	.797	.924	.951	.582	.165	.592	.184
GREEN	.855	.846	.912	.965	.600	.199	.574	.148
GLEU	.863	.846	.909	.965	.672	.343	.673	.347
Scribendi	.674	.762	.837	.888	.660	.320	.672	.345
SOME	.932	.881	.893	.944	.778	.555	.766	.532
IMPARA	.939	.923	.901	.944	.753	.506	.752	.504
GPT-4-S	.887	.860	.960	.958	.784	.567	.798	.595
+Grammaticality	.888	.867	.961	.937	.796	.592	.807	.615
+Fluency	.913	.874	**.974**	**.979**	.819	.637	**.831**	**.662**
+ Meaning Preservation	.958	.881	.911	.960	.810	.620	.813	.626
BERT$_{\text{Base}}$	.915	.895	.875	.930	.756	.512	.754	.508
+2-class	.916	.909	.850	.902	.773	.545	.763	.527
+4-class	.908	.902	.859	.923	.787	.574	.774	.548
+25-class	.925	.902	.875	.923	.771	.543	.752	.503
+55-class	.900	.902	.842	.923	.763	.526	.750	.499
DeBERTa-v3$_{\text{Large}}$	.960	.937	.912	.944	.784	.568	.779	.558
+2-class	.951	.923	.895	.916	.797	.593	.784	.568
+4-class	.939	.895	.899	.916	.793	.585	.772	.544
+25-class	.945	.930	.906	.930	.801	.602	.786	.573
+55-class	.955	.930	.913	.958	.782	.564	.763	.527
ModernBERT$_{\text{Large}}$	.949	.909	.912	.937	.767	.533	.749	.497
+2-class	.971	.930	.919	.930	**.829**	**.658**	.797	.594
+4-class	.964	.916	.926	.923	.812	.624	.794	.588
+25-class	**.972**	.937	.933	.944	.801	.603	.783	.567
+55-class	.965	**.951**	.910	.909	.749	.498	.741	.483

Table 2: Meta-evaluation results on SEEDA. Each IMPARA-GED variant is identified by the name of the PLM used for training. The first value in each row shows the result without additional GED training, and the following rows show the results after additional training on the GED task using each class label. **Bold** scores indicate the best performance, underline indicates the second-best, red cells indicate improvements owing to GED task training, and purple values marks statistically significant improvements ($p < 0.05$) over the version without GED training.

Number of GED types. Increasing the number of error classification types for GED training does not necessarily lead to better performance, as even binary classification was sufficient to improve evaluation results. We suspect that label reliability may be influencing the results. In the 55-class setting, although the labels carry more information, their reliability tends to decrease. In contrast, binary classification conveys less information per label but offers higher overall reliability. In the context of GEC evaluation, label reliability may be more critical than label informativeness. This observation aligns with the findings of Yuan et al. (2021), and our results with IMPARA-GED support this view: the binary setting performed best, followed by the 4-class setting. Since the goal of this task is GEC evaluation, using more reliable labels appears

[1] https://github.com/gotutiyan/gec-metrics

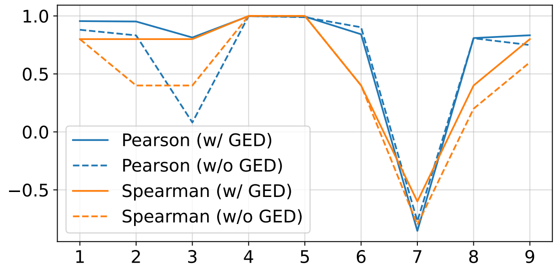

Figure 1: Results of the window analysis with a window size of 4. The x-axis represents the starting rank in the human rankings; for example, $x = 2$ corresponds to the results for systems ranked 2nd to 6th in human evaluation. A comparison is made between the ModernBERT$_{\text{Large}}$ without GED training and with additional training using binary-labeled GED.

better suited to this objective.

System-level Window Analysis. Figure 1 shows the results of a window analysis on system-level SEEDA-S using ModernBERT$_{\text{Large}}$, following the analysis by Kobayashi et al. (2024b). From Figure 1, we observe that additional training on the GED task improves the evaluation performance of top-ranked systems.

Sentence-level Pairwise Comparison. We investigate whether IMPARA-GED can distinguish the output quality of each GEC system in a pairwise comparison at the sentence level. Figure 2 shows the improvement in pairwise discriminative ability on SEEDA-S through the GED task. The results indicate that GED enhances the ability to distinguish between systems with more significant rank differences.

6 Conclusion

We proposed IMPARA-GED, a novel reference-free automatic GEC evaluation method, by enhancing the GED capabilities of PLMs. When using ModernBERT as the PLM, additional training on the binary-labeled GED task achieved the highest correlation with human evaluation among existing methods on SEEDA. Furthermore, window analysis revealed that IMPARA-GED improves evaluation performance, particularly for top-ranked systems. Moreover, we revealed that current similarity estimators fail to adequately capture meaning preservation. We suggest that future development of GEC evaluation metrics may proceed in two directions: either by building a stronger similarity estimator or by enhancing the quality

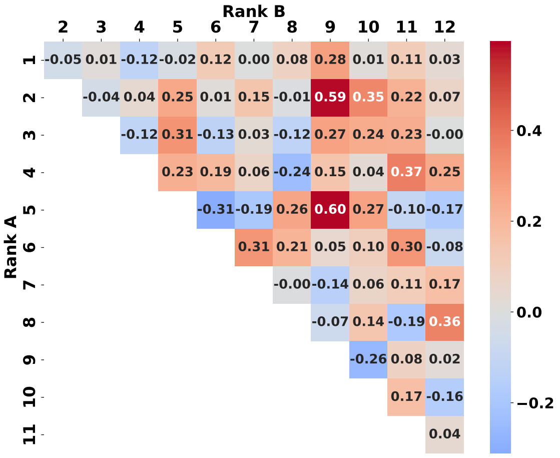

Figure 2: Results of pairwise sentence-level analysis. The hypothesis sentences for each source are ranked according to an evaluation metric, and all pairs are created from the hypotheses while retaining the rank information. Then, the same ranked pairs (A, B) are grouped together at the corpus level, and the percentage of agreement with the human evaluation is calculated for each pairs; $0 \leq A \leq 11$ and $A + 1 \leq B \leq 12$, based on SEEDA's Base setting. The figure shows a heatmap of the differences in Kendall's tau after performing this pairwise analysis for each of the original IMPARA and IMPARA-GED. For example, for Rank A=2 and Rank B=9, the judgment in the hypothesis that the automatic evaluation ranked 2nd and 9th for each source. The value of 0.59 at the cell indicates that the GED greatly improves the evaluation results in this pair. The difference is calculated between ModernBERT$_{\text{Large}}$ with and without additional training using binary-labeled GED.

estimator. We publicly release IMPARA-GED as the official version, based on ModernBERT with binary classification, available at: `https://huggingface.co/naist-nlp/IMPARA-GED`.

7 Limitations

GED. We demonstrated that additional training with the GED task improves IMPARA's performance. However, we did not determine which class type contributes the most to this improvement. Similarly, Yuan et al. and other studies related to GED-boosted GEC system construction have not explored which class type is most effective. Therefore, identifying the optimal number of class types is not the primary focus; rather, the key takeaway is that GED is effective for building automatic GEC evaluation models. That said, our findings suggest that even a two-class setting can yield improvements. Thus, determining the optimal class configuration is an important direction for future work.

Training method. We sequentially apply IMPARA after training on the GED task. This training approach allows for various future extensions, such as multi-task learning or adopting GRECO-style classifier types, e.g., word and gap labels (Qorib and Ng, 2023). The contribution of this paper lies in demonstrating that similarity filtering does not function effectively in reference-free GEC evaluation and that error-type classification, such as GED, has the potential to improve GEC evaluation. Therefore, we did not conduct a comprehensive investigation of alternative training strategies.

Parameter tuning. While we followed the training settings of Maeda et al. (2022) and Yuan et al. (2021), further parameter tuning might lead to even better performance. However, to ensure a fair comparison with properly tuned results, we carefully monitored performance and conducted comparisons under a well-controlled experimental setup. Therefore, while we did not perform extensive parameter tuning, we believe that our evaluations and comparisons are sufficiently tuned to achieve the objectives of our study.

Datasets. In this study, we used CoNLL-2013 and FCE to construct the models, following the setups of Maeda et al. (2022) and Yuan et al. (2021). Therefore, leveraging high-quality, large-scale datasets such as W&I+LOCNESS (Bryant et al., 2019) may lead to even better performance. Additionally, exploring data augmentation methods that enable effective GED training even with small-scale datasets like CoNLL-2013 remains a future challenge. However, in this study, we specifically investigated the impact of additional training with the GED task using the same dataset, ensuring robust and interpretable results. Therefore, while these points are important future directions, they are beyond the scope of this short paper.

Evaluation. In this study, we used SEEDA as the meta-evaluation dataset. SEEDA was introduced to address limitations in previous datasets, such as GJG (Grundkiewicz et al., 2015), which lacked coverage of modern neural network-based GEC models and suffered from a small number of system comparisons. SEEDA includes two benchmarks, SEEDA-S and SEEDA-E, both of which were used in our evaluation. Therefore, the evaluation conducted in this study is comprehensive and addresses current challenges in the field. Furthermore, while SEEDA is currently the only effective dataset available for automatic GEC evaluation, conducting evaluations on more specialized domains would be meaningful. However, due to resource limitations, this remains a broader challenge for the field rather than an issue specific to this study.

PLMs. IMPARA-GED can be applied to any encoder-based model. Therefore, leveraging other PLMs may enable the development of even higher-quality automatic GEC evaluation models. In this study, we aimed to verify the performance impact of additional training with the GED task. To this end, we conducted experiments using BERT$_{\text{Base}}$, which was originally used in IMPARA, as well as DeBERTa$_{\text{Large}}$ and ModernBERT$_{\text{Large}}$, two of the latest improved versions of BERT. Furthermore, we conducted pilot studies with other PLMs and confirmed that additional training with the GED task consistently enhanced performance across different models. However, due to page limitations and resource constraints, we did not include these results in this paper. Thus, while our experiments fulfill the intended verification objective, achieving higher performance would require further evaluation with additional PLMs.

8 Ethical Considerations

In this study, we use open-source tools, PLMs, and datasets that are permitted for research use, ensuring no license issues. Additionally, all datasets used are publicly available and widely recognized in related research, guaranteeing that no harmful data was included in the experiments. For reproducibility, we provide the detailed settings in Appendices A and B. Thus, this study has no ethical considerations.

References

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

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

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

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

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

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

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

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. gec-metrics: A unified library for grammatical error correction evaluation. Preprint, arXiv:2505.19388.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human? Preprint, arXiv:2502.09416.

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

Yvette Graham and Timothy Baldwin. 2014. Testing for significance of increased correlation with human judgment. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 172–176, Doha, Qatar. Association for Computational Linguistics.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2014. Randomized significance tests in machine translation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 266–274, Baltimore, Maryland, USA. Association for Computational Linguistics.

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

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In The Eleventh International Conference on Learning Representations.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill™: A bayesian skill rating system. In Advances in Neural Information Processing Systems, volume 19. MIT Press.

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

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

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024a. Large language models are state-of-the-art evaluator for grammatical error correction. In Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024), pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024b. Revisiting meta-evaluation for grammatical error correction. Transactions of the Association for Computational Linguistics, 12:837–855.

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.

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, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

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

Muhammad Reza Qorib and Hwee Tou Ng. 2023. System combination via quality estimation for grammatical error correction. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 12746–12759, Singapore. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. Preprint, arXiv:2412.13663.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

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

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

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of two systems. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Details of Each PLM

We used the Hugging Face transformers library (Wolf et al., 2020) for all experiments. Table 3 shows the PLMs we used in this study and their corresponding Hugging Face IDs.

PLMs	HuggingFace ID
BERT-Base	google-bert/bert-base-cased
BERT-Large	google-bert/bert-large-cased
BERT-Base-uncased	google-bert/bert-base-uncased
BERT-Large-uncased	google-bert/bert-large-uncased
ELECTRA-Base	google/electra-base-discriminator
ELECTRA-Large	google/electra-large-discriminator
DeBERTa-v3-Base	microsoft/deberta-v3-base
DeBERTa-v3-Large	microsoft/deberta-v3-large
ModernBERT-Base	answerdotai/ModernBERT-base
ModernBERT-Large	answerdotai/ModernBERT-large

Table 3: Lists of the PLMs we used in this study and their corresponding Hugging Face IDs.

B Details of Implementations

For the training setup of IMPARA-GED, we followed the settings of Yuan et al. (2021) for building the GED model and Maeda et al. (2022) for constructing the quality evaluation model. For GED training, we used `ged_baselines`[2]. For IMPARA training, we used its public reproduction implementation[3]. The quality estimator, IMPARA-QE, we used in Table 1, a public reproduction model available on Hugging Face[4]. Unless otherwise specified, we used the default hyperparameters of these tools. We used a single NVIDIA GeForce RTX 3090 GPU for all experiments. We are ready to publish all experimental codes after acceptance to ensure reproducibility. Additionally, IMPARA-GED weights are made publicly available[5].

[2] https://github.com/gotutiyan/ged_baselines

[3] https://github.com/gotutiyan/IMPARA

[4] https://huggingface.co/gotutiyan/IMPARA-QE

[5] https://huggingface.co/naist-nlp/IMPARA-GED