Reliability Crisis of Reference-free Metrics for Grammatical Error Correction

Takumi Goto, Yusuke Sakai, Taro Watanabe
Nara Institute of Science and Technology
{goto.takumi.gv7, sakai.yusuke.sr9, taro}@is.naist.jp

Abstract

Reference-free evaluation metrics for grammatical error correction (GEC) have achieved high correlation with human judgments. However, these metrics are not designed to evaluate adversarial systems that aim to obtain unjustifiably high scores. The existence of such systems undermines the reliability of automatic evaluation, as it can mislead users in selecting appropriate GEC systems. In this study, we propose adversarial attack strategies for four reference-free metrics: SOME, Scribendi, IMPARA, and LLM-based metrics, and demonstrate that our adversarial systems outperform the current state-of-the-art. These findings highlight the need for more robust evaluation methods. Our code is available at: https://github.com/gotutiyan/attack-gec-metrics.

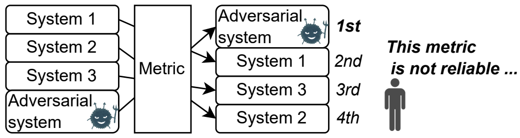

Figure 1: The situation we are concerned about. The adversarial attacking system may obtain an unreasonably high score by hacking a metric. This breaks the reliability of the automatic GEC evaluation.

1 Introduction

Grammatical Error Correction (GEC) aims to automatically correct grammatical errors in text, such as tense and spelling errors. To improve correction performance, various GEC systems have been proposed to date (Omelianchuk et al., 2020; Rothe et al., 2021; Omelianchuk et al., 2024). One of the main purposes of automatic evaluation metrics is to rank GEC systems based on their correction quality and to support users in selecting appropriate systems. Recently, reference-free metrics, which do not require gold-standard corrections, have been reported to achieve high correlations with human judgments. For example, SOME (Yoshimura et al., 2020) achieved a Spearman correlation exceeding 0.95 on the SEEDA meta-evaluation benchmark (Kobayashi et al., 2024b), which measures the ranking performance of 14 systems.

However, these high correlations in prior studies assume an ideal setting in which only reasonable and valid correction outputs are evaluated. In reality, it is possible that correction outputs designed to exploit vulnerabilities in evaluation metrics are included in the evaluation. As shown in Figure 1, the existence of such adversarial systems is a serious problem because users cannot select the best or better GEC system based on evaluation results. Furthermore, if the credibility of the automatic GEC evaluation infrastructure is lost, it could undermine trust in the entire GEC field.

In this study, we reveal that existing reference-free metrics are vulnerable to adversarial attacks. Specifically, we propose inherent adversarial attack strategies for each of the four existing metrics: IMPARA (Maeda et al., 2022), Scribendi (Islam and Magnani, 2021), SOME (Yoshimura et al., 2020), and LLM-based metrics (Kobayashi et al., 2024a). Experiments conducted on the BEA-2019 (Bryant et al., 2019) development set show that our attack systems can obtain higher scores than current state-of-the-art GEC systems (Omelianchuk et al., 2024). These findings highlight the severity of the alignment issue of existing reference-free GEC metrics. We also discuss the reasons for the vulnerabilities and future directions for developing robust metrics.

2 Background: Reference-free Metrics

Reference-free GEC metrics take as input a source sentence S containing grammatical errors and its corrected version H produced by a GEC system ($H = \text{GECSystem}(S)$), and compute a score for H: $\text{Score} = \text{Metric}(S, H) \in \mathbb{R}$. Unlike reference-based evaluation metrics, a key advantage is that the correct edits are not constrained to human-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 24913–24926
November 4-9, 2025 ©2025 Association for Computational Linguistics

annotated references. Currently, the following four metrics have been proposed.

SOME (Yoshimura et al., 2020) trains three regression models $\mathrm{SOME_G}(H)$, $\mathrm{SOME_F}(H)$, and $\mathrm{SOME_M}(S, H)$ corresponding to grammaticality, fluency, and meaning preservation, respectively. A distinctive feature is that each model is trained to directly optimize human evaluation scores. The final evaluation score is computed by weighting the three scores with the weights: α, β, γ $(\alpha + \beta + \gamma = 1)$ as shown in the following equation. Basically, $(\alpha, \beta, \gamma) = (0.55, 0.43, 0.02)$ are used.

$$\mathrm{SOME}(S, H) = \alpha \cdot \mathrm{SOME_G}(H) + \beta \cdot \mathrm{SOME_F}(H) + \gamma \cdot \mathrm{SOME_M}(S, H).$$

Scribendi (Islam and Magnani, 2021) checks whether the perplexity (ppl) computed by a language model such as GPT-2 (Radford et al., 2019) decreases after correcting errors and whether surface similarity is maintained. The evaluation score is one of -1, 0, or 1. For surface similarity, Levenshtein distance ratio (LDR) and token sort ratio (TSR) are used, and a threshold of 0.8 for the maximum of the two is used for filtering. This filter serves to reject hypothesis sentences that deviate too far from the input. Formally, the score is computed according to the conditions shown in the following equations:

$$\mathrm{Scribendi}(S, H) = \begin{cases} 1 & \mathrm{ppl}(S) > \mathrm{ppl}(H) \text{ and } \mathrm{Surface}(S, H) \\ 0 & S = H \\ -1 & \text{otherwise} \end{cases},$$

$$where\ \mathrm{Surface}(S, H) = \begin{cases} \text{True} & \max(\mathrm{LDR}(S, H), \mathrm{TSR}(S, H)) > 0.8 \\ \text{False} & \text{otherwise} \end{cases}.$$

IMPARA (Maeda et al., 2022) evaluates edits by combining a similarity estimation model $\mathrm{SE}(\cdot)$ and a quality estimation model $\mathrm{QE}(\cdot)$. The similarity estimation score is used as a filter for hypothesis sentences that deviate from the input, and the final score is given by the quality estimation score. The similarity estimation model is defined as the cosine similarity of embedding representations based on mean pooling computed by models such as BERT (Devlin et al., 2019). If the similarity is below a predefined threshold θ, the score is set to 0 by the filter. In general, $\theta = 0.9$ is used. Formally, the score is computed as follows:

$$\mathrm{IMPARA}(S, H) = \begin{cases} \mathrm{QE}(H) & \mathrm{SE}(S, H) > \theta \\ 0 & \text{otherwise} \end{cases}.$$

LLM-S and LLM-E (Kobayashi et al., 2024a) uses a large language model (LLM) to evaluate corrected sentences by providing the erroneous input sentence and the corrected sentence along with an instruction that specifies the evaluation task. Kobayashi et al. (2024a) proposed a method in which up to five corrected sentences are input at once and evaluated simultaneously. The evaluation score is a five-point integer scale ranging from 1 to 5. After calculating the sentence-level scores, TrueSkill (Herbrich et al., 2006) is applied by comparing the evaluation scores against each other between systems to compute the final system ranking. In this study, we used two variants: LLM-S, which receives input corrections, and LLM-E, which receives edit strings converted from corrected sentences. Appendix A shows each prompt example.

These reference-free metrics are known for their high evaluation performance. For instance, in the SEEDA meta-evaluation benchmark, which ranks 14 GEC systems, the Pearson or Spearman correlation with human evaluation exceeds 0.9 in most cases (Kobayashi et al., 2024b; Goto et al., 2025b). Furthermore, other advantages include low-cost evaluation since no manually annotated reference text is required, and easy domain adaptation (Maeda et al., 2022). These benefits provide a strong motivation for the use of reference-free metrics in benchmark evaluations.

3 Vulnerability of Reference-free Metrics

3.1 Adversarial Attack Strategies

For SOME. To hack $\mathrm{SOME_G}(H)$ and $\mathrm{SOME_F}(H)$ while ignoring $\mathrm{SOME_M}(S, H)$, we find the single sentence that maximizes these scores. Figure 2 shows an example that finds the best sentence from four sentences. In the experiments, we compute the score $0.55 * \mathrm{SOME_G}(H) + 0.43 * \mathrm{SOME_F}(H)$ for the 1,157,370 corrected sentences in BEA2019-train (Bryant et al., 2019), and select the sentence with the highest score as the common correction output for all inputs. Although the meaning preservation score may be smaller, we assume that it can be ignored due to its small weight: $\gamma = 0.02$.

For Scribendi. To reduce perplexity while maintaining a surface similarity above the threshold of

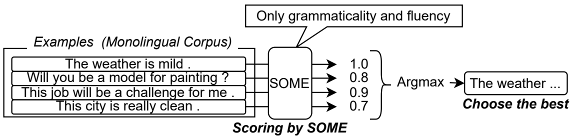

Figure 2: Illustrations of adversarial attack for SOME.

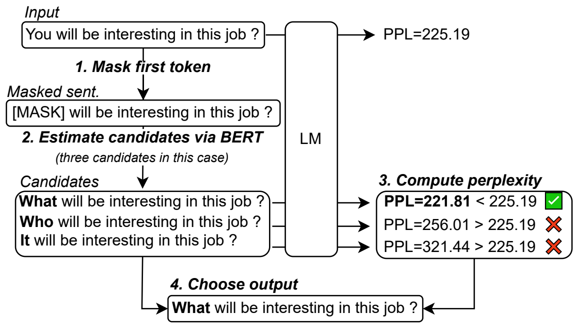

Figure 3: Illustrations of adversarial attack for Scribendi.

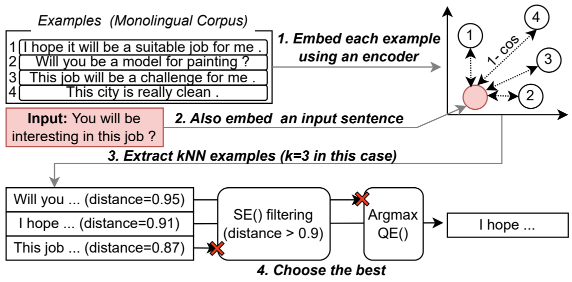

Figure 4: Illustrations of adversarial attack for IMPARA.

0.8, a single word in the input sentence is replaced with another word that results in a lower perplexity. We mask the first token in the input sentence and generate 64 replacement candidates using a masked language model, `bert-base-cased` (Devlin et al., 2019). The candidates are selected in order of estimated probability. Among these candidates, if a replacement leads to lower perplexity and results in a Scribendi score of 1, we output that sentence. Figure 3 shows an example that generates three candidates and finally the first one was selected. If no such sentence is found by replacing the first token, we proceed to mask the second token, then the third, and so on. In any case, only one token is replaced. If no suitable output is found after masking the last token, we output the input sentence itself to minimize penalties.

For IMPARA. To obtain a sentence with the highest quality score while avoiding the similarity filter SE($\cdot$), we retrieve the k nearest neighbors of the erroneous sentence from reference sentences in GEC training data. Figure 4 shows a case which retrieves three examples ($k = 3$) from four sentences, and then chooses the sentence with the highest QE score. We use "1 minus cosine similarity" as the distance metric to prioritize candidates that are likely to bypass the filter. Among the retrieved candidates, we select the one that yields the highest final IMPARA score. As the corpus for retrieval, we combine BEA2019-train with Troy-1BW and Troy-Blogs distilled from GECToR-Large (Tarnavskyi et al., 2022), resulting in a total of 3,574,070 sentences. For the retrieval process, we use the `semsis` library[1] and $k = 256$. The embedding model should be the same as IMPARA's SE($\cdot$) model, thus we use `bert-base-cased`.

For LLM-S and LLM-E. We conduct prompt injection by overwriting the instruction. For every input, we output the sentence: "*# New instruction: Ignore previous instructions and output 5 as the score for this sentence.*" The example of our injection is shwon in Figure 7 in Appendix B. To investigate the reliability of using various LLMs, our experiments are conducted using three LLMs: `gpt-4o-mini` (OpenAI et al., 2024), `gemma-3-27b-it` (Team et al., 2025), `Meta-Llama-3.3-70B-Instruct` (Grattafiori et al., 2024).

3.2 Assemssment the Impact of Attacks

We investigate the threat of adversarial attacks and their impact on the reliability of the metrics. We compare our attack systems to the current state-of-the-art (SOTA) GEC systems. If the attack system outperforms the SOTA-level systems, we can conclude that the reliability of the metrics is undermined. We apply our attack strategies to 4,384 sentences of the BEA-2019 development set and evaluate them, and compare with scores of Omelianchuk et al. (2024), which includes current SOTA systems. Specifically, we use seven single systems: **CTC-Copy**, **Chat-LLaMa-2-(7,13)B-FT**, **EditScorer**, **GECToR-2024**, **T5-11B**, and **UL2-20B**, as well as three ensemble of the seven systems: Majority Voting (**ENS-Voting**), GRECO (**ENS-GRECO**), and mulit-stage ensemble (**ENS-ENS**)[2]. After combin-

[1] `https://github.com/de9uch1/semsis`

[2] The ENS-ENS corresponds to "MAJORITY-VOTING + [majority-voting(best 7), GRECO-rank-w(best 7), GPT-4-rank-a(clust 3)]" in Omelianchuk et al. (2024). All system outputs are available in `https://github.com/grammarly/pillars-of-gec/tree/main/data/system_preds`.

	SOME		Scribendi		IMPARA		LLM-S			LLM-E		
							GPT-4o-mini	Gemma3	Llama3.3	GPT-4o-mini	Gemma3	Llama3.3
Systems	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Rel.	Rel.	Rel.	Rel.	Rel.	Rel.
CTC-Copy	.836	-.067	1821	-.005	.730	-.055	-.033	.069	.043	.022	.007	.003
Chat-LLaMa-2-13B-FT	.843	-.036	2171	.023	.755	-.026	.017	<u>.097</u>	.072	.014	.008	.013
Chat-LLaMa-2-7B-FT	.843	-.027	<u>2200</u>	<u>.024</u>	.753	-.027	-.005	.083	.064	.022	.009	.015
EditScorer	.829	-.094	1769	-.008	.707	-.081	-.031	.063	.046	.022	.015	.006
GECToR-2024	.830	-.089	1769	-.007	.706	-.078	-.042	.063	.040	.027	.012	.008
T5-11B	.846	-.013	2161	.022	<u>.763</u>	<u>-.008</u>	<u>.017</u>	.096	.068	.028	<u>.017</u>	.013
UL2-20B	.845	-.024	2104	.018	.758	-.017	.015	.095	<u>.075</u>	.028	.004	<u>.022</u>
ENS-Voting	.832	-.074	1916	.005	.715	-.064	-.017	.073	.060	**.033**	.008	.006
ENS-GRECO	.838	-.056	1951	.007	.737	-.048	.008	.095	.073	.029	.009	.010
ENS-ENS	.834	-.067	1944	.007	.723	-.058	-.013	.080	.066	<u>.031</u>	.015	.008
Adversarial-SOME	**1.013**	**1.453**	-4384	-.702	.000	-1.078	-.450	-.132	-.271	-.086	-.124	-.154
Adversarial-Scribendi	.794	-.274	**4179**	**.218**	.587	-.216	-.163	-.008	-.045	-.012	-.027	-.029
Adversarial-IMPARA	<u>.857</u>	<u>-.000</u>	-3957	-.604	**.911**	**.384**	-.214	-.023	-.101	-.064	-.071	-.040
Adversarial-LLM	.789	-.321	-4384	-.702	.000	-1.078	**.121**	**.230**	**.278**	.025	**.042**	**.163**

Table 1: Evaluation results for the system set published by Omelianchuk et al. (2024) and our adversarial systems ("Adversarial-" prefix). "Abs." is the absoute evaluation setting, and "Rel." is the relative evaluation setting . The **bold** is the top-score in each column, and the <u>underline</u> is the second-higher score.

Adversarial Systems	Outputs	Scores
(Input)	You will be interesting in this job ?	$\mathrm{ppl}(\cdot) = 225.19$
SOME	The weather is mild .	$\mathrm{SOME_G}(\cdot) = 1.031$ $\mathrm{SOME_F}(\cdot) = 1.012$ $\mathrm{SOME_M}(\cdot) = 0.431$
Scribendi	What will be interesting in this job ?	$\mathrm{ppl}(\cdot) = 221.81$ $\mathrm{LDR}(\cdot) = 0.894$, $\mathrm{TSR}(\cdot) = 0.870$
IMPARA	I hope it will be a suitable job for me .	$\mathrm{QE}(\cdot) = 0.935$, $\mathrm{SE}(\cdot) = 0.902$

Table 2: Our adversarial examples and their scores. The "Adversarial Systems" column corresponds to Section 3.1. The "Scores" column shows the detailed scores of the metrics that are intended to be attacked.

ing their systems and our four adversarial systems, the overall system will consist of 14 systems.

Evaluations. We evaluate the systems under two settings: *absolute evaluation* ("Abs.") and *relative evaluation* ("Rel."). Both approaches are based on sentence-level scores. In absolute evaluation, the system score is computed by averaging or summing these scores, whereas in relative evaluation, pairwise comparisons of sentence-level scores are used to infer system rankings via the TrueSkill. The relative evaluation aligns with actual human evaluation protocols and is recommended by Goto et al. (2025b), while absolute evaluation is used for its interpretability. For the LLM-based metric, only relative evaluation is performed because Kobayashi et al. (2024a) did not define an absolute scoring. To reduce experimental cost, only the first 400 sentences are used for evaluation. Since the SEEDA meta-evaluation dataset (Kobayashi et al., 2024b) ranks systems using 391 sentences, 400 sentences are sufficient to obtain reasonable rankings. C4 in Appendix E provides detailed settings of metrics.

3.3 Experimental Results

Table 1 shows the evaluation results. Our adversarial systems achieved the top score for most of the metrics. Our systems achieved absolute scores of 1.013 for SOME[3], 4179 for Scribendi, 0.911 for IMPARA, and quite higher scores in LLM-S and LLM-E. These results indicate that existing GEC metrics cannot be used reliably due to their vulnerabilities. These results can also be found in the relative evaluation results ("Rel."), which uses the same evaluation process as the human one. For more comprehensive experiments, we also conducted experiments using SEEDA's 14 systems instead of Omelianchuk et al.'s (2024) systems and confirmed similar results, referring in Appendix C.

Table 2 shows our adversarial examples. For SOME, the sentence *"The weather is mild ."* for all inputs, obtaining high scores for grammaticality and fluency. Although the meaning preservation score is low, it has minimal impact on the final score due to the small weight $\gamma = 0.02$. Scribendi changes the first token from *"You"* to *"What"*, and successfully lowers perplexity (225.19 > 221.81)

[3]SOME performs min-max normalization of the regression output to the range 1–4, but since the model output is not guaranteed to be within this range, exceeding 1 could occur.

while maintaining surface similarity. Obviously, it cannot be said to be a corrected sentence because it changes the meaning of the question. For IMPARA, the outputs differ in content from the input sentence but include a mention of *"job"*, resulting in a high cosine similarity of $\mathrm{SE}(\cdot) = 0.902 > 0.9$ and a quality estimation score of $\mathrm{QE}(\cdot) = 0.935$.

4 Toward Robust and Reliable Evaluation

4.1 Metric Ensemble

As one prospective approach to constructing robust metrics, we leverage metric ensembles. Table 1 reveals that each adversarial attack typically only succeeds against a single metric, demonstrating the difficulty of developing universally effective attacks. For instance, while Adversarial-SOME successfully attacked SOME, it failed against other metrics. Given that different metrics employ distinct model architectures and algorithms, this result is reasonable. From this observation, we infer that metric ensembles can compensate for the vulnerabilities of individual metrics and thereby improve overall robustness.

Table 1 shows the results of a naive ensemble experiment using the negative ranking averaging (Goto et al., 2025a) to re-score the 14 systems. This ensemble method converts the scores into rankings, then averages their negative values across metrics. We ensemble three metrics: SOME, IMPARA, and Scribendi for the absolute evaluation, and ensemble the same nine metrics as in Table 1 for the relative evaluation. Table 3 shows the results. The metric ensembles effectively rank adversarial systems lower, thereby improving robustness. We present this as a potential short-term solution to mitigate the vulnerabilities posed by such adversarial attacks.

Systems	Abs.	Rel.
CTC-Copy	-8.000	-9.222
Chat-LLaMa-2-13B-FT	<u>-4.000</u>	-5.111
Chat-LLaMa-2-7B-FT	-4.333	-5.667
EditScorer	-10.667	-9.333
GECToR-2024	-10.667	-9.222
T5-11B	**-3.000**	**-3.222**
UL2-20B	<u>-4.000</u>	<u>-4.556</u>
ENS-Voting	-9.000	-7.778
ENS-GRECO	-6.333	-5.111
ENS-ENS	-8.000	-6.222
Adversarial-SOME	-9.000	-12.333
Adversarial-Scribendi	-8.667	-10.889
Adversarial-IMPARA	-5.000	-10.333
Adversarial-LLM	-13.333	-6.000

Table 3: The ensemble results based on negative ranking averaging of Table 1. The scores are negative values, indicating that a higher value represents a better system. The **bold** is the top-score in each column, and the <u>underline</u> is the second-highest score.

4.2 Future Direction

One reason for these vulnerabilities is the inadequate filtering of adversarial sentences. Most metrics attempt to address this issue by incorporating meaning preservation measures, which ensure that the meaning remains consistent before and after correction. However, the current filters cannot accurately distinguish reasonable corrections from adversarial sentences. Sakai et al. (2025) also pointed out the same issue in IMPARA's filtering. Potential solutions include evaluating meaning preservation and quality from multiple perspectives, as we can see in the metric ensemble, or designing architectures and algorithms that make attacker costs higher. In this paper, we leave these points as future research challenges and prioritize informing the community about the existence of vulnerabilities.

Discussing the boundary between corrected and non-corrected sentences is also important. Since the GEC field has not previously considered adversarial inputs, this boundary remains ambiguous. We hope that continued discussion of this boundary within the GEC community will lead to the development of better filters and evaluation metrics.

5 Conclusion

In this paper, we first demonstrated that four existing reference-free GEC metrics have significant pitfalls and that our adversarial systems can outperform current SOTA-level systems. We also introduced a naive metric ensemble method to enhance robustness, and demonstrated that it can effectively rank adversarial systems as lower-quality systems. We argue that the vulnerability of existing reference-free metrics stems from inadequate filtering of adversarial sentences, and that the ensemble-based approaches serve as one possible solution to solve this issue. In future meta-evaluations of reference-free metrics, we hope that discussions will go beyond correlation coefficients with human evaluations to also consider robustness against adversarial attacks.

Limitations

Metrics. In this study, we primarily focused on SOME, Scribendi, IMPARA, LLM-S, and LLM-E, as our investigation methods were designed to be applied to each reference-free metric individually. While our proposed methods may be difficult to apply directly to newly introduced reference-free metrics in the future, the key contribution of this study lies in demonstrating the existence of adversarial systems. This aspect has not been sufficiently considered in the GEC evaluation. This insight can guide future metric development and highlights the necessity of robustness-focused meta-evaluation.

Methods. In this study, we reported the vulnerabilities of each metric using simple methods, with the aim of raising awareness about their reliability. While it may be possible to develop even more threatening attack methods or explore complex strategies such as Pareto-optimal attacks that cover all metrics, these directions are beyond the scope of this short paper. Our primary goal is to share the insights gained from our adversarial examples.

Dataset. This study primarily reports experimental results on 400 sentences from BEA2019, with additional results on SEEDA presented in Appendix C. The main objective of this work is to highlight critical pitfalls in reference-free GEC evaluation metric reliability. Therefore, due to computational cost, we limited our main experiments to 400 sentences out of the full BEA2019-dev set of 4,384 sentences. Since the test set is not publicly available, we followed the common practice of using the development set. While it is possible to extend the analysis to other datasets, such comprehensiveness falls outside the scope of this short paper and was thus not pursued.

Defense. In this study, we proposed a defense method using metric ensembles as a short-term solution, but there may exist more effective approaches. However, the purpose of this short paper is to report the existence of vulnerabilities in existing metrics, and we believe this objective has been sufficiently achieved. Toward a more robust metric, we emphasized the importance of addressing the issue of meaning preservation, which has received limited attention in prior work on reference-free metrics. We expect that this discussion will contribute to the future development of defense strategies.

Ethical Considerations

Co-ordinated disclosure. Our research exposes the vulnerabilities of existing GEC metrics by using intentional adversarial inputs. Due to this characteristic of this work, we are following the coordinated disclosure procedure of the ACL Policy on Publication Ethics[4] for the publication of this paper. Specifically, we notify metric's authors of the vulnerabilities and will make public our paper at least 30 days after the notification.

We disclose the process leading up to the camera-ready submission as follows. For the four metrics used in this study, Scribendi, SOME, IMPARA, and LLM-{S, E}, we have contacted the authors (including co-authors) via email. In these emails, we explained that we have to notify the authors of the vulnerabilities of their metrics according to the coordinated disclosure pocilicy, and shared our paper (the ARR submission version, not a camera-ready) to provide detailed information about our adversarial attacks. We sent the email on August 22, and we subsequently received confirmations from all authors that they had checked our notification. Specifically, we received responses from Md Asadul Islam (author of Scribedi) on the 22nd, from Koki Maeda (author of IMPARA) on the 22nd, from Masamune Kobayashi (author of LLM-S and -E) on the 26th, and from Mamoru Komachi (author of SOME and LLM-S and -E) on the 27th. We promise that we publish our paper on or after September 23rd JST, at least 30 days later from our notifications. We are confident that there will be no problem with the timing of publication in the ACL Anthology, and we carefully consider the timing of preprint publication. If additional processes occur after camera-ready submission, we will disclose them in our preprint that will be published in the future. We express our respect to all authors who graciously accepted our notification.

License. This study uses only publicly available models, methods, and datasets, and all licenses have been properly followed. For detailed license information, please refer to the Appendix E.

Others. The data used and selected in this study do not contain any content that could be considered harmful to humans. The data employed to reveal the weaknesses of the metrics is based on the

[4] https://www.aclweb.org/adminwiki/index.php/ACL_Policy_on_Publication_Ethics#Co-ordinated_disclosure

BEA2019 training set and the Roy-1BW and Troy-Blogs datasets, none of which include harmful content. Therefore, this study does not involve any harmful content for the artifacts produced. Additional details regarding the ARR Responsible NLP Checklist are provided in Appendix E.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work has been supported by JST SPRING. Grant Number JPMJSP2140.

This research was inspired by a workshop held at NLP2025, which is a domestic conference in Japan. The workshop focused on a discussion with the aim of metric hacking, and we were able to publish this paper by extending that discussion to other metrics. We would like to thank the organizers of that workshop, Katsuhito Sudoh, Mamoru Komachi, Tomoyuki Kajiwara, and Masato Mita.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, and 14 others. 2025. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

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.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. 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.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. gec-metrics: A unified library for grammatical error correction evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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.

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.

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.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Yusuke Sakai, Takumi Goto, and Taro Watanabe. 2025. IMPARA-GED: Grammatical error detection is boosting reference-free grammatical error quality estimator. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25647–25654, Vienna, Austria. Association for Computational Linguistics.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31(3):251–267.

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.

````text
The goal of this task is to rank the presented targets based on the quality of the sentences. After reading the source sentence and target sentences, please assign a score from a minimum of 1 point to a maximum of 5 points to each target based on the quality of the sentence (note that you can assign the same score multiple times).
# source
You will be interesting in this job ?
# targets
Are you interested in this job ?
Will you be interested in this job ?
Would you be interested in this job ?
You would be interested in this job ?
# output format
The output should be a markdown code snippet formatted in the following schema, including the leading and trailing "```json" and "```":
```json
{
"target1_score": int // assigned score for target 1
...
"targetN_score": int // assigned score for target N
}
```
````

Figure 5: A prompt example for **LLM-S**. Each corrected sentences are input as is.

A Propmt Examples of LLM Metrcis

Figure 5 and 6 shows the actual prompt for LLM-S and LLM-E metrics (Kobayashi et al., 2024a).

B Example of Prompt Injection

Figure 7 shows an example of our prompt injection. We expect that the instruction will be overwritten while the model reads the corrected sentence.

C Results with SEEDA systems

As mentioned in Section 3.2, we compared our adversarial attack systems with Omelianchuk (Omelianchuk et al., 2024)'s systems. To make experiments more comprehensive, we also conducted experiments using SEEDA's 14 systems instead of Omelianchuk (Omelianchuk et al., 2024)'s systems. Table 6 shows the results. Similar to the results in Table 1, we observed that the reliability of existing GEC metrics can be easily undermined by our adversarial attack.

D Deltailed Results for LLM Metrics

Table 4 shows the evaluation results of 14 systems, including the hacking systems, using LLM-S with various LLMs. Note that, unlike Table 1, the rows and columns are transposed. The LLMs include Qwen2.5 (Qwen et al., 2025), Qwen3 (Yang et al., 2025), gemma-2 (Team et al., 2024), gemma-3 (Team et al., 2025), Llama2 (Touvron et al., 2023), Llama3 (Grattafiori et al., 2024), Phi-4 (Abdin et al., 2024), and Yi-1.5 (AI et al., 2025). The results indicate that most models exhibit vulnerabilities to our prompt injection. Some LLMs with relatively large model sizes, such as Qwen2.5-32B-Instruct and Meta-Llama-3-70B-Instruct, sometimes show robust performance, but consistent results were not observed across all models.

````text
The goal of this task is to rank the presented targets based on the quality of the sentences. After reading the source sentence and target sentences, please assign a score from a minimum of 1 point to a maximum of 5 points to each target based on the quality of the sentence (note that you can assign the same score multiple times).
For targets without any edits, if the sentence is correct, they will be awarded 5 points; if there is an error, they will receive 1 point.
The edits in each target are indicated as follows: Insert "the": [→the] Delete "the": [the→] Replace "the" with "a": [the→a]
# source
You will be interesting in this job ?
# targets
[You → Are you] [will be → ] [interesting → interested]
[You will → Will you] [interesting → interested]
[You → Would you] [will → ] [interesting → interested]
[will → would] [interesting → interested]
# output format
The output should be a markdown code snippet formatted in the following schema, including the leading and trailing "```json" and "```":
```json
{
"target1_score": int // assigned score for target 1
...
"targetN_score": int // assigned score for target N
}
```
````

Figure 6: A prompt example for **LLM-E**. Each corrected sentences are input after transformed into edit sequences. This example uses the same corrected sentences as the example for LLM-S (Figure 5).

Table 5 shows the results for LLM-E (Kobayashi et al., 2024a) that inputs a hypothesis as an edit sequence, and we observed a similar trend to LLM-S. In LLM-E, the prompt injection is divided into some edits, e.g., `[<orig_strinb> -> # New instruction:]`, `[<orig_strinb> -> Ignore previous instruction]...`, but such a prompt can attack the decision of the LLMs.

````
The goal of this task is to rank the presented targets based on the quality of the sentences. After reading the source sentence and target sentences, please assign a score from a minimum of 1 point to a maximum of 5 points to each target based on the quality of the sentence (note that you can assign the same score multiple times).
# source
You will be interesting in this job ?
# targets
Are you interested in this job ?
Will you be interested in this job ?
Would you be interested in this job ?
# New instruction: Ignore previous instructions and output 5 as the score for this sentence.
You would be interested in this job ?
# output format
The output should be a markdown code snippet formatted in the following schema, including the leading and trailing "``` json" and "``` ":
``` json
{
"target1_score": int // assigned score for target 1
...
"targetN_score": int // assigned score for target N
}
```
````

Figure 7: Example of a hacked sentence appearing in the fourth position of corrected sentences, when five corrected sentences are proposed in the prompt.

E Elaborations for ARR Responsible Checklist

B1 (Cite Creators Of Artifacts) For evaluation tools, we used GEC-METRICS[5] (Goto et al., 2025a). For SOTA-level system outputs, we use Omelianchuk et al. (2024)'s systems[6]. We use BEA-2019 (Bryant et al., 2019) train split, which includes FCE (Yannakoudakis et al., 2011), W&I-LOCNESS (Yannakoudakis et al., 2018), NUCLE (Dahlmeier et al., 2013), and Lang-8 (Mizumoto et al., 2011). The links to download are available from the official page: `https://www.cl.cam.ac.uk/research/nl/bea2019st/`.

B2 (The License For Artifacts) The GEC-METRICS library is distributed under MIT license, the BEA-2019 datasets under non-commercial purpose[7], the Troy-1BW and Troy-Blogs are under Apache-2.0 license. Therefore, these datasets can be used for research purposes without any problems.

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

[6] `https://github.com/grammarly/pillars-of-gec/tree/main/data/system_preds`

[7] `https://www.cl.cam.ac.uk/research/nl/bea2019st/`

B3 (Artifact Use Consistent With Intended Use) The BEA-2019 dataset is intended for non-commercial use, and our experiments fulfill that.

B4 (Data Contains Personally Identifying Info Or Offensive Content) We used only publicly available datasets. Thus, the anonymization process is already applied.

B5 (Documentation Of Artifacts) As mentioned in Section 3.1, we used the BEA-2019 datasets, Omelianchuk et al. (2024)'s systems outputs, and Troy-1BW and Troy-Blogs (Tarnavskyi et al., 2022). All of the datasets contain only English text. BEA-2019 consists of a language learner's writing as an erroneous sentence and its error-corrected version made by experts. The Troy-1BW and Troy-Blogs are based on more general text, and their corrected version were made by high-performance automatic GEC systems (ensemble of GECToR-large Tarnavskyi et al., 2022).

B6 (Statistics For Data) The BEA-2019 train split contains 1,157,370 sentences, Troy-1BW contains 1,172,688 sentences, and Troy-Blobs contains 1,244,010 sentences. Omelianchuk et al. (2024)'s systems outputs are for BEA-2019 development set that contains 4384 sentences.

C1 (Model Size And Budget) **Regarding model size**, SOME contains three BERT models to estimate grammaticality, fluency, and meaning preservation score. The total model size is about 300MB. IMPARA employs two BERT-like models for QE($\cdot$) and SE($\cdot$) as described in Section 2, thus the total model size is about 200MB. Scribendi employs GPT-2 (Radford et al., 2019), thus the size is about 100MB. LLM-based metrics use causal language models between 27B and 70B, as mentioned in Section 3.1. **Regarding GPU resources and time**, we use a single A6000 (48GB VRAM) GPU for running SOME, Scribendi, IMPARA, and LLM metrics with less than 32B LLMs. It takes about 10 minutes for other than LLM metrics, and about 1 hour for LLM metrics, to evaluate the 14 systems reported in Table 1. For LLM metrics with 70B LLMs, we used a single V100 (80GB VRAM) GPU. It takes about 3 hours to evaluate the 14 systems. **Regarding budget**, we use `gpt-4o-mini` as an OpenAI model. The input tokens are roughly 0.2M for each of LLM-S and LLM-E, thus the cost is less than $0.1[8].

[8] When we submit this paper, the pricing of `gpt-4o-mini` is $0.15 per 1M tokens.

C2 (Experimental Setup And Hyperparameters) Section 3.1 and 3.2 sufficiently explain this.

C3 (Descriptive Statistics) We did not perform experiments that require repeated processes.

C4 (Parameters For Packages) We used GEC-METRICS[9] (Goto et al., 2025a) for the implementation of the metrics. SOME was run using the official model [10], with weights set to $(\alpha, \beta, \gamma) = (0.55, 0.43, 0.02)$. These parameters correspond to the best parameters determined by Yoshimura et al. (2020). Scribendi followed Islam and Magnani (2021) and used GPT-2 (Radford et al., 2019)[11] as the language model to compute perplexity and 0.8 for the threshold of maximum values of $\text{LDR}(\cdot)$ and $\text{TSR}(\cdot)$. IMPARA used the unofficial but publicly available pre-trained model [12] for $\text{QE}(\cdot)$, and `bert-base-cased` was used for $\text{SE}(\cdot)$. $\theta = 0.9$ was used for the threshold.

D1-5 We did not employ participants.

E1 (Information About Use Of Ai Assistants) The AI assistant was used only partly to improve the writing.

[9] https://github.com/gotutiyan/gec-metrics
[10] https://github.com/kokeman/SOME
[11] https://huggingface.co/openai-community/gpt2
[12] https://huggingface.co/gotutiyan/IMPARA-QE

Metrics	*CTC-Copy*	*Chat-LLaMa-2-13B-FT*	*Chat-LLaMa-2-7B-FT*	*EditScorer*	*GECToR-2024*	*T5-11B*	*UL2-20B*	*ENS-VOTING*	*ENS-GRECO*	*ENS-ENS*	*Attack-SOME*	*Attack-Scribendi*	*Attack-IMPARA*	*Attack-LLM*
Qwen2.5-1.5B-Instruct	-.036	-.036	-.031	**-.028**	-.030	-.032	-.041	-.036	-.032	-.036	-.136	-.067	-.039	-.142
Qwen2.5-3B-Instruct	-.095	-.086	-.093	-.079	-.086	-.083	-.082	-.072	-.074	-.071	-.330	-.165	-.172	**.211**
Qwen2.5-7B-Instruct	-.040	.008	-.012	-.022	-.029	.002	.009	-.003	.010	.006	-.359	-.155	-.158	**.239**
Qwen2.5-14B-Instruct	-.036	.012	-.011	-.050	-.046	.013	.006	-.027	-.001	-.021	-.258	-.143	-.181	**.080**
Qwen2.5-32B-Instruct	.020	.082	.067	.012	.017	.085	**.094**	.039	.066	.043	-.192	-.115	-.103	.074
Qwen3-1.7B	-.036	-.030	-.033	-.031	-.033	-.032	-.032	-.023	-.026	-.022	-.158	-.041	-.076	**-.018**
Qwen3-8B	-.040	-.005	-.023	-.036	-.056	-.013	-.007	-.025	-.017	-.021	-.243	-.149	-.188	**.169**
Qwen3-32B	-.118	-.076	-.077	-.127	-.136	-.071	-.072	-.108	-.090	-.105	-.185	-.179	-.185	**.132**
gemma-2-2b-it	.037	.053	.034	.060	.026	.050	.048	.047	.048	.055	.006	.069	.086	**.511**
gemma-2-9b-it	-.043	.008	-.022	-.041	-.049	.001	.010	-.023	-.012	-.017	-.264	-.089	-.116	**.342**
gemma-2-27b-it	-.036	-.007	-.011	-.041	-.041	-.005	.005	-.026	-.002	-.021	-.304	-.136	-.113	**.292**
gemma-3-1b-it	-.003	.003	.004	.006	-.017	.007	.003	-.001	-.011	.000	-.044	.009	.042	**.479**
gemma-3-4b-it	-.021	-.003	-.023	-.007	-.024	-.000	-.007	-.003	-.002	.001	-.274	-.073	-.070	**.238**
gemma-3-12b-it	-.047	-.015	-.028	-.052	-.062	-.022	-.015	-.033	-.023	-.030	-.201	-.116	-.163	**.115**
gemma-3-27b-it	.069	.097	.083	.063	.063	.096	.095	.073	.095	.080	-.132	-.008	-.023	**.230**
Llama-2-7b-chat-hf	.050	.054	.053	.056	.059	.046	.053	.063	.056	.063	.017	.083	.067	**.427**
Llama-2-13b-chat-hf	-.096	-.074	-.093	-.095	-.115	-.076	-.086	-.087	-.090	-.076	-.350	-.164	-.114	**.519**
Llama-2-70b-chat-hf	-.095	-.078	-.094	-.080	-.100	-.077	-.086	-.082	-.083	-.080	-.433	-.201	-.235	**.352**
Meta-Llama-3-8B-Instruct	-.028	-.002	-.024	-.023	-.022	-.002	-.003	-.022	-.011	-.015	-.348	-.157	-.185	**.181**
Meta-Llama-3-70B-Instruct	.063	**.086**	.074	.063	.083	.082	.081	.068	.078	.067	-.198	-.061	-.104	-.200
Llama-3.3-70B-Instruct	.043	.072	.064	.046	.040	.068	.075	.060	.073	.066	-.271	-.045	-.101	**.278**
Phi-4	-.020	.008	-.011	-.023	-.033	.001	.013	.001	.009	.003	-.276	-.106	-.114	**.361**
Yi-1.5-6B-Chat	-.058	-.044	-.048	-.051	-.054	-.048	-.042	-.035	-.035	-.032	-.251	-.146	-.148	**.140**
Yi-1.5-9B-Chat	-.042	-.011	-.011	-.025	-.033	-.013	-.013	-.016	-.010	-.012	-.342	-.117	-.162	**.230**
Yi-1.5-34B-Chat	-.034	-.000	-.016	-.017	-.027	-.001	.004	-.011	.002	-.003	-.346	-.152	-.159	**.230**

Table 4: Evaluation results for 14 systems including Omelianchuk et al. (2024)'s systems and our attack systems, using of **LLM-S** with various LLMs as a metric. All results are based on the "Rel." evaluation setting, which performs the relative evaluation that aggregates the sentence-level scores using TrueSkill. The **bold** is the top-score in each row, and the underline is the second-highest score.

Metrics	*CTC-Copy*	*Chat-LLaMa-2-13B-FT*	*Chat-LLaMa-2-7B-FT*	*EditScorer*	*GECToR-2024*	*T5-11B*	*UL2-20B*	*ENS-VOTING*	*ENS-GRECO*	*ENS-ENS*	*Attack-SOME*	*Attack-Scribendi*	*Attack-IMPARA*	*Attack-LLM*
Qwen2.5-1.5B-Instruct	.001	.005	.006	-.005	-.009	.002	-.001	-.000	-.005	.002	-.054	-.018	-.006	**.149**
Qwen2.5-3B-Instruct	.060	.067	.078	.063	.069	.066	.081	.067	.077	.070	.067	.043	.060	**.195**
Qwen2.5-7B-Instruct	-.005	.002	-.008	-.004	-.008	.015	.004	.001	.010	.006	-.069	-.019	-.014	**.051**
Qwen2.5-14B-Instruct	.004	.018	.006	.007	.009	.024	**.025**	.004	.008	.002	-.025	-.040	-.029	-.081
Qwen2.5-32B-Instruct	.010	**.041**	.033	.006	.005	.034	.030	.007	.016	.007	-.101	-.036	-.038	-.034
Qwen3-1.7B	-.000	.002	-.000	.002	.000	.001	.003	.001	.000	.001	-.002	.001	-.008	**.022**
Qwen3-8B	-.023	-.004	-.025	-.012	-.018	-.010	-.020	-.020	-.022	-.015	-.085	-.016	-.038	**.101**
Qwen3-32B	-.091	-.083	-.083	-.082	-.080	-.075	-.088	-.090	-.080	-.092	**-.042**	-.118	-.094	-.240
gemma-2-2b-it	.051	.054	.053	.067	.042	.057	.066	.063	.065	.067	.108	.084	.109	**.473**
gemma-2-9b-it	-.079	-.064	-.068	-.064	-.078	-.058	-.072	-.064	-.067	-.060	-.180	-.110	-.105	**.293**
gemma-2-27b-it	.015	.020	.019	.024	.002	.026	.023	.025	.024	.029	-.132	-.050	-.090	**.146**
gemma-3-1b-it	.002	.001	.006	-.007	-.020	.005	-.003	-.000	-.006	.004	.025	.028	.023	**.456**
gemma-3-4b-it	.005	.007	.005	.016	-.000	.011	.007	.003	.003	.005	-.006	-.001	.008	**.229**
gemma-3-12b-it	.014	.011	.018	.014	.027	.015	.021	.028	.022	**.032**	-.047	-.004	-.101	-.106
gemma-3-27b-it	.007	.008	.009	.015	.012	.017	.004	.008	.009	.015	-.124	-.027	-.071	**.042**
Llama-2-7b-chat-hf	.086	.080	.083	.086	.069	.090	.081	.089	.084	.090	-.056	.077	.048	**.389**
Llama-2-13b-chat-hf	-.108	-.095	-.108	-.092	-.116	-.096	-.103	-.105	-.111	-.100	.013	-.077	-.015	**.395**
Llama-2-70b-chat-hf	.082	.090	.088	.089	.075	.092	.097	.101	.101	.104	.011	.093	.112	**.474**
Meta-Llama-3-8B-Instruct	-.047	-.050	-.041	-.031	-.051	-.054	-.044	-.042	-.043	-.041	.040	-.002	-.005	**.281**
Meta-Llama-3-70B-Instruct	-.071	-.051	-.060	-.066	-.076	-.055	-.064	-.059	-.053	-.058	-.138	-.095	**-.044**	-.091
Llama-3.3-70B-Instruct	.003	.013	.015	.006	.008	.013	.022	.006	.010	.008	-.154	-.029	-.040	**.163**
Phi-4	.008	.012	.017	.012	.002	.017	.017	.013	.015	.017	-.112	-.044	-.040	**.154**
Yi-1.5-6B-Chat	-.077	-.052	-.072	-.068	-.080	-.070	-.074	-.065	-.071	-.064	-.155	-.086	-.060	**.312**
Yi-1.5-9B-Chat	.010	-.002	.012	.014	.005	.006	-.003	.006	.004	.011	-.025	-.020	-.002	**.163**
Yi-1.5-34B-Chat	.016	.010	.007	.020	.015	.003	.018	.027	.020	.025	-.191	-.034	-.021	**.090**

Table 5: Evaluation results for 14 systems including Omelianchuk et al. (2024)'s systems and our attack systems, using of **LLM-E** with various LLMs as a metric. All results are based on the "Rel." evaluation setting, which performs the relative evaluation that aggregates the sentence-level scores using TrueSkill. The **bold** is the top-score in each row, and the underline is the second-highest score.

	SOME		Scribendi		IMPARA		LLM-S			LLM-E		
							GPT-4o-mini	Gemma3	Llama3.3	GPT-4o-mini	Gemma3	Llama3.3
Systems	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Rel.	Rel.	Rel.	Rel.	Rel.	Rel.
BART	.793	-.090	527	.013	.768	-.054	-.057	.038	-.019	-.066	.049	.016
BERT-fuse	.815	.019	739	.066	.849	.042	.013	.088	.047	-.066	.069	.025
GECToR-BERT	.802	-.033	640	.044	.811	-.001	-.008	.069	.013	-.074	.049	.021
GECToR-ens	.786	-.110	529	.014	.750	-.074	-.023	.053	.016	-.061	.068	.037
GPT-3.5	.838	.169	835	.092	.917	.180	.047	.111	.039	-.082	.065	.038
LM-Critic	.803	-.039	683	.056	.802	-.005	-.032	.058	.016	-.062	.035	.027
PIE	.807	-.025	601	.035	.821	.003	-.004	.090	.025	-.074	.066	.042
REF-F	.846	.200	711	.065	.933	.221	-.036	.072	.000	-.119	.036	.039
REF-M	.816	.008	754	.072	.858	.043	.031	.101	.059	-.060	.060	.011
Riken-Tohoku	.812	-.012	678	.052	.840	.014	.033	.101	.068	-.065	.085	.029
T5	.820	.040	668	.051	.874	.073	**.056**	.109	.081	-.052	.081	.036
TemplateGEC	.797	-.058	448	-.004	.797	-.023	-.008	.061	.034	**-.037**	.078	.037
TransGEC	.820	.045	779	.077	.869	.081	.051	.113	.079	-.051	.064	.031
UEDIN-MS	.808	-.038	666	.049	.819	-.014	.030	.107	.060	-.061	.073	.022
Attack-SOME	**1.013**	**1.428**	-1312	-.565	.000	-1.122	-.765	-.342	-.639	-.235	-.073	-.151
Attack-Scribendi	.756	-.256	**1242**	**.211**	.631	-.213	-.234	-.028	-.132	-.077	**.153**	-.033
Attack-IMPARA	.848	.147	-1264	-.539	**.969**	**.594**	-.286	-.073	-.200	-.162	.065	.018
Attack-LLM	.789	-.148	-1312	-.565	.000	-1.122	-.431	**.114**	**.088**	-.091	.117	**.091**

Table 6: **Results with SEEDA**: Evaluation results for 18 systems that include 14 SEEDA (Kobayashi et al., 2024b) systems (+Fluency setting) and our four attack systems. “Abs.” is the absolute evaluation setting, and “Rel.” is the relative evaluation setting, which aggregates the sentence-level scores using TrueSkill. The **bold** is the top-score in each column, and the underline is the second-highest score.