JELV: A *J*udge of *E*dit-*L*evel *V*alidity for Evaluation and Automated Reference Expansion in Grammatical Error Correction

Anonymous ACL submission

Abstract

Existing Grammatical Error Correction systems suffer from limited reference diversity, leading to restricted model generalization and underestimated evaluation. To address this issue, We introduce the **Judge of Edit-Level Validity (JELV)**, an automated framework to validate correction edits from grammaticality, faithfulness, and fluency. Using our proposed human-annotated Pair-wise Edit-level Validity Dataset (PEVData) as benchmark, JELV offers two implementations: a multi-turn LLM-as-Judges pipeline achieving 90% agreement with human annotators, and a distilled DeBERTa classifier with 85% precision on valid edits. We then apply JELV to filter LLM-generated correction candidates, expanding the BEA19's single-reference dataset containing 38,692 source sentences. Retraining the leading GEC systems on this expanded dataset yields measurable performance gains. We also apply JELV to reclassify misjudged false positives in evaluation and derive a comprehensive evaluation metric by integrating false positive decoupling and fluency scoring, resulting in state-of-the-art correlation with human judgments. JELV provides a scalable solution for enhancing reference diversity and strengthening both evaluation and model robustness. Code and data are available in `https://anonymous.4open.science/r/JELV-8E3F`.

1 Introduction

Grammatical Error Correction (GEC) aims to detect and correct writing errors in text (Bryant et al., 2023). Typical GEC datasets consist of source sentences and their manually corrected versions (*i.e.*, *references*), which form the basis for training and evaluating GEC systems.

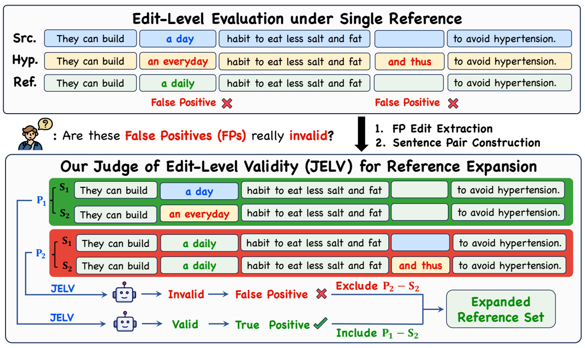

Figure 1: Comparison between biased edit-level evaluation using a single reference and our automated expansion of valid edits via the Judge of Edit-Level Validity (JELV). **Src.**, **Hyp.** and **Ref.** denote the source, hypothesis and reference sentences, respectively. We form two sentence pairs, $\mathbf{P_1}$ and $\mathbf{P_2}$. In each pair, $\mathbf{S_1}$ (from **Src.**) and $\mathbf{S_2}$ (from **Hyp.**) differ only in the single edited segment, remaining identical to the reference elsewhere.

However, creating high-quality corrections for GEC datasets usually requires substantial time and expert effort. Consequently, most GEC datasets (Bryant et al., 2019; Ng et al., 2014; Flachs et al., 2020) contain only one or two references per source sentence. This limited reference set does not represent the numerous valid ways an error can be corrected, causing two main issues: (1) *Limited model performance*: training on a narrow reference set constrains the model to specific correction patterns, leading to poor generalization. (2) *Underestimated evaluation*: reference-based evaluation metrics undercount acceptable corrections that deviate from the given references (Gomez and Rozovskaya, 2024; Zhang et al., 2022a; Choshen and Abend, 2018b). While prior studies show that increasing the number of references improves both evaluation accuracy (Bryant and Ng, 2015) and training effectiveness (Liu et al., 2024), existing reference expansions are still created manually, making them difficult to scale. Thus, an automated approach for generating diverse and valid corrections is crucial to advance research and development in GEC.

However, clearly defining what makes a correction edit *valid* remains challenging in GEC. Current human annotation practices follow two primary guidelines: *Minimal Edit* enforces grammaticality and faithfulness[1] (Ye et al., 2024), and *Fluent Edit* further requires improvements in sentence fluency (Bryant et al., 2023). While most existing datasets adopt the Minimal Edit approach, a more comprehensive standard for validity should recognize overall improvements to the text, including fluency. Based on this, we define an edit as **valid** only if it satisfies all three essential criteria: (1) grammatical correctness, (2) meaning preservation, and (3) fluency improvement—each contributing to enhancing the overall validity of the text.

To provide a gold-standard benchmark for edit-level validity, we introduce the Pair-wise Edit-level Validity Dataset (PEVData). In particular, in PEVData, to identify valid edits for expanding the reference set, we extract each hypothesis sentence[2] that contains FPs and pair it with its source. We then align all other tokens to the reference so that only the edit span differs. Expert annotators subsequently evaluate each pair's validity against our three golden criteria, producing the PEVData[3].

With PEVData as benchmark, we introduce the **J**udge of **E**dit-**L**evel **V**alidity (**JELV**) to automate edit validity assessment. JELV offers two implementations to balance accuracy and inference efficiency. JELV1.0 leverages the evaluation capacity of large language models (LLMs) (Li et al., 2024; Zhu et al., 2023) and implements a multi-turn LLM-as-Judges pipeline, achieving over 90% accuracy compared to human labels on PEVData. To reduce inference cost and enable large-scale deployment, we distill this pipeline into JELV2.0, a lightweight DeBERTa (He et al., 2020) classifier that maintains over 85% precision on valid edits. The complete JELV workflow is illustrated in Figure 1.

We apply JELV in two ways to address the main issues caused by reference scarcity. (1) For *limited model performance*, we employ a generation-then-filtering pipeline to automate reference expansion. By leveraging LLMs' GEC expertise and lower cost than human annotation, we generate candidate edits for BEA19's (Bryant et al., 2019) 38,692 single-reference sentences and use JELV to retain only valid ones. Retraining leading GEC systems on this expanded corpus yields clear performance gains on CoNLL14 benchmark. This demonstrates the effectiveness of our reference expansion strategy in improving model performance. (2) For *underestimated evaluation*, exhaustive enumeration of valid references is infeasible (Choshen and Abend, 2018b), causing reference-based metrics to misclassify valid edits as FPs inevitably. We therefore apply JELV to re-evaluate and reclassify these FPs as true positives. By further decoupling the remaining FPs into overcorrection and non-overcorrection and integrating fluency scoring for fine-grained and comprehensive evaluation, we achieve state-of-the-art correlation with human judgments across multiple evaluation dimensions.

Our contributions are threefold:

1. We introduce JELV to automate edit-level validity assessment, offering (1) an LLM-as-Judges pipeline with >90% accuracy and (2) a distilled DeBERTa classifier with >85% precision, both validated on our human-annotated PEVData benchmark.
2. We propose a JELV-based generation-then-filtering pipeline for automated reference expansion and retraining leading GEC systems on the expanded corpus yields measurable performance gains.
3. We enhance evaluation reliability by using JELV to reclassify FPs as true positives, decoupling the remaining FPs, and integrating fluency scoring, achieving SOTA correlation with human judgments.

2 Related Work

We only discuss the most relevant studies here and provide further discussion in Appendix A.

GEC Reference Expansion Most GEC training corpora, including BEA19-train (Bryant et al., 2019), provide only one reference per source sentence, while evaluation benchmarks typically include two (Bryant et al., 2019; Ng et al., 2014; Flachs et al., 2020) or four (Napoles et al., 2017) references. Manual reference expansion has improved evaluation reliability by adding eight annotations for CoNLL14 (Bryant and Ng, 2015), averaging 2.3 references per sentence in the Chinese MuCGEC dataset (Zhang et al., 2022a), and providing three references per sentence in a Russian GEC corpus (Gomez and Rozovskaya, 2024). Besides, (Liu et al., 2024) has shown that multiple references have great potential for better training effectiveness. However, these approaches depend on extensive human effort and do not scale easily. An automated, scalable method for generating di-

[1] Faithfulness: Corrections should maintain the original textual meaning and syntactic structure.

[2] candidate correction produced by the GEC system

[3] For more details on PEVData, please refer to Sec. 3.2.

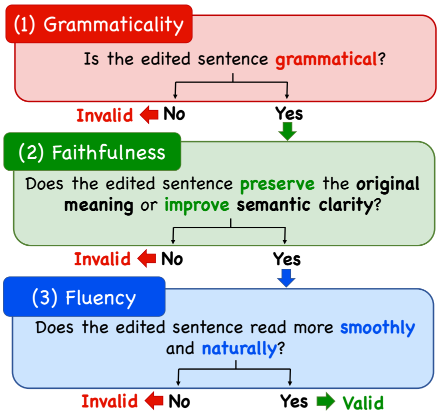
(1) Grammaticality
Is the edited sentence grammatical?
Invalid
No
Yes
(2) Faithfulness
Does the edited sentence preserve the original meaning or improve semantic clarity?
Invalid
No
Yes
(3) Fluency
Does the edited sentence read more smoothly and naturally?
Invalid
No
Yes
Valid

Figure 2: Three criteria for judging edit-level validity.

verse, valid correction edits is therefore essential for advancing GEC.

3 Validity Judgment

3.1 Criteria for Validity

We begin by analyzing edits (*i.e.*, *source*→ *target*) that are truly **valid** but misclassified as FPs due to limited reference coverage. In GEC, these FPs fall into two categories: (1) $\mathbf{FP_{oc}}$ (overcorrection): valid edits applied to grammatically correct text for improving fluency or semantic clarity. (2) $\mathbf{FP_{noc}}$ (non-overcorrection): valid edits applied to ungrammatical text but absent from reference set.

Under the assumption that all text outside the edit span is correct, we define three criteria for edit-level validity to ensure comprehensive improvement: **(1) Grammaticality**: the edit target must be free of grammatical errors. **(2) Faithfulness**: the edit must preserve the original intended meaning or enhance semantic clarity. **(3) Fluency**: the edit must render the sentence more fluent, either by correcting errors or smoothing awkward phrasing[4]. An edit is judged valid only if it meets all three criteria simultaneously (see Figure 2).

3.2 PEVData: Dataset Curation

To construct a benchmark for edit-level validity, we assemble Pair-wise Edit-level Validity Dataset (PEVData) in following three stages.

Sentence Pair Construction Under the Correction Independence Assumption that grammatical error corrections are independent (Ye et al., 2023), we extract each hypothesis sentence containing false positives and pair it with its source. All other tokens are realigned to the reference so that only the edit span varies. This controlled pairing isolates each edit's effect and prevents unrelated changes from influencing validity judgments.

Data Collection We sample single-edit pairs from three public datasets: *CoNLL14* (Ng et al., 2014), *ArgRewrite* (Zhang et al., 2017), and *JFLEG* (Napoles et al., 2017). We provide more details of the datasets in Appendix B.

Annotation Protocol To equip annotators with sufficient context, each sentence pair includes its preceding and following sentences. Three experts (with backgrounds in English teaching, proofreading, and linguistics) applied our three validity criteria to the 1,118 CoNLL14 pairs in three stages: (1) independent labeling with unanimous judgments finalized (2) independent re-labeling of remaining pairs with new unanimous decisions finalized, and (3) a joint discussion to resolve conflicts by consensus or majority vote. ArgRewrite pairs were annotated by a single expert after an initial majority-vote filter, and JFLEG pairs were valid without further review. In total, PEVData comprises 2,797 sentence pairs: 1,118 from CoNLL14, 844 from ArgRewrite and 835 from JFLEG. Annotators judged 1,459 edits as valid and 1,338 as invalid, yielding a balanced benchmark for edit-level validity.

3.3 JELV: Developing an Automated Judge

We propose the Judge of Edit-Level Validity (JELV), a framework for automatically evaluating edit-level validity. To balance high accuracy and inference efficiency, JELV comprises (1) an LLM-as-judges pipeline for high accuracy and (2) a distilled DeBERTa classifier for inference efficiency. Figure 3 shows the overview of the JELV workflow.

3.3.1 JELV1.0: LLM-as-Judges Pipeline

Multi-Turn Optimization JELV1.0 applies DeepSeek-V3 (DS-V3), GPT-4o, and Qwen-Max in sequence. Each model reviews and refines the previous model's explanations and judgments, enabling iterative calibration that corrects early errors and ensures consistency. Qwen-Max's outputs serve as the final validity labels. See Appendix D.1 for prompt details.

In-Context Learning (ICL) DS-V3 first generates explanations for 25 valid and 25 invalid edits sampled from PEVData against our three validity

[4] For $\mathrm{FP_{noc}}$, fluency improves by fixing grammatical errors. For $\mathrm{FP_{oc}}$, only edits that make the sentence clearer or more natural qualify are valid; simple synonym substitutions that do not measurably improve clarity or fluency are invalid (i.e., overcorrection).

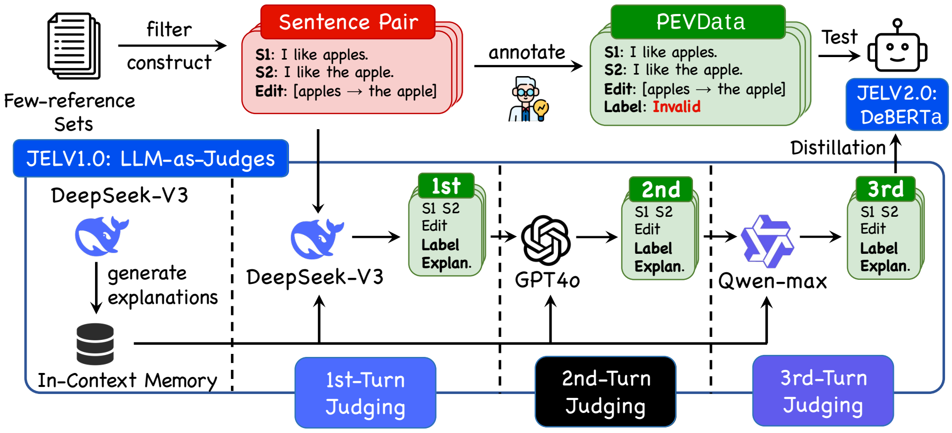

Figure 3: Overview of the JELVworkflow. Starting from few reference sets, we extract candidate sentence pairs and process them in two independent streams. One stream is manually annotated by experts to create the PEVData benchmark. The other is evaluated by a three turn LLM as Judges pipeline (JELV1.0) and the resulting labels are distilled into a DeBERTa classifier (JELV2.0).

LLM	Prec.	Rec.	$F_{0.5}$	Accuracy
DS-V3	0.6920	**0.9038**	0.7261	0.8415
GPT4o	0.8417	0.8349	0.8403	0.8976
Qwen	**0.8771**	0.8462	**0.8707**	**0.9134**

Table 1: Comparison of LLM-as-Judges JELV1.0 predictions and human annotations on the PEVDatabenchmark. **Prec.** and **Rec.** denote precision and recall, respectively. **Bold** indicates the **highest** score.

criteria, forming an *in-context memory*. We then include one valid example and one invalid example from this memory as two-shot demonstrations in the prompt, guiding the LLMs to apply our criteria more accurately.

Explain-then-Annotate Before giving judgments, the model generates a one-sentence explanation for its decision, ensuring transparency and consistency with our validity criteria.

Context-Aware Prompting We include the preceding and following sentences as additional context for more informed judgments.

Table 1 reports how well the LLM-as-Judges predictions align with human annotations on the PEVData benchmark. We use the $F_{0.5}$ metric, which weights precision twice as much as recall, to prioritize correct identification of *valid* edits for inclusion in the high quality reference set. The third-turn Qwen-max achieves the highest $F_{0.5}$ (0.8707) and Accuracy (0.9134), demonstrating the robustness of our LLM-as-Judges pipeline. Figure 5 presents an ablation study of JELV1.0 on the PEVData benchmark. Starting from the baseline of independent LLM judgments, we find that adding "explain-then-annotate", then in-context learning, and finally context-aware prompting each delivers a clear accuracy increase. Moreover, every subsequent LLM in the pipeline outperforms its predecessor, highlighting the strength of multi-turn optimization. When all three strategies are combined, overall accuracy exceeds 90%.

3.3.2 JELV2.0: Classifier Model Distillation

To reduce inference cost and enable large-scale deployment, we distill JELV1.0 into a DeBERTa classifier with under one billion parameters. Using our sentence pair construction approach (Section 3.2), we extract candidate edits from the eight extended CoNLL14 reference sets and construct 12,984 sentence pairs. After removing overlaps with PEVData to avoid data leakage, 12,416 unique pairs remain. JELV1.0 then labels these pairs, yielding 5,786 valid and 6,630 invalid examples, which form the training data. We train the DeBERTa classifier on this dataset, reserving half of PEVData for validation and half for testing.

We integrate two auxiliary features into DeBERTa's final embeddings: GPT-2 perplexity change and SBERT semantic similarity. These features are projected and concatenated through a lightweight classification head. To address class imbalance, we apply focal loss and weighted sampling. Robustness is further enhanced by FreeLB adversarial training (Zhu et al., 2019) and by maintaining an exponential moving average of model weights. Our training curriculum (Bengio et al., 2009) gradually unfreezes encoder layers and improves computational efficiency with mixed precision training, gradient accumulation, and a warmup

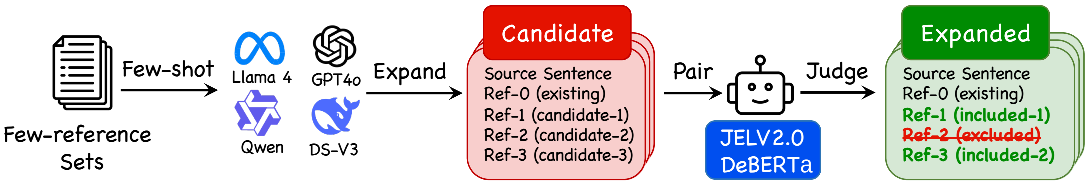

Figure 4: Automated expansion workflow for large scale few reference sets. Four LLMs generate candidate corrections using few shot prompts. Each candidate correction is paired with its source sentence following the Sentence Pair Construction protocol and evaluated by the distilled DeBERTa classifier (JELV2.0). Valid corrections are added to the reference set to produce the expanded corpus.

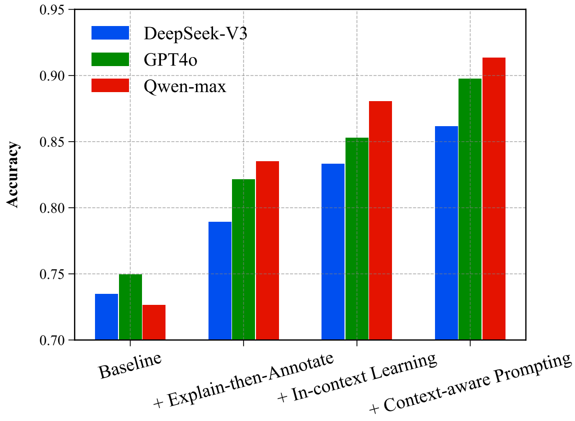

Figure 5: Ablation of JELV1.0 strategies on JEPV.

cosine learning rate schedule. The final model is selected using stratified k-fold cross validation with early stopping. On the test set, JELV2.0 achieves 85.25% precision, an $F_{0.5}$ score of 82.24%, and 78.91% accuracy, demonstrating high precision with low inference latency.

4 Reference Expansion and Retraining

Figure 4 shows our automated two-stage pipeline for expanding references: generation followed by filtering. We apply this pipeline to the BEA-2019 train and dev sets (Bryant et al., 2019), which comprise comprising 38,692 source sentences paired with a single human reference.

Generation To leverage LLMs' GEC expertise and reduce reliance on costly human annotation, we use four models (Llama 4, GPT-4o, Qwen, and DS-V3) to generate correction candidates. Each model is prompted with the source sentence and its single human reference as a one-shot example, and is briefed on our three edit-level validity criteria to guide outputs toward high-quality valid edits (see Appendix D.1 for the full prompt). On average, these models produce 8.74 (Llama 4), 6.82 (GPT-4o), 4.06 (Qwen) and 2.41 (DS-V3) candidates per sentence.

Filtering Each candidate edit is constructed into a sentence pair and evaluated by our distilled DeBERTa classifier (JELV2.0), and only edits judged valid are included in the final expanded reference set. JELV-based filtering reduces the average candidates per sentence from 11.54 to 3.90 on BEA-train set and from 14.21 to 4.08 on BEA-dev set (see Appendix C for full statistics). This demonstrates JELV's effectiveness in filtering diverse LLM-generated corrections into a concise, high-quality reference set.

Retraining To assess the impact of our expanded data, we retrain top GEC systems on the CoNLL14 benchmark and compare their performance to the original baselines trained on the BEA19 train and dev sets. Specifically, each system was retrained on three variants of our data: (1) the raw BEA19 train and dev sets (2) the full candidate set (expanded but not JELV-filtered) and (3) the fully expanded and JELV-filtered set. We evaluate the following eight GEC systems: Ensembles of best 7 models + GRECO + GTP-rerank, Majority-voting ensemble on best 7 models (Omelianchuk et al., 2024), GRECO (Qorib and Ng, 2023), GEC-DI (Zhou et al., 2023), Unsupervised GEC + cLang8 (Cao et al., 2023), MoECE (Qorib et al., 2024), SynGEC (Zhang et al., 2022b) and Sequence tagging + token-level transformations + two-stage fine-tuning (Omelianchuk et al., 2020).

Table 2 reports $F_{0.5}$ scores for the eight leading GEC systems trained on four different dataset variants and evaluated on CoNLL14. The "Raw(paper)" results, obtained by training on multiple publicly available datasets (including BEA19), slightly outperform our "Raw(our)" baseline which uses only BEA19. Nonetheless, most systems reach their highest $F_{0.5}$ when trained on the expanded and filtered BEA19 datasets, representing a measurable improvement over the baseline trained exclusively on the raw BEA19 data. Applying JELV-based filtering yields additional gains over unfiltered ex-

Dataset	Ensembles	Majority-voting	GRECO	GEC-DI	UGEC	MoECE	SynGEC	Sequence tagging
Raw (paper)	72.80	71.80	72.12	**69.60**	69.60	67.79	**67.60**	66.50
Raw (our)	72.68	71.44	70.99	69.43	69.38	67.45	67.29	66.31
Expanded w.o. Filtered	72.98	71.44	71.25	69.57	69.62	67.74	67.47	66.36
Expanded w. Filtered	**73.05**	**71.91**	**71.36**	69.58	**69.74**	**67.92**	67.57	**66.67**

Table 2: $F_{0.5}$ scores of eight GEC systems on CoNLL14 under four training configurations. Raw (paper) – published results using multiple publicly available datasets; Raw (our) – reproduction trained only on BEA19; Expanded w.o. Filtered – BEA19 plus all LLM-generated candidates; Expanded w. Filtered – BEA19 plus JELV-validated edits. **Bold** marks the highest score; underline marks the second highest.

pansion, highlighting the importance of validating candidate edits. Interestingly, even the unfiltered expansion improves over the raw baseline, indicating that simply increasing reference diversity without strictly quality examination can enhance model performance. A detailed analysis of this effect is left for future work.

5 Evaluation

5.1 Method

JELV-based Reclassification (Choshen and Abend, 2018b) shows that even short sentences admit over a thousand valid corrections on average, making exhaustive reference expansion infeasible and causing reference-based metrics to misclassify some valid edits as false positives inevitably. To mitigate this inherent bias, we apply the JELV to re-evaluate each edit initially marked as a FP. Edits judged valid by JELV are reclassified as true positives (TPs), reducing misclassification and improving evaluation reliability. After reclassifying edits, we apply CLEME-independent (Ye et al., 2023), a chunk-level multi-reference evaluation metric, to compute $F_{0.5}$ scores and this yields the JELV-based CLEME.

False Positive Decoupling FPs in GEC fall into two categories: non-overcorrections (FP_{noc}), which edit ungrammatical text, and overcorrections (FP_{oc}): which alter already correct text. Considering the two different edit properties that deserve different penalties, we apply a relative penalty weight α to FP_{oc} (*i.e.*, overcorrection). We then derive the *generalized precision*:

$$P_G = \frac{TP}{TP + FP} = \frac{TP}{TP + FP_{noc} + \alpha\, FP_{oc}}. \quad (1)$$

In experiments, we iterate α in the range of [0, 2] with step size of 0.01. Combining P_G with recall R, we further propose the *generalized F-score* as the edit-level metric:

$$F_{\beta-G} = (1 + \beta^2) \cdot \frac{P_G \cdot R}{\beta^2 \cdot P_G + R}. \quad (2)$$

Fluency Score Integration Prior study (Kobayashi et al., 2024b) recommends the combination of both edit-level metric and sentence-level metric in GEC evaluation. Therefore, we incorporate a sentence-level fluency measure based on GPT-2 perplexity (Ge et al., 2018):

$$f(x) = \frac{1}{1 + H(x)},\ H(x) = -\frac{\sum_{i=1}^{|x|} logP(x_i|x_{<i})}{|x|}, \quad (3)$$

where $P(x_i \mid x_{<i})$ is computed by GPT-2 in our experiments and $|x|$ denotes sentence length. Higher $f(x)$ indicates greater fluency (Yasunaga et al., 2021).

Comprehensive Metric Building on the complementary strengths of edit-level and sentence-level evaluations, we combine $F_{\beta\text{-}G}$ and the fluency score $f(x)$ via an interpolation weight γ:

$$F(x) = (1 - \gamma) \cdot F_{\beta\text{-}G} + \gamma \cdot f(x). \quad (4)$$

We iterate γ in the range of [0, 1] with step size of 0.01 in experiments. Figure 6 presents an overview of this comprehensive evaluation metric.

5.2 Settings

Meta-evaluation We evaluate our metrics on the SEEDA benchmark (Kobayashi et al., 2024b), which provides edit-level (SEEDA-E) and sentence-level (SEEDA-S) pairwise judgments for twelve GEC system corrections ("Base") and two additional fluent human corrections ("+Fluent corr.") on a subset of CoNLL14 test set[5]. For system-level evaluation, we compute the overall scores across all corrections for each system, derive a system ranking, and compare it to the human ranking produced by TrueSkill (Sakaguchi et al., 2014) using Pearson's r and Spearman's ρ. For sentence-level evaluation, we compute a score for each correction sentence, compare each pairwise decisions of

[5] In SEEDA, for correction pairs (A, B) sampled from these corrected sentence collections, three annotators provide 5-point scores for each granularity, resulting in 5347 pairwise judgments (A>B, A=B, A<B).

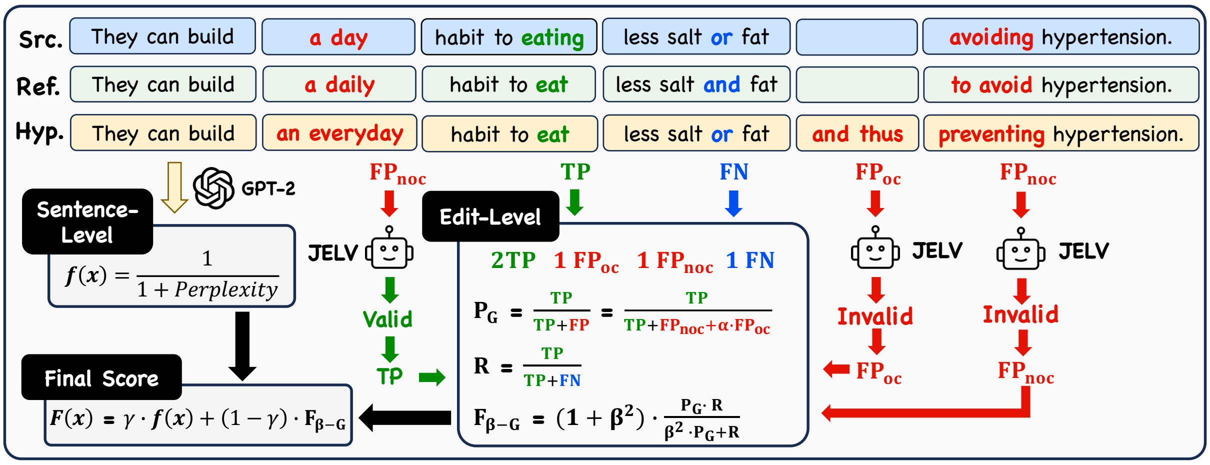

Figure 6: Overview of the comprehensive evaluation metric. Edits flagged as FPs are first reclassified via JELV2.0, then false positive decoupling distinguishes remaining FP_{noc} and FP_{oc} to compute the edit-level generalized F-score $\text{F}_{\beta\text{-G}}$. In parallel, GPT-2 perplexity produces the sentence-level fluency score $f(x)$ for the hypothesis. A interpolation weight γ combines these two streams into the final metric $F(x)$.

Metric	System-level								Sentence-level							
	SEEDA-E				SEEDA-S				SEEDA-E				SEEDA-S			
	Base		+ Fluent corr.		Base		+ Fluent corr.		Base		+ Fluent corr.		Base		+ Fluent corr.	
	r	ρ	r	ρ	r	ρ	r	ρ	Acc	τ	Acc	τ	Acc	τ	Acc	τ
M^2	0.791	0.764	-0.239	0.161	0.658	0.487	-0.336	-0.013	0.582	0.328	0.527	0.216	0.512	0.200	0.496	0.170
ERRANT	0.697	0.671	-0.502	0.051	0.557	0.406	-0.587	-0.116	0.573	0.310	0.511	0.188	0.498	0.189	0.471	0.129
GoToScorer	0.901	0.937	0.667	0.916	0.929	0.881	0.627	0.881	0.521	0.042	0.505	0.009	0.477	-0.046	0.504	0.009
PT-M^2	0.896	0.909	-0.083	0.442	0.845	0.769	-0.162	0.336	0.587	0.293	0.542	0.200	0.527	0.204	0.528	0.180
CLEME	0.721	0.699	-0.465	0.121	0.614	0.462	-0.545	-0.029	0.621	0.243	0.562	0.123	0.606	0.212	0.551	0.103
GLEU	0.911	0.897	0.053	0.482	0.847	0.886	-0.039	0.475	0.695	0.404	0.630	0.266	0.673	0.351	0.611	0.227
Scribendi Score	0.830	0.848	0.721	0.847	0.631	0.641	0.611	0.717	0.377	-0.196	0.359	-0.240	0.354	-0.238	0.345	-0.264
SOME	0.901	0.951	0.943	0.969	0.892	0.867	0.931	0.916	0.747	0.512	0.743	0.494	0.768	0.555	0.760	0.531
IMPARA	0.889	0.944	0.935	0.965	0.911	0.874	0.932	0.921	0.742	0.502	0.725	0.455	0.761	0.540	0.742	0.496
Qwen	0.965	0.979	0.939	0.987	0.923	0.909	0.979	0.943	0.779	0.558	0.768	0.536	0.795	0.591	0.780	0.560
DeepSeek-V3	0.953	0.986	0.960	0.991	0.910	0.902	0.981	0.938	0.795	0.589	0.762	0.524	0.797	0.593	0.762	0.523
GPT4o	0.976	0.986	0.958	0.987	0.913	0.874	0.952	0.916	0.786	0.572	0.766	0.532	0.807	0.615	0.785	0.569
Average	0.965	0.984	0.952	0.988	0.915	**0.895**	**0.971**	**0.932**	0.787	0.573	0.765	0.531	0.800	0.600	0.776	0.551
JELV-based CLEME	0.960	0.965	0.473	0.802	0.905	0.853	0.395	0.714	0.773	0.546	0.729	0.458	0.773	0.547	0.724	0.449
JELV-based $\mathbf{F}(\mathbf{x})$	**0.968**	**0.986**	**0.962**	**0.991**	**0.928**	0.860	0.923	0.908	**0.815**	**0.631**	**0.796**	**0.593**	**0.812**	**0.623**	**0.781**	**0.562**

Table 3: Results of system- and sentence-level meta-evaluations on the SEEDA dataset. JELV-based CLEME first reclassifies FPs and then computes the chunk-level $\text{F}_{0.5}$ score using CLEME-independent. JELV-based $\mathbf{F}(\mathbf{x})$ further decouples false positives and integrates fluency score, with each reported correlation maximized over the (α, γ) grid. **Boldface** indicates the **highest** correlation score in each column.

metric to SEEDA's human judgments, and report classification accuracy and Kendall's τ.

Considered Metrics We adopt three categories of evaluation metrics to capture different levels of GEC evaluation. Edit-based metrics (EBMs) comprise five measures: M^2 (Dahlmeier and Ng, 2012), ERRANT (Bryant et al., 2017), GoToScore (Gotou et al., 2020), PT-M^2 (Gong et al., 2022) and CLEME (Ye et al., 2023). Sentence-based metrics (SBMs) include GLEU (Napoles et al., 2015), Scribendi Score (Islam and Magnani, 2021), SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022). For LLM-based metrics we use DeepSeek-V3, Qwen-max and GPT4o to assign each sentence an integer score from one to five. We follow the prompt design and experimental setup of the "GPT4-S + Fluency" configuration (Kobayashi et al., 2024a) and report the average correlation across the three models to mitigate individual model bias.

5.3 Result and Analysis

Table 3 presents our system-level and sentence-level meta-evaluation on the SEEDA dataset. By applying JELV reclassification, JELV-based CLEME achieves substantial gains over the original CLEME and other edit-based metrics, particularly at the sentence level. When we further decouple remaining FPs and integrate the fluency score via our comprehensive metric F(x) (Eq. 4), the correlations improve even more, achieving state-of-the-art (SOTA) correlations in most levels.

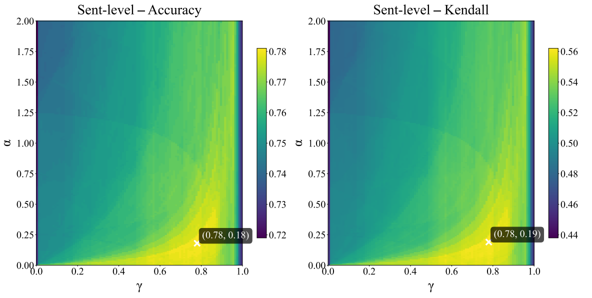

Figure 7: Heat map of sentence-level SEEDA-S correlations across all values of the penalty weight α and the interpolation weight γ. The marked point shows global maximum correlation and corresponding α and γ.

Metric	**SEEDA-E**				**SEEDA-S**			
	Base		+ Fluent corr.		Base		+ Fluent corr.	
	r	ρ	r	ρ	r	ρ	r	ρ
Gene. Max	0.959	0.986	0.974	0.991	0.887	0.879	0.947	0.924
Gene.2Soc.	0.972	0.965	0.974	0.987	0.931	0.853	0.947	0.912
Soc. Max	0.978	0.986	0.974	0.991	0.939	0.916	0.947	0.943
Soc.2Gene.	0.955	0.986	0.974	0.987	0.879	0.825	0.947	0.877

Table 4: Cross-validation of optimized metric parameters across SEEDA subsets. “Gene. Max” and “Soc. Max” denote parameters (α, γ) tuned and evaluated within the genetic-testing and social-media domains, respectively. “Gene. 2Soc.” indicates parameters tuned on genetic-testing and applied to social-media; “Soc. 2Gene.” indicates the reverse. Pearson (r) and Spearman (ρ) correlations with human judgments at the system level are reported for both SEEDA-E and SEEDA-S.

Effect of weight α and γ To explore how the penalty weight α (from Eq. 1) and the interpolation weight γ (from Eq. 4) influence correlations, we evaluated sentence-level SEEDA-S correlations for every (α, γ) combination and visualized the results in Figure 7. The highlighted optimal point corresponds to a relatively low α, indicating that under-penalizing overcorrections ($\mathrm{FP}_{\mathrm{oc}}$) compared to non-overcorrections ($\mathrm{FP}_{\mathrm{noc}}$) better aligns with human judgments. The best γ exceeds 0.5, showing that the fluency component $f(x)$ contributes more than the edit-level generalized F-score $\mathrm{F}_{\beta\text{-G}}$. Complete heat maps for each evaluation level are provided in Appendix E.

Scalability SEEDA’s human-rated corrections are split into two domains: genetic testing and social media. To test our metric’s scalability, we first identified the optimal α and γ on one domain via a global search, then applied those parameters to the other domain and measured the resulting Pearson and Spearman correlations in system level. Table 4 presents these cross-validation results. Parameters tuned on one subset maintain high correlation on the other, demonstrating that our metric generalizes across different types of corrections while preserving strong alignment with human evaluations.

6 Conclusions

In this paper, we introduce Judge of Edit-Level Validity (JELV), a framework for automated edit validity assessment in GEC. JELV offers two implementations: JELV1.0, a multi-turn LLM-as-judges pipeline, and JELV2.0, a distilled DeBERTa classifier. Both achieve strong agreement with human annotations on our PEVData. We demonstrate two applications that mitigate reference scarcity. First, a JELV-based generation-then-filtering pipeline automates reference expansion and yields clear improvements in GEC model performance. Second, our evaluation metric applies JELV to reclassify false positives and then incorporates false positive decoupling together with fluency scoring, resulting in state-of-the-art correlation with human judgments. JELV therefore provides a scalable solution to enrich reference diversity and enhance both evaluation reliability and model robustness in GEC.

7 Limitations

There are two possible limitations in this work. First, although JELVdemonstrates strong agreement with human annotations, it does not achieve perfect accuracy in edit-level validity assessment, leaving room for further improvement. Second, we have not yet analyzed in detail why automated reference expansion boosts model performance. In particular, we have not examined the mechanisms by which adding LLM-generated candidate edits without JELV-based filtering yields performance benefits, nor quantified how the number of added references correlates with accuracy gains. We will address these questions in future work.

8 Ethics Statement

Our study builds on publicly available GEC datasets. Three expert annotators with backgrounds in English teaching, proofreading, and linguistics manually judged edit validity and were compensated at market rates. We thank them for their contributions. To our knowledge, there are no ethical concerns arising from the use of these data or annotations.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

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

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

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

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

Hannan Cao, Liping Yuan, Yuchen Zhang, and Hwee Tou Ng. 2023. Unsupervised grammatical error correction rivaling supervised methods. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3072–3088.

Shamil Chollampatt and Hwee Tou Ng. 2018. A reassessment of reference-based grammatical error correction metrics. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2730–2741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022. On the limitations of reference-free evaluations of generated text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10960–10977, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Frank Palma Gomez and Alla Rozovskaya. 2024. Multi-reference benchmarks for russian grammatical error correction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1253–1270.

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

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095, Barcelona, Spain (Online). International Committee on 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, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

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.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Yumeng Liu, Zhenghua Li, Haochen Jiang, Bo Zhang, Chen Li, and Ji Zhang. 2024. Towards better utilization of multi-reference training data for chinese grammatical error correction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3044–3052.

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.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There‘s no comparison: Referenceless evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2115, Austin, Texas. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

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

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

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. *arXiv preprint arXiv:2404.14914*.

Muhammad Reza Qorib, Alham Fikri Aji, and Hwee Tou Ng. 2024. Efficient and interpretable grammatical error correction with mixture of experts. *arXiv preprint arXiv:2410.23507*.

Muhammad Reza Qorib and Hwee Tou Ng. 2023. System combination via quality estimation for grammatical error correction. *arXiv preprint arXiv:2310.14947*.

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

Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms. *arXiv preprint arXiv:2406.07791*.

Jinxiang Xie, Yilin Li, Xunjian Yin, and Xiaojun Wan. 2024. Dsgram: Dynamic weighting sub-metrics for grammatical error correction in the era of large language models. *arXiv preprint arXiv:2412.12832*.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Jingheng Ye, Zishan Xu, Yinghui Li, Xuxin Cheng, Linlin Song, Qingyu Zhou, Hai-Tao Zheng, Ying Shen, and Xin Su. 2024. Cleme2. 0: Towards more interpretable evaluation by disentangling edits for grammatical error correction. *arXiv preprint arXiv:2407.00934*.

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.

Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578, Vancouver, Canada. Association for Computational Linguistics.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022a. Mucgec: a multi-reference multi-source evaluation dataset for chinese grammatical error correction. *arXiv preprint arXiv:2204.10994*.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. Syngec: Syntax-enhanced grammatical error correction with a tailored gec-oriented parser. *arXiv preprint arXiv:2210.12484*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving seq2seq grammatical error correction via decoding interventions. *arXiv preprint arXiv:2310.14534*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

Appendix

A Detailed Related Work

A.1 GEC Evaluation Metrics

A core component of a GEC system is the ability to measure model performance. **Reference-based metrics**, the traditional paradigm, rely on alignment with human-crafted references. Early approaches like the M^2 scorer used edit-based $F_{0.5}$ scoring but faced criticism for artificially inflated precision (Bryant et al., 2017), prompting linguistically grounded alternatives like *ERRANT* (Bryant et al., 2017) for improved edit alignment. Despite their effectiveness, reference-based metrics often suffer from issues such as overfitting to reference-specific patterns and inevitably penalizing valid but non-reference corrections, which introduces inherent biases (Choshen and Abend, 2018b). To circumvent reliance on references, **reference-less metrics** emerged, leveraging quality estimation frameworks: Grammaticality-Based Metrics (Napoles et al., 2016) combined fluency and meaning preservation scores, later extended by SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022) with BERT-based models and Scribendi Score (Islam and Magnani, 2021) using perplexity-edit hybrids. However, these metrics, while bypassing the need for references, can lack interpretability and may inherit biases from the models they employ (Deutsch et al., 2022), and struggle to detect issues like over-correction. **The advent of LLMs** introduced a paradigm shift, with GPT-4 (Kobayashi et al., 2024a) and Prompt Engineering (Xie et al., 2024) enabling direct numerical scoring via natural language criteria interpretation, aiming to mimic human-like evaluation and potentially capturing more complex aspects of correction quality. Despite their promise, LLM-based metrics face challenges in explainability (acting as ”black-box” evaluators), instability, multiple biases (Li et al., 2024), and prohibitive computational costs.

A.2 Meta-evaluation

Meta-evaluation frameworks determine metric validity: human correlation. This involves evaluating metrics themselves by correlating their scores with human judgments based on two dominant datasets GJG15 (Grundkiewicz et al., 2015) and SEEDA (Kobayashi et al., 2024b), providing a gold standard for determining the true effectiveness of GEC evaluation methodologies and guiding the development of superior metrics. However, GJG15 was later found to be problematic, and many of the conclusions drawn using these datasets were called into question(Choshen and Abend, 2018a; Chollampatt and Ng, 2018). Additionally, GJG15 may yield different results depending on granularity, such as sentence-level or edit-level evaluations(Kobayashi et al., 2024b). SEEDA, on the other hand, performs human evaluations based on two different granularities: SEEDA-E for edit-based evaluation and SEEDA-S for sentence-based evaluation, making it more reliable.

A.3 LLM-based Evaluation

Prior work (Kobayashi et al., 2024a) has observed the huge potential for LLM as a scorer in GEC task and get quite high correlations with human judgments. However, we find that utilizing LLM (e.g. GPT-4) as a scorer has these following shortcomings.

Lack of interpretability Acting as ”black-box” evaluators, the results are coarse-granularity, unexplainable and may have potential subjective bias.

Position-Related Bias and Instability Recent studies (Shi et al., 2024; Zheng et al., 2023) have examined position bias in the LLMs-as-judges context. To assess GPT-4’s sensitivity to input order, we evaluated the same set of five candidate corrections (numbered 1–5) across twenty randomly generated permutations. Using a standardized prompt template[6] that differed only in the sequence of corrections, we collected quality ratings on a five-point scale. The prompt template is as follows.

```
The goal of this task is to rank the presented
targets based on the quality of the sentences.

The context consists of three sentences from an
essay written by an English learner.

After reading the context to understand the
flow, 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).

Please evaluate each target with a focus on the
fluency of the sentence.

# Context Source:
[PREVIOUS] So, they have to also prepare
mentally .
[SOURCE] Secondly, genetic diseases costs
highly for the treatment and medication
[FOLLOWING] Albinism is one of the examples.

# Targets
[CORRECTION 1] ”Secondly , genetic disease cost
higher for the treatment and medication .”
[CORRECTION 2] ”Secondly , genetic diseases
cost highly for treatment and medication .”
[CORRECTION 3] ”Secondly , genetic diseases
cost highly for the treatment and medication .”
[CORRECTION 4] ”Secondly , genetic diseases
cost high for the treatment and medication .”
[CORRECTION 5] ”Secondly , genetic diseases
costs highly for treatment and medication .”

# output format
The output should be a markdown code snippet
formatted in the following schema

{{ "target1_score": int // assigned score for
target 1
 ...
"target5_score": int // assigned score for
target 5 }}
```

Figure 8 presents the score distribution of each correction: the violins’ narrow, vertically extended profiles indicate high variability and instability in ratings and reveal position-related biases caused by different permutations of corrections. In particular, we also find that corrections placed in the first or last positions tend to receive higher scores than those in intermediate slots, which is align with prior conclusions (Li et al., 2024).

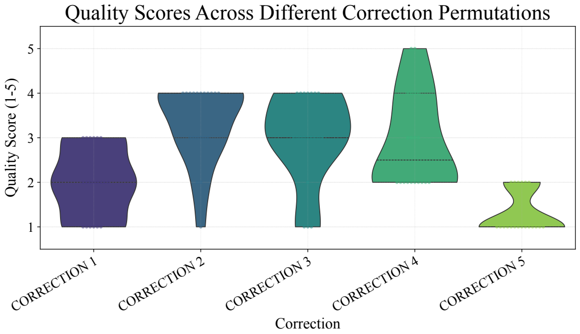

Figure 8: Score distributions for five corrections over 20 random input orderings. Narrow, elongated violins indicate unstable scoring and reflect position-related biases from different permutations of corrections.

In terms of GEC evaluation metric, our JELV-based metric $\mathbf{F}(\mathbf{x})$ has advantages of stability, high-correlation and explainability compared to LLM-based evaluation, which is more suitable for a standardized GEC metric for benchmarking.

B Dataset Collection

We sample single-edit pairs from three public datasets:

[6] This prompt template is from the “GPT4-S + Fluency” configuration from (Kobayashi et al., 2024a) and also used to generate the evaluation metric results in Section 5.2.

- **CoNLL14.** (Ng et al., 2014; Bryant and Ng, 2015) the official test set provides two references per sentence, with an eight-reference extension offering additional corrections. Noting that some of these extended edits are not so reliable, we extract 1,118 edits unique to this extension for our expert annotation on validity.
- **ArgRewrite.** (Zhang et al., 2017) Seven prior annotators labeled each revision as “Better” or “NotBetter”. However, “Better” does not guarantee that an edit satisfies all three of our validity criteria. We therefore select 844 single-edit revisions with at least five “Better” votes for our expert validity re-annotation.
- **JFLEG.** (Napoles et al., 2017) Designed for *fluency* evaluation, this dataset provides four references per source sentence. We extract 835 single-edit references, pair each with its source sentence, and treat them as valid since they originate from existing references.

C Statistics about JELV-based Filtering

#Ref. / Sent.	**BEA-Train**		**BEA-Dev**	
	Pre-J	Post-J	Pre-J	Post-J
Mean	11.54	3.90	14.21	4.08
S.D.	4.74	3.23	6.47	3.63
Max	104	40	78	35

Table 5: Number of references per sentence in BEA-Train and BEA-Dev before (Pre-J) and after (Post-J) JELV2.0 filtering. Post-J values reflect only edits judged valid, yielding lower averages and reduced variability.

D Complete Prompts

D.1 LLM-as-Judges Pipeline (JELV1.0)

1st Turn: DeepSeek-V3

You are a linguist tasked with judging whether a proposed sentence edit is VALID or INVALID.
VALID = 1, INVALID = 0.

A VALID edit must satisfy THREE CRITERIA:
1. GRAMMATICALITY: the resulting SENTENCE is free of grammatical errors.
2. FAITHFULNESS: the edit preserves the INTENDED meaning you infer from the source sentence and its context; acceptable modifications that IMPROVE clarity or expression are allowed as long as the INTENDED meaning is effectively conveyed.
3. FLUENCY: Does the hypothesis edit IMPROVE the fluency of the sentence?
The improvement in fluency can result from enhancements in grammaticality, naturalness, or readability.
ONLY when ALL of the three criteria are satisfied can the edit be judged as valid.

NOTE: Simple synonym substitutions that do NOT measurably IMPROVE clarity or fluency are INVALID. Refer to the In-Context Examples provided above when making your judgment.

Only output a one-sentence analysis prefixed with ”Analysis”:, then on a new line ”Final Judgment: [0/1]”.

2nd and 3rd Turn: GPT4o and Qwen

You are an expert language model specializing in evaluating edit judgments within context.
You will receive a JSON object containing:
’src’: the original source sentence. ’edit_noc’: a description of the hypothesis edit. ’hypo_noc’: the sentence after applying the hypothesis edit. ’llm_analysis’: a one-sentence analysis of the edit. ’llm_prediction’: a binary prediction from the previous model (1 = valid, 0 = invalid).

Use the following GOLD CRITERIA to decide if the edit is VALID (1) or INVALID (0):

1. GRAMMATICALITY: the resulting sentence MUST be grammatically correct.
2. FAITHFULNESS: the edit MUST preserve the INTENDED meaning you infer from the source and its context; acceptable modifications that IMPROVE clarity or expression are allowed as long as the INTENDED meaning is effectively conveyed.
3. FLUENCY: Does the hypothesis edit IMPROVE the fluency of the sentence? The improvement in fluency can result from enhancements in grammaticality, naturalness, or readability.
NOTE: Simple synonym substitutions that do NOT contribute to clarity or fluency improvement are NOT considered valid.

ONLY when ALL THREE CRITERIA are fully satisfied should the edit be considered VALID (1).

Your task:
a) Verify that ’llm_analysis’ correctly assesses grammaticality, faithfulness, AND fluency. b) Ensure that ’llm_prediction’ (0/1) matches that assessment.
If you find any discrepancy, output a revised ’llm_analysis’ and adjust the ’llm_prediction’.
Respond with a JSON object containing ONLY two keys:
’llm_analysis’: the corrected one-sentence analysis, ’llm_prediction’: the corrected binary judgment.

D.2 Candidate Correction Generation

You are tasked with correcting an English sentence using a reference example. The format is as follows:
Original: original_sentence Reference: reference_sentence

Your goal is to generate ALL POSSIBLE corrections for the original sentence based on the reference sentence, without being restricted by the specific style, modification approach, or pattern of the reference. Please generate corrections that could improve the sentence, ensuring they follow the **GOLD CRITERIA** outlined below.

ONLY generate valid edits, meaning:
- **GRAMMATICALITY**: The correction MUST be free of grammatical errors.
- **FAITHFULNESS**: The correction MUST PRESERVE the **INTENDED meaning** of the original sentence. Acceptable modifications that improve clarity or expression are allowed, as long as the intended meaning is preserved.
- **FLUENCY**: The correction MUST IMPROVE the **naturalness** and **readability** of the sentence.

If no valid modifications are possible beyond the reference, output ”ONLY one reference!”. Format the corrections as follows:
[correction 1] ...
[correction 2] ...
[correction N] ...

E Heat Maps

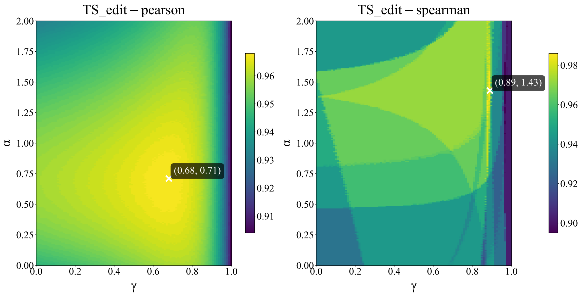

Figure 9: Heat map of system-level SEEDA-E base.

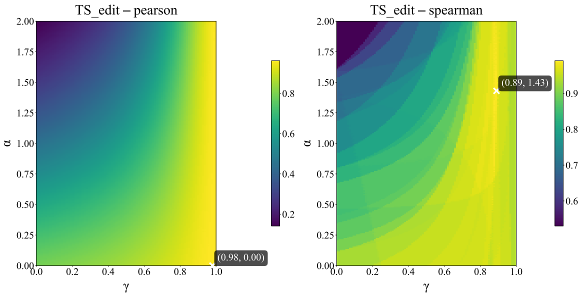

Figure 10: Heat map of system-level SEEDA-E +Fluent corr.

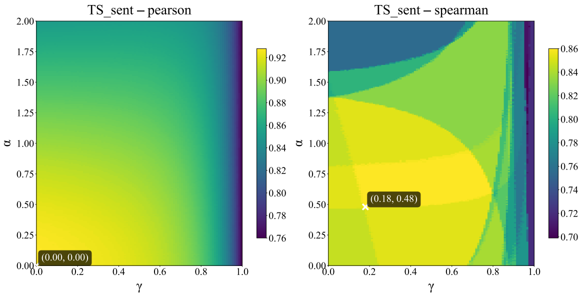

Figure 11: Heat map of system-level SEEDA-S base.

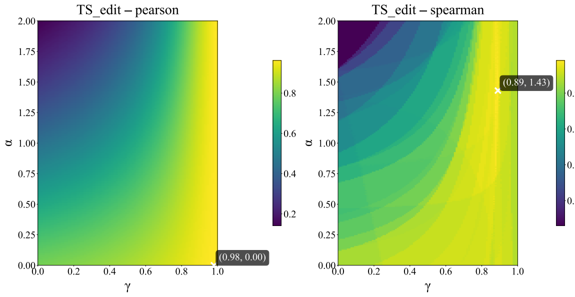

Figure 12: Heat map of system-level SEEDA-S +Fluent corr.

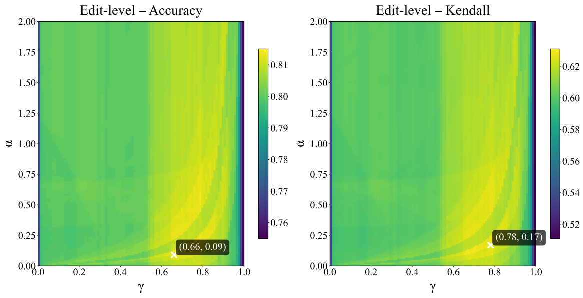

Figure 13: Heat map of sentence-level SEEDA-E base.

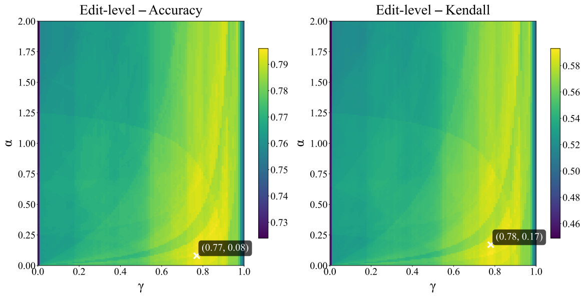

Figure 14: Heat map of sentence-level SEEDA-E +Fluent corr.

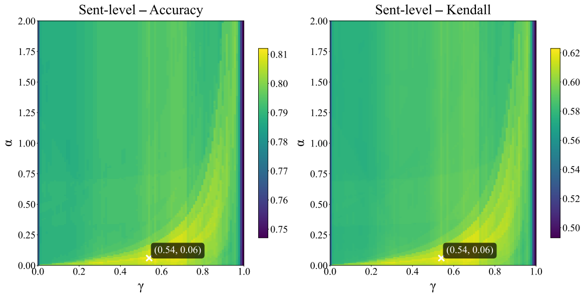

Figure 15: Heat map of sentence-level SEEDA-S base.

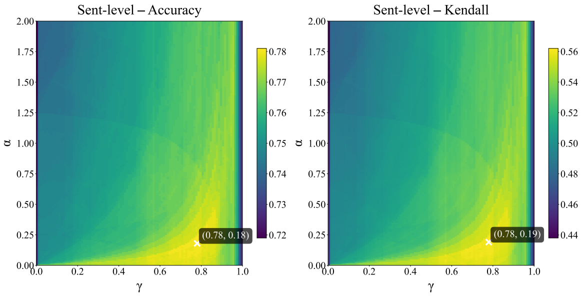

Figure 16: Heat map of sentence-level SEEDA-S +Fluent corr.