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

Anonymous submission

Abstract

Existing Grammatical Error Correction (GEC) systems suffer from limited reference diversity, leading to underestimated evaluation and restricted model generalization. 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 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. We also apply JELV to filter LLM-generated correction candidates, expanding the BEA19's single-reference dataset containing 38,692 source sentences. Retraining top GEC systems on this expanded dataset yields measurable performance gains. JELV provides a scalable solution for enhancing reference diversity and strengthening both evaluation and model generalization.

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.

Background 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) *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). (2) *Limited model performance*: training on a narrow reference set constrains the model to specific correction patterns, leading to poor generalization. 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. This manual approach hinders scalability, restricts the diversity of valid corrections, and perpetuates inherent biases in reference-based evaluation (Choshen and Abend 2018b). Thus, an automated approach for expanding diverse and valid corrections is crucial to advance research and development in GEC.

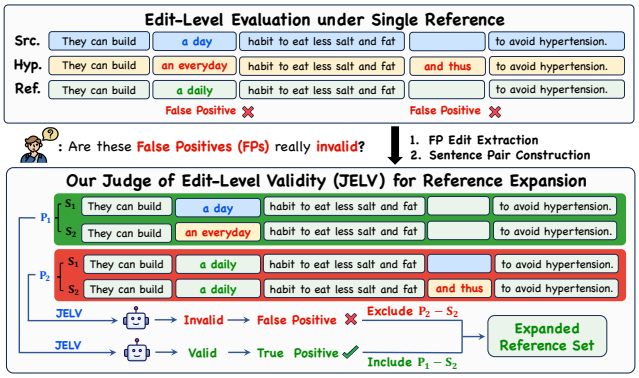

Figure 1: Comparison between biased edit-level evaluation using a single reference and our automated expansion of valid edits via the Judge of Edit-Level Validity (JELV). **Src.**, **Hyp.** and **Ref.** denote the source, hypothesis and reference sentences, respectively. We form two sentence pairs, $\mathbf{P_1}$ 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, 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 golden criteria: (1) grammatical correctness,

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

(2) meaning preservation, and (3) fluency improvement.

Present work To provide a gold-standard benchmark for edit-level validity, we introduce the Pair-wise Edit-level Validity Dataset (PEVData). To collect valid edits that are often overlooked, we notice that many actually *valid* hypothesis edits[2] are misclassified as *invalid*—False Positive (FP)—in reference-based evaluation due to incomplete reference coverage, as shown in Figure 1. Therefore, we extract every hypothesis sentence containing edits initially classified as FPs and pair it with its source sentence. We then align all other tokens to the reference so that only single 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, Wang, and Wang 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. With JELV's high-precision edit judgments, valid corrections are automatically identified and integrated into the reference set. The complete automated reference expansion workflow is shown in Figure 1.

We apply JELV to mitigate reference scarcity in two complementary ways. (1) **Implicit** reference expansion for *underestimated evaluation*. It is infeasible to eliminate misclassified FPs through exhaustive explicit reference expansion since even short sentences have over 1000 valid corrections on average (Choshen and Abend 2018b). Instead, we introduce JELV into evaluation pipeline to reclassify those misclassified FPs as true positives (TPs)—achieving the equivalent of exhaustive reference enumeration at minimal cost. 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. (2) **Explicit** reference expansion for *limited model performance*. We employ a generation-then-filtering pipeline to automate explicit reference expansion. 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 top 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.

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

[2]candidate correction produced by the GEC system

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

2 Related Work

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

GEC Evaluation Automatic evaluation metrics for GEC are divided into *reference-based* and *reference-less* approaches (Maeda, Kaneko, and Okazaki 2022). **Reference-based metrics** score outputs by comparing them to human references. These methods can penalize valid corrections that are not in the reference set, reflecting biases in limited reference coverage (Choshen and Abend 2018b). **Reference-less metrics** (Yoshimura et al. 2020; Maeda, Kaneko, and Okazaki 2022) assess correction quality without references using statistical and language models, but may lack transparency and inherit biases from their underlying models (Deutsch, Dror, and Roth 2022). **LLM-based metrics** (Kobayashi, Mita, and Komachi 2024a; Xie et al. 2024) use LLMs as scorers to evaluate corrections, offering strong human correlation at the expense of higher computational cost and potential instability. **Meta-evaluation** benchmarks such as GJG15 (Grundkiewicz, Junczys-Dowmunt, and Gillian 2015) and SEEDA (Kobayashi, Mita, and Komachi 2024b) measure the correlation between GEC metrics and human judgments, providing the standard for validating GEC evaluation methods. 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). 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.

3 Validity Judgment

This section establishes our framework for automated edit-level validity judgment. We first define clear criteria for valid edits, then introduce our human-annotated benchmark dataset PEVData, and finally present JELV 1.0 and 2.0.

3.1 Criteria for Validity

We begin by analyzing edits (*i.e.*, *source*→ *target*) that are truly **valid** but misclassified as FPs due to incomplete 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.

For each reference, under the assumption that all text outside the edit span is correct, we define three criteria for

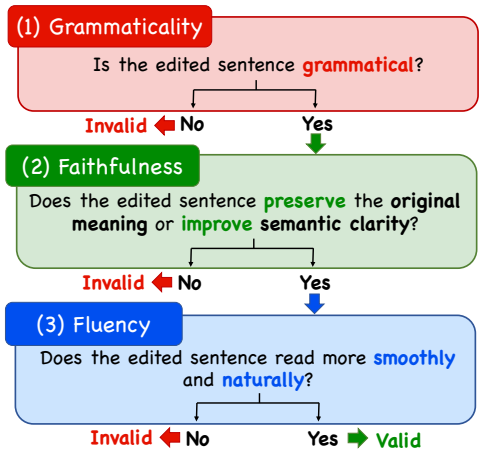

Figure 2: Three criteria for judging edit-level validity.

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. 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[4] (Ye et al. 2023), we extract each hypothesis sentence containing edits initially judged as FPs and pair it with its source. All other tokens are realigned to the reference so that only single edit span differs. 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: 1,118 from *CoNLL14* (Ng et al. 2014), 844 from *ArgRewrite* (Zhang et al. 2017), and 835 from *JFLEG* (Napoles, Sakaguchi, and Tetreault 2017). We provide more details of these datasets in Appendix B.

Annotation Protocol All sentence pairs were judged for edit validity by annotators with backgrounds in English teaching, proofreading, or linguistics; each pair is presented with its preceding and following sentences for context. Since *CoNLL14* pairs have no prior validity labels, three annotators apply our three criteria in a three-stage process: (1) independent labeling with only unanimous decisions retained, (2) a second independent pass on the remaining pairs, again requiring unanimity, and (3) a joint discussion to resolve any conflicts by consensus or majority vote. By contrast, *ArgRewrite* pairs already carried "Better/NotBetter" judgments from seven prior annotators, so a single expert performed a final validity check, and *JFLEG* pairs—drawn directly from the reference set—were accepted as valid without further annotation. In total, PEVData comprises 2,797 sentence pairs (1,459 valid, 1,338 invalid), yielding a balanced benchmark for edit-level validity.

[4]This assumption allows us to isolate individual edits for analysis: even when multiple grammatical errors exist in a sentence, we can focus on one edit by aligning all other tokens to the reference.

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 PEVData benchmark. **Prec.** and **Rec.** denote precision and recall, respectively. **Bold** indicates the **highest** score.

3.3 JELV: Developing an Automated Judge

While PEVData provides manually annotated judgments on edit-level validity, automatic evaluation at scale is essential. Therefore, we propose the Judge of Edit-Level Validity (JELV) and demonstrate its effectiveness on PEVData benchmark. Besides, to balance accuracy and efficiency, JELV comprises two implementations: *JELV1.0*, an LLM-as-judges pipeline for high accuracy, and *JELV2.0*, a distilled DeBERTa classifier for fast inference. Figure 3 shows the JELV workflow overview.

JELV1.0: LLM-as-Judges Pipeline

- **Multi-Turn Optimization.** JELV1.0 sequentially applies DeepSeek-V3 (DS-V3), GPT-4o, and Qwen-Max. Each model reviews and refines the previous model's explanations and judgments, correcting errors through iterative calibration. We use Qwen-Max's outputs as the final validity labels. See Appendix G.1 for prompt details.
- **In-Context Learning (ICL).** To increase JELV1.0's judgment accuracy, we help it understand our criteria through demonstrations. We use DS-V3 to generate explanations for 50 edits sampled from PEVData[5] (excluded from the test set) against our three validity criteria, creating an *in-context memory*. For each judgment, we include one valid and one invalid example as two-shot demonstrations.
- **Explain-then-Annotate.** Before giving a judgment, the model generates a one-sentence explanation for its decision to ensure transparency and consistency with our validity criteria.
- **Context-Aware Prompting.** We include the preceding and following sentences of the sentence being judged

Table 1 reports how well the LLM-as-Judges predictions align with human annotations on PEVData benchmark. Higher values indicate better alignment with human

[5]We limit to 50 edits (25 valid, 25 invalid) to preserve test set integrity while ensuring balanced representation.

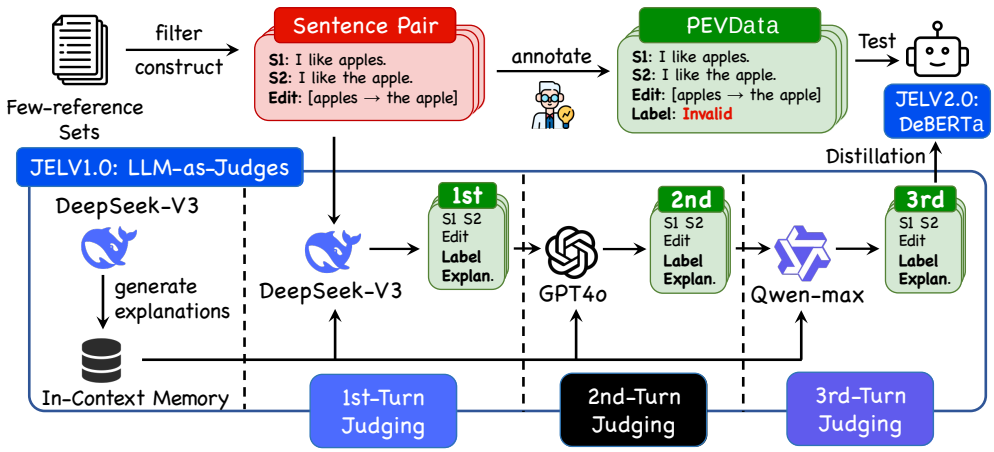

Figure 3: Overview of the JELV workflow. Starting from few reference sets, we extract candidate sentence pairs and process them in two independent streams. One stream is manually annotated by experts to create the PEVData. The other is evaluated by a three turn LLM as Judges pipeline (JELV1.0) and the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

judgments. 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 effectiveness and strength of our LLM-as-Judges pipeline. Figure 4 presents an ablation study evaluating the contribution of each component in JELV1.0 on the PEVData benchmark. Starting from the baseline of independent LLM judgments, we find that adding "explain-then-annotate", then ICL, 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%.

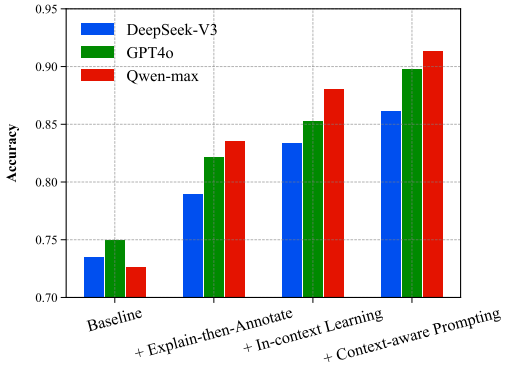

Figure 4: Ablation of JELV1.0 strategies on PEVData.

JELV2.0: Classifier Model Distillation While JELV1.0 achieves high alignment with human judgments, it incurs high inference costs that limit large-scale deployment. To address this, we distill JELV1.0 into a lightweight DeBERTa classifier (<1B parameters). Using our sentence pair construction approach (Section 3.2), we extract candidate edits from the extended CoNLL14 reference sets (Bryant and Ng 2015), yielding 12,984 sentence pairs. After removing overlaps with PEVData to avoid data leakage, 12,416 pairs remain. We then apply JELV1.0 to label these pairs, producing 5,786 valid and 6,630 invalid examples as training data. We train the DeBERTa classifier on this dataset, reserving half of PEVData for validation and half for testing.

To enhance JELV2.0's judgment accuracy, we integrate two auxiliary features that capture complementary aspects of sentence pairs: GPT-2 probability (indicating grammaticality and fluency (Yasunaga, Leskovec, and Liang 2021)) and SBERT semantic similarity (reflecting meaning preservation)[6]. These features are projected and concatenated with DeBERTa's final embeddings through a lightweight classification head to avoid overwhelming the primary text representations. To address class imbalance, we apply focal loss and weighted sampling (Lin et al. 2017). We enhance robustness through FreeLB adversarial training (Zhu et al. 2019) and exponential moving average of model weights to stabilize training dynamics. Our training employs curriculum learning (Bengio et al. 2009) with gradual layer unfreezing to prevent overfitting and 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 its high precision with low inference latency.

4 Evaluation

This section presents how we integrate JELV2.0 into GEC evaluation pipelines to derive a comprehensive metric with additional evaluation strategies, and demonstrate its effectiveness through meta-evaluation.

4.1 Method

JELV-based Reclassification (Choshen and Abend 2018b) shows that even short sentences have over a

[6]See ablation study on the two features in Appendix C.

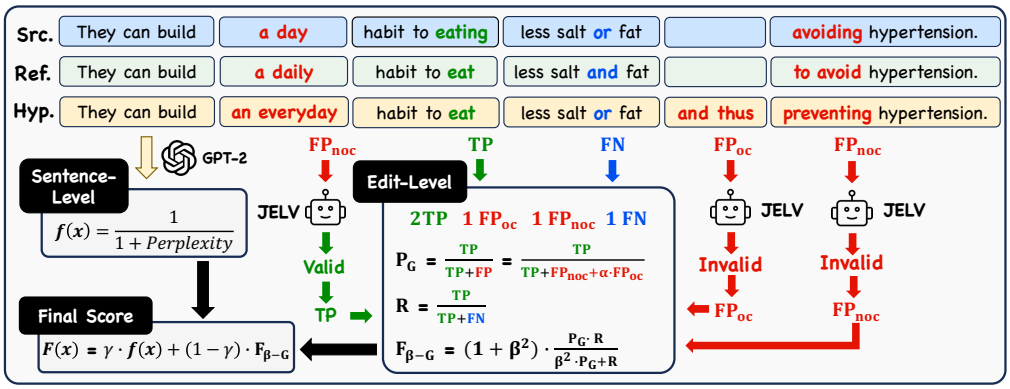

Figure 5: Overview of the comprehensive evaluation metric. Edits flagged as FPs are first reclassified via JELV2.0, then false positive decoupling distinguishes the remaining $\mathrm{FP}_{\mathrm{noc}}$ and $\mathrm{FP}_{\mathrm{oc}}$ to compute the edit-level generalized F-score $\mathrm{F}_{\beta\text{-G}}$. 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 $F(x)$.

thousand valid corrections on average, making exhaustive reference expansion infeasible and causing reference-based metrics to misclassify some valid edits as FPs inevitably. To mitigate this inherent bias, we apply JELV2.0 to re-evaluate each edit initially marked as a FP during evaluation. Edits judged valid by JELV2.0 are reclassified as TPs, achieving the equivalent of exhaustive inclusion of JELV-validated references in evaluation (*i.e.*, implicit reference expansion) without incurring the cost of explicit expansion, which reduces misclassification and enhances evaluation reliability. After reclassifying edits, we apply CLEME-independent (Ye et al. 2023), a leading F-score based metric, to compute $\mathrm{F}_{0.5}$ scores and this yields the *JELV-based CLEME*.

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

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

Combining $\mathrm{P_G}$ with recall R, we further propose the *generalized F-score* as the edit-level metric:

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

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

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

where $P(x_i|x_{<i})$ is computed by GPT-2 and $|x|$ denotes sentence length. Higher $f(x)$ indicates greater fluency.

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

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

Figure 5 presents an overview of our comprehensive evaluation metric.

4.2 Experiments

We conduct comprehensive experiments to evaluate our metric against existing approaches. Our evaluation consists of hyperparameter tuning on a training set, followed by meta-evaluations on a held-out test set of SEEDA benchmark.

Settings To reach higher correlation with human judgments at each level, we tune the hyperparameters α and γ on the training set by selecting the values that maximize the correlation[7]. Once identified, these optimal parameters are fixed and applied to the metric for evaluation on the test set. To prevent any in-domain overlap, we exploit SEEDA's two distinct domains—genetic (163 source sentences) and social media (228 source sentences)—by assigning one domain entirely to training and the other entirely to testing. Finally, we assess our metric's robustness via system-level and sentence-level meta-evaluations on the test set and compare its performance against existing metrics.

Meta-evaluation SEEDA provides edit-level (SEEDA-E) and sentence-level (SEEDA-S) pairwise human judgments for twelve GEC system corrections ("Base") and two additional fluent human corrections ("+Fluent corr.") For **system-level evaluation**, we compute the overall scores across all corrections for each GEC system, derive a system ranking, and compare it to the human ranking produced by TrueSkill (Sakaguchi, Post, and Van Durme 2014) using Pearson's r and Spearman's ρ. For **sentence-level evaluation**, we compute a score for each correction sentence, then

[7] $\alpha \in [0, 2]$, $\gamma \in [0, 1]$, both with step size 0.01.

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	0.711	0.699	-0.224	0.160	0.654	0.462	-0.293	0.002	0.578	0.156	0.458	-0.084	0.567	0.135	0.454	-0.092
ERRANT	0.628	0.629	-0.531	0.024	0.516	0.364	-0.604	-0.143	0.511	0.215	0.460	0.118	0.485	0.174	0.438	0.085
CLEME	0.644	0.587	-0.480	0.051	0.547	0.336	-0.552	-0.108	0.443	0.188	0.398	0.078	0.428	0.175	0.390	0.077
CLEME2.0	0.836	0.888	-0.589	0.187	0.716	0.636	-0.665	0.029	0.364	0.182	0.364	0.182	0.583	0.583	0.583	0.583
PT-M^2	0.820	0.902	-0.187	0.305	0.820	0.720	-0.244	0.191	0.220	-0.031	0.220	-0.028	0.228	-0.003	0.218	-0.024
GLEU	0.886	0.867	0.206	0.600	0.820	0.790	0.118	0.547	0.673	0.365	0.616	0.247	0.671	0.358	0.610	0.234
SOME	0.860	0.923	0.926	0.952	0.868	0.776	0.917	0.859	0.394	-0.034	0.405	-0.027	0.409	-0.014	0.405	-0.036
Scribendi Score	0.718	0.692	0.696	0.745	0.513	0.413	0.592	0.547	0.377	0.241	0.357	0.208	0.369	0.247	0.324	0.176
IMPARA	0.843	0.937	0.860	0.960	0.878	**0.860**	0.853	**0.912**	0.695	0.413	0.700	0.414	0.721	0.473	0.713	0.445
LLM-based	0.951	0.977	0.917	0.981	0.919	0.858	0.881	0.906	0.717	0.536	0.692	0.500	0.753	0.614	0.711	**0.557**
JELV-based F(x)	**0.975**	**0.986**	**0.974**	**0.991**	**0.932**	**0.860**	**0.947**	**0.912**	**0.780**	**0.559**	**0.770**	**0.541**	**0.807**	**0.630**	**0.772**	0.543

Table 2: Results of system- and sentence-level meta-evaluations on SEEDA's test set of social media domain. For "LLM-based", our prompt and experiment setting aligns with previous works's protocol (Kobayashi, Mita, and Komachi 2024a). **Boldface** indicates the **highest** correlation score in each column.

for every pair of corrections we derive a metric-based preference, compare this to SEEDA's human judgment, and report classification accuracy and Kendall's τ.

4.3 Result and Analysis

Main Results Table 2 reports our meta-evaluation on SEEDA's test set of social media domain, where our metric achieves state-of-the-art (SOTA) correlations with human judgments across nearly all evaluation levels and granularities. Results for the test set of genetic domain are provided in the Appendix D, showing similarly strong performance and confirming our metric's robustness across domains.

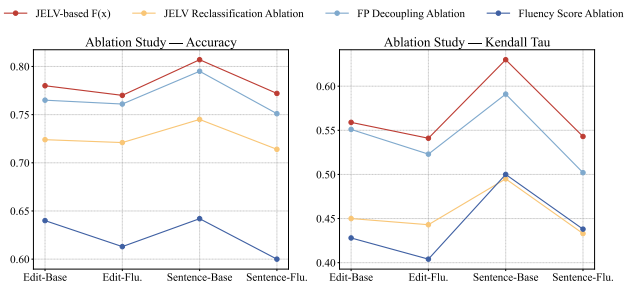

Figure 6: Ablation Study for JELV-based F(x).

Ablation Study We perform ablation studies to isolate each component—disabling JELV-based reclassification, fixing α=1 to remove FP decoupling, and fixing γ=0 to exclude fluency score integration. Figure 6 reports sentencxe-level meta-evaluation results, showing that omitting any component reduces correlation with human judgments and thus confirms the necessity of each method in the final metric $\mathrm{F(x)}$.

Analysis of Hyperparameter We investigate how the overcorrection penalty α and fluency weight γ vary across evaluation levels and granularities. We expect edit-level evaluation to require a larger α (Equation 1) than sentence-level evaluation, because it penalizes individual corrections more strictly. We also expect the "+Fluent corr." setting to require a larger γ (Equation 4) than the "Base" setting, since

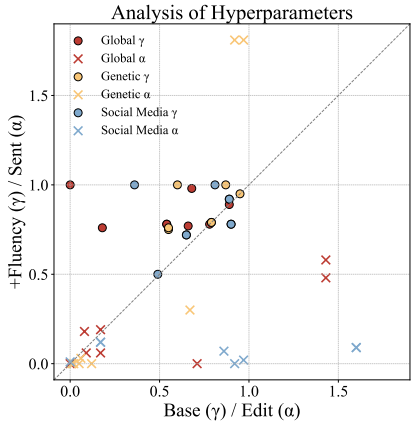

Figure 7: Comparison of optimal hyperparameters across SEEDA domains and evaluation levels. Base(γ) and +Fluency(γ) denote the optimal γ values for the "Base" and "+Fluent corr." settings, respectively; Edit(α) and Sent(α) denote the optimal α values for edit-level and sentence-level evaluation, respectively.

it places greater emphasis on fluency. To verify this, we extract the optimal α and γ values for the genetic, social media, and overall SEEDA domains at each evaluation level. Figure 7 shows that most of the edit-level α exceeds the sentence-level α, and that the "+Fluent corr." γ exceeds the "Base" γ, confirming our hypotheses and demonstrating the flexibility and robustness of our hyperparameter design. All the optimal hyperparameters can be found in Appendix E.

5 Reference Expansion and Retraining

Figure 8 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

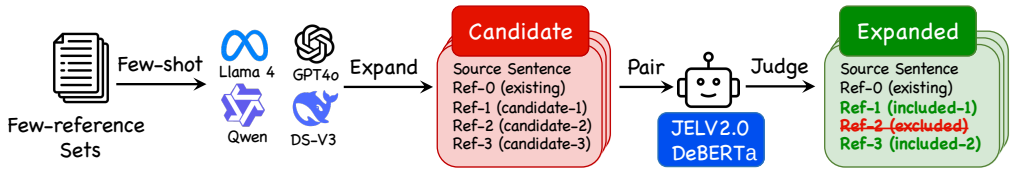

Figure 8: Automated expansion pipeline for large scale few reference sets. Four LLMs generate candidate corrections using few shot prompts. Each candidate correction is paired with its source sentence following the Sentence Pair Construction protocol and evaluated by the distilled DeBERTa classifier (JELV2.0). Valid corrections are added to the reference set to produce the expanded corpus.

Dataset	Ensembles	Majority-voting	GRECO	GEC-DI	UGEC	MoECE	SynGEC	Sequence tagging
Raw	72.68	71.44	70.99	69.43	69.38	67.45	67.29	66.31
Expanded w/o Filtering	72.98	71.44	71.25	69.57	69.62	67.74	67.47	66.36
Expanded w/ Filtering	**73.05**	**71.91**	**71.36**	**69.58**	**69.74**	**67.92**	**67.57**	**66.67**
Raw	73.54	71.84	71.34	69.37	69.44	66.87	66.68	65.09
Expanded w/o Filtering	74.08	72.25	71.60	**70.02**	70.00	67.13	66.75	65.23
Expanded w/ Filtering	**74.19**	**72.87**	**72.16**	69.84	**70.14**	**67.59**	**67.16**	**65.96**
Raw	79.47	78.10	78.08	77.34	77.80	76.78	76.80	75.42
Expanded w/o Filtering	80.29	79.60	78.98	**78.51**	78.14	76.47	76.26	75.70
Expanded w/ Filtering	**80.45**	**80.01**	**79.57**	78.41	**78.85**	**77.35**	**77.33**	**76.95**

Table 3: Evaluation scores for eight GEC systems on CoNLL14 under three BEA19 training variants. Each group of bars (top to bottom) shows M^2, CLEME, and JELV-based $F(x)$ scores. Raw: reproduction trained on raw BEA19; Expanded w/o Filtering: BEA19 plus all LLM-generated candidates; Expanded w/ Filtering: BEA19 plus JELV-validated edits. **Bold** indicates the highest score.

(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 G.2 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 F 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 reference expansion strategy on model performance, we retrain eight top CoNLL14 GEC systems[8] twice on three BEA19 variants: the raw train and dev sets; the expanded set (BEA19 plus LLM-generated candidates); and the filtered set (expanded set with JELV validation). The systems include Ensembles, Majority-voting (Omelianchuk et al. 2024), GRECO (Qorib and Ng 2023), GEC-DI (Zhou et al. 2023), Unsupervised GEC (Cao et al. 2023), MoECE (Qorib, Aji, and Ng 2024), SynGEC (Zhang et al. 2022b), and GECTOR (Omelianchuk et al. 2020). Table 3 presents each system's performance under three training variants, evaluated with M^2, CLEME, and JELV-based $F(x)$. Training on the expanded and filtered dataset yields the highest performance for most systems—significantly outperforming the Raw baseline (paired t-test and Wilcoxon signed-rank test: both $p < 0.01$), especially on JELV-based $F(x)$—Unfiltered expansion also improves over the baseline for most systems, while JELV filtering delivers further gains, underscoring the value of JELV-based quality-controlled reference expansion.

[8]Systems are selected from the CoNLL14 leaderboard, and training follows the original papers' experiment protocols.

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, 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. Second, a JELV-based generation-then-filtering pipeline automates reference expansion and retraining on the expanded dataset yields clear improvements in GEC model performance. JELV therefore provides a scalable solution to enrich reference diversity and enhance both evaluation reliability and model robustness in GEC.

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Bryant, C.; Felice, M.; Andersen, Ø. E.; and Briscoe, T. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In Yannakoudakis, H.; Kochmar, E.; Leacock, C.; Madnani, N.; Pilán, I.; and Zesch, T., eds., *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 52–75. Florence, Italy: Association for Computational Linguistics.

Bryant, C.; and Ng, H. T. 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)*, 697–707.

Bryant, C.; Yuan, Z.; Qorib, M. R.; Cao, H.; Ng, H. T.; and Briscoe, T. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3): 643–701.

Cao, H.; Yuan, L.; Zhang, Y.; and Ng, H. T. 2023. Unsupervised grammatical error correction rivaling supervised methods. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 3072–3088.

Chollampatt, S.; and Ng, H. T. 2018. A Reassessment of Reference-Based Grammatical Error Correction Metrics. In Bender, E. M.; Derczynski, L.; and Isabelle, P., eds., *Proceedings of the 27th International Conference on Computational Linguistics*, 2730–2741. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Choshen, L.; and Abend, O. 2018a. Automatic Metric Validation for Grammatical Error Correction. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1372–1382. Melbourne, Australia: Association for Computational Linguistics.

Choshen, L.; and Abend, O. 2018b. Inherent Biases in Reference-based Evaluation for Grammatical Error Correction. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 632–642. Melbourne, Australia: Association for Computational Linguistics.

Deutsch, D.; Dror, R.; and Roth, D. 2022. On the Limitations of Reference-Free Evaluations of Generated Text. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 10960–10977. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Flachs, S.; Lacroix, O.; Yannakoudakis, H.; Rei, M.; and Søgaard, A. 2020. Grammatical Error Correction in Low Error Density Domains: A New Benchmark and Analyses. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8467–8478. Online: Association for Computational Linguistics.

Ge, T.; Wei, F.; and Zhou, M. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Gomez, F. P.; and Rozovskaya, A. 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)*, 1253–1270.

Grundkiewicz, R.; Junczys-Dowmunt, M.; and Gillian, E. 2015. Human Evaluation of Grammatical Error Correction Systems. In Màrquez, L.; Callison-Burch, C.; and Su, J., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 461–470. Lisbon, Portugal: Association for Computational Linguistics.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Kobayashi, M.; Mita, M.; and Komachi, M. 2024a. Large Language Models Are State-of-the-Art Evaluator for Grammatical Error Correction. In Kochmar, E.; Bexte, M.; Burstein, J.; Horbach, A.; Laarmann-Quante, R.; Tack, A.; Yaneva, V.; and Yuan, Z., eds., *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, 68–77. Mexico City, Mexico: Association for Computational Linguistics.

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

Li, H.; Dong, Q.; Chen, J.; Su, H.; Zhou, Y.; Ai, Q.; Ye, Z.; and Liu, Y. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, Y.; Li, Z.; Jiang, H.; Zhang, B.; Li, C.; and Zhang, J. 2024. Towards Better Utilization of Multi-Reference Training Data for Chinese Grammatical Error Correction. In *Findings of the Association for Computational Linguistics ACL 2024*, 3044–3052.

Maeda, K.; Kaneko, M.; and Okazaki, N. 2022. IMPARA: Impact-Based Metric for GEC Using Parallel Data. In Calzolari, N.; Huang, C.-R.; Kim, H.; Pustejovsky, J.; Wanner, L.; Choi, K.-S.; Ryu, P.-M.; Chen, H.-H.; Donatelli, L.; Ji, H.; Kurohashi, S.; Paggio, P.; Xue, N.; Kim, S.; Hahm, Y.; He, Z.; Lee, T. K.; Santus, E.; Bond, F.; and Na, S.-H., eds., *Proceedings of the 29th International Conference on Computational Linguistics*, 3578–3588. Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Napoles, C.; Sakaguchi, K.; and Tetreault, J. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In Lapata, M.; Blunsom, P.; and Koller, A., eds., *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Vol-*

ume 2, Short Papers, 229–234. Valencia, Spain: Association for Computational Linguistics.

Ng, H. T.; Wu, S. M.; Briscoe, T.; Hadiwinoto, C.; Susanto, R. H.; and Bryant, C. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In Ng, H. T.; Wu, S. M.; Briscoe, T.; Hadiwinoto, C.; Susanto, R. H.; and Bryant, C., eds., *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 1–14. Baltimore, Maryland: Association for Computational Linguistics.

Omelianchuk, K.; Atrasevych, V.; Chernodub, A.; and Skurzhanskyi, O. 2020. GECToR–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Omelianchuk, K.; Liubonko, A.; Skurzhanskyi, O.; Chernodub, A.; Korniienko, O.; and Samokhin, I. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. *arXiv preprint arXiv:2404.14914*.

Qorib, M. R.; Aji, A. F.; and Ng, H. T. 2024. Efficient and Interpretable Grammatical Error Correction with Mixture of Experts. *arXiv preprint arXiv:2410.23507*.

Qorib, M. R.; and Ng, H. T. 2023. System combination via quality estimation for grammatical error correction. *arXiv preprint arXiv:2310.14947*.

Sakaguchi, K.; Post, M.; and Van Durme, B. 2014. Efficient Elicitation of Annotations for Human Evaluation of Machine Translation. In Bojar, O.; Buck, C.; Federmann, C.; Haddow, B.; Koehn, P.; Monz, C.; Post, M.; and Specia, L., eds., *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 1–11. Baltimore, Maryland, USA: Association for Computational Linguistics.

Xie, J.; Li, Y.; Yin, X.; and Wan, X. 2024. DSGram: Dynamic Weighting Sub-Metrics for Grammatical Error Correction in the Era of Large Language Models. *arXiv preprint arXiv:2412.12832*.

Yasunaga, M.; Leskovec, J.; and Liang, P. 2021. LM-Critic: Language Models for Unsupervised Grammatical Error Correction. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7752–7763. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Ye, J.; Li, Y.; Zhou, Q.; Li, Y.; Ma, S.; Zheng, H.-T.; and Shen, Y. 2023. CLEME: Debiasing Multi-reference Evaluation for Grammatical Error Correction. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 6174–6189. Singapore: Association for Computational Linguistics.

Ye, J.; Xu, Z.; Li, Y.; Cheng, X.; Song, L.; Zhou, Q.; Zheng, H.-T.; Shen, Y.; and Su, X. 2024. CLEME2. 0: Towards More Interpretable Evaluation by Disentangling Edits for Grammatical Error Correction. *arXiv preprint arXiv:2407.00934*.

Yoshimura, R.; Kaneko, M.; Kajiwara, T.; and Komachi, M. 2020. SOME: Reference-less Sub-Metrics Optimized for Manual Evaluations of Grammatical Error Correction. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics*, 6516–6522. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Zhang, F.; Hashemi, H. B.; Hwa, R.; and Litman, D. 2017. A Corpus of Annotated Revisions for Studying Argumentative Writing. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1568–1578. Vancouver, Canada: Association for Computational Linguistics.

Zhang, Y.; Li, Z.; Bao, Z.; Li, J.; Zhang, B.; Li, C.; Huang, F.; and Zhang, M. 2022a. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. *arXiv preprint arXiv:2204.10994*.

Zhang, Y.; Zhang, B.; Li, Z.; Bao, Z.; Li, C.; and Zhang, M. 2022b. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. *arXiv preprint arXiv:2210.12484*.

Zhou, H.; Liu, Y.; Li, Z.; Zhang, M.; Zhang, B.; Li, C.; Zhang, J.; and Huang, F. 2023. Improving Seq2Seq grammatical error correction via decoding interventions. *arXiv preprint arXiv:2310.14534*.

Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; and Liu, J. 2019. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*.

Zhu, L.; Wang, X.; and Wang, X. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) yes

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) yes

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) partial

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) NA

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) no

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes