Toward Robust Evaluation for Multilingual Grammatical Error Correction: Can Large Language Models Replace Human References?

Alla Rozovskaya
City University of New York
arozovskaya@qc.cuny.edu

Dan Roth
University of Pennsylvania and Oracle AI
danroth@seas.upenn.edu

Abstract

A standard method for evaluating grammatical error correction systems severely underestimates performance, as it compares outputs against a small, fixed set of human references, despite the large space of possible valid corrections. Prior research has shown that using *a closest-gold reference* – i.e., a human reference generated with respect to the system output rather than the original text – yields more accurate performance estimates. Yet, producing such references for each system individually is costly. We introduce *an automated method for generating closest-gold references* by prompting a large language model (LLM) with system outputs. We find that performance scores computed using automatic closest-gold references correlate well with human closest-golds, whereas standard reference-based evaluations show weak or no correlation.

Building on this insight, we use both fixed human references and closest-gold references generated by Claude and Llama to compare the performance of supervised models and GPT-4 across 14 benchmarks spanning 12 languages. Consequently, while prior work has shown that GPT-4 appears to lag behind traditional models, we demonstrate that this is due to the failures of the standard evaluation method that systematically underestimates GPT-4 performance more severely than that of supervised models. We show that a more appropriate evaluation approach, based on the closest gold method, reveals that GPT-4 outperforms traditional state-of-the-art models on almost all languages.[1]

1 Introduction

Transformer-based large language models (LLMs) have led to a paradigm shift from training supervised models on hand-labeled data (Min et al., 2024), and achieved new state-of-the-art results on numerous language tasks, while eliminating the reliance on large amounts of supervised data (Brown et al., 2020). However, experiments in grammatical error correction (GEC) yielded mixed results.

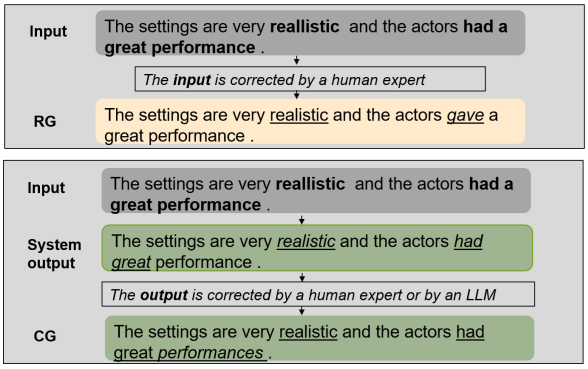

Figure 1: **Top**: An input sentence and a *fixed gold reference (RG)*, created by correcting the input text. **Bottom**: An input sentence, system output, and a *closest-gold reference (CG)*, created by correcting the output. Erroneous tokens are in bold, and the changes are underlined.

While studies on English suggest that LLMs have the ability to propose high-quality corrections, standard evaluations indicate that *LLMs lag behind state-of-the-art supervised GEC models* (Davis et al., 2024), and their performance on other languages is particularly poor (Katinskaia and Yangarber, 2024). Prior studies hypothesize, however, that LLM performance may be far better than reported, due to the weaknesses of standard evaluation metrics, especially when applied to LLM-generated outputs (Katinskaia and Yangarber, 2024).

The goals of this work are twofold: (1) we aim to develop an automated evaluation approach that more accurately reflects model performance, by emulating human references, enabling the computation of *interpretable performance metrics* such as precision, recall, and F-score; (2) we aim to assess LLM prompting for GEC in comparison to supervised models across a diverse set of languages.

To address goal (1), we adopt the evaluation methodology with *closest golds, or CGs* (Rozovskaya and Roth, 2021). CG is a reference text that is as close as possible to the system output.

[1]Data and code are available at https://github.com/arozovskaya/GEC-LLMs-Closest-Golds

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 47440–47463

Evaluation with closest-gold references provides a more realistic assessment of system performance than standard evaluation using *fixed* reference golds (RGs), which are generated independently of the system output (see Section 4.2). Rozovskaya and Roth (2021) propose to create CGs by manually correcting the system output itself, which is costly. Figure 1 illustrates the difference in the process of creating a fixed reference (RG) and a closest gold.

We introduce *an approach that creates CGs automatically*, by prompting an LLM – Claude (Anthropic, 2024) and Llama (Touvron et al., 2023) – on system outputs. We find that both Claude and Llama CGs show high correlations with human CGs on four benchmarks in three languages. In contrast, fixed references attain low correlations with human CGs. We then set out to address goal (2) and evaluate supervised models and GPT-4 on 14 GEC benchmarks, using automatic CGs.

Contributions: (1) We propose to automate the CG evaluation methodology, by making use of an LLM to generate the CGs. We show that this can be done reliably in multiple languages, as long as the model generating the CGs is good enough in the target language. We establish the reliability of LLM CGs by showing that they correlate well with human CGs. We argue that a high-performing LLM can replace human CG references, enabling more accurate evaluation of GEC models than fixed reference sets; (2) We use Claude and Llama CGs to compare the performance of supervised models and GPT-4 on 14 GEC benchmarks (both sentence and essay level); GPT-4 appears to lag behind supervised models when evaluated with fixed references. However, under closest-gold evaluation, GPT-4 exhibits stronger performance on the majority of benchmarks, with several exceptions on low-resource languages; (3) We show that Claude and Llama are consistent in their evaluations of high- and mid-resource languages, but disagree on lower-resource languages in essay-level benchmarks, following a clear pattern: Claude ranks GPT-4 outputs higher than supervised models, while Llama shows the opposite preference. We conjecture that Llama may be a weaker model for lower-resource languages and tends to make fewer corrections, which may yield artificially inflated results.

2 Related Work

The challenges of reference-based evaluation Evaluating GEC performance is known to be challenging due to a large space of correct outputs (Choshen and Abend, 2018). The standard approach makes use of reference-based measures, where system output is compared against a fixed pre-determined set of human references. A system is rewarded for proposing corrections that are in the reference(s), and penalized for proposing changes not found in the reference and for failing to identify corrections that are present in the reference(s). A sample sentence with errors (Ng et al., 2013), along with two corrected versions (reference 1 and 2), also shown in Figure 1 as RG and CG), is depicted in Figure 2: a higher overlap in edits between system output and reference 2 results in a higher F-score.

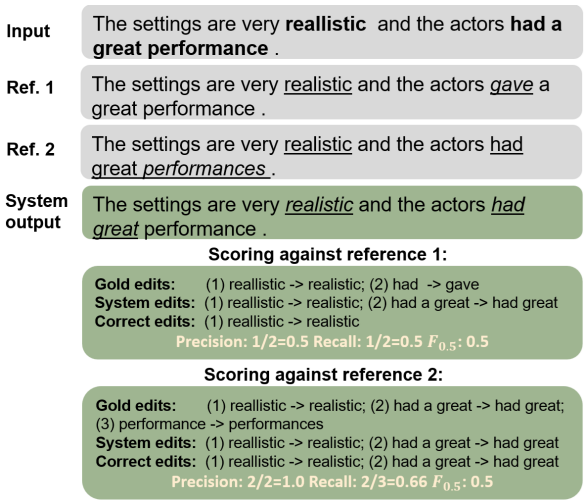

Figure 2: **Top (grey)**: Original sentence with errors, and two human references. Erroneous tokens are in bold, and the changes are underlined. **Bottom (green)**: System output and $F_{0.5}$ scores computed against reference 1 and reference 2, respectively.

A large body of work strongly suggests that evaluating against a small fixed set of references *severely underestimates system performance* (Choshen and Abend, 2018; Bryant et al., 2019; Mita et al., 2019; Bryant and Ng, 2015; Felice and Briscoe, 2015). This is because a given erroneous sentence can have many equally valid corrections. A standard evaluation compares system output against a small fixed set of references. If the system produces a perfectly valid correction that happens to differ from those references, it gets penalized unfairly. Evaluating LLM output has additional challenges. On the one hand, LLMs are known to hallucinate. On the other hand, they tend to propose "fluency corrections"[2] that may further improve text that

[2]Fluency is typically considered as an alternative method of re-writing (Napoles et al., 2017), where texts are modified for naturalness, although the two methods are not mutually

is acceptable already, further underestimating performance (Katinskaia and Yangarber, 2024; Fang et al., 2023).

Reference-free evaluations To address the above challenges, prior work has explored reference-less approaches for evaluating GEC outputs, based on grammaticality and fluency (Napoles et al., 2016; Yoshimura et al., 2020; Asano et al., 2017). Kobayashi et al. (2024) and Islam and Magnani (2021) propose metrics that use LLMs as evaluators, however, these metrics *do not provide an intuitive and interpretable evaluation of the system performance*. Furthermore, these approaches are assessed exclusively on English. This is a problem, since LLMs are known to perform much better in English than on other languages. Appendix A provides more detail on the evaluations with LLMs.

In contrast to these studies, we use an LLM to emulate human references, which allows us to use a more interpretable evaluation of model performance in terms of precision, recall, and F-score. Furthermore, we assess the use of an LLM as an evaluator for a large set of languages.

Evaluating LLMs on English GEC Most of the previous work that assessed the performance of LLMs for GEC was carried out on English (Schick et al., 2022; Dwivedi-Yu et al., 2022; Coyne et al., 2023; Davis et al., 2024). These studies suggest that LLM performance is far below that of supervised models, according to standard reference-based evaluations. It is further suggested that this happens because LLMs tend to make fluency edits (Loem et al., 2023). See Appendix A for more detail.

Prompting LLMs for GEC on languages other than English Evaluation based on reference-based metrics in languages other than English remains limited and suggests that supervised models outperform LLMs, with a larger performance gap observed in non-English languages. (Fang et al., 2023; Katinskaia and Yangarber, 2024).

3 Methodology

GEC benchmarks We use 14 publicly available GEC benchmarks in 12 languages. To our knowledge, this is the first study of such a large scope that evaluates both on sentence- and essay-level benchmarks.[3] The benchmarks are listed in Table 1.[4] On average, 65%–90% of sentences in sentence-level datasets contain at least one gold error, i.e., an error identified and corrected by a human expert. In the essay-level datasets, every paragraph contains at least one such error error. The sizes of the hand-labeled training data, reported in Appendix Table 6, provide an indication of the amount of supervision used to train the supervised models. The training sizes vary between 1M tokens (Arabic) to 5K tokens (Russian and Slovene). With the exception of Arabic and Czech, the other languages can be considered low-resource as they have fewer than 150K words of hand-labeled data for training. For comparison, English has multiple diverse hand-labeled datasets from amounting to more than 1M tokens.

Lang.	Dataset	Total inputs	Total words	Avg. len.
EN	BEA	4,384	86K	20
EN	CoNLL	1,312	30K	23
CZ	GECCC	7,909	98K	12
DE	Falko-Merlin	2,337	36K	16
ES	COWS-L2H	1,127	14K	13
RO	RONACC	1,519	17K	11
RU	RULEC	5,000	81K	16
RU	RU-Lang8	2,444	31K	13
UA	UA-GEC	1,506	24K	14
AR⋆	QALB	158	22K	144
IT⋆	Merlin	81	9K	112
LV⋆	LaVA	537	15K	28
SL⋆	Solar-EvalH	332	27K	82
SV⋆	SweLL-gold	289	12K	42

Table 1: GEC evaluation benchmarks. Datasets marked with a ⋆ are essay-level; the rest are sentence-level. *Input* refers to paragraphs and sentences in the essay- and sentence-level benchmarks, respectively. We use *CoNLL* to denote CoNLL-14. Languages: English (EN); Czech (CZ); German (DE); Spanish (ES); Romanian (RO); Russian (RU); Ukrainian (UA); Arabic (AR); Italian (IT); Latvian (LV); Slovene (SL); Swedish (SV). Last column shows average input length (words).

Supervised approaches to GEC Supervised approaches can be broken down into sequence-to-sequence (seq2seq) (Chollampatt and Ng, 2018; Yuan and Briscoe, 2016; Grundkiewicz et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019; Kiyono et al., 2019; Ji et al., 2017; Katsumata and Ko-

exclusive (Bryant et al., 2023).

[3]Sentence-level benchmarks may not contain contiguous passages but individual sentences from different essays or paragraphs. Essay-level benchmarks consist of full texts, making it possible to train models that take broader context into account.

[4]The Arabic QALB dataset contains both learner and native data. We evaluate on the learner partition.

machi., 2019; Xie et al., 2018; Sorokin, 2022; Rothe et al., 2021), and sequence-to-editing (seq2edits) (Omelianchuk et al., 2024; Awasthi et al., 2019; Li and Shi, 2021; Tarnavskyi et al., 2022). See Appendix C for more detail.

Evaluation To compare with published work, we adopt ERRANT (Bryant et al., 2017) for English (BEA), Ukrainian, Italian, Latvian, Slovene, Swedish, and Romanian, and MaxMatch (M2) scorer (Dahlmeier and Ng, 2012) for the other benchmarks. Both compute precision, recall, and $F_{0.5}$,[5] weighing precision twice as high as recall. All results are on the test partitions, with the exception of English (BEA), Italian, Latvian, Slovene, Swedish, and Ukrainian that are evaluated on the development partitions since the gold references for the test are not publicly available.

Prompting GPT-4 for GEC There has been prior work on comparing LLMs for English and other languages (see Section 2). We build on these studies and do not experiment with multiple LLMs but focus instead on high-performing LLMs. As such, we utilize the GPT-4 model (version gpt-4-0613), queried through the API provided by OpenAI.[6] We assume a setting in which the input is a single potentially ungrammatical text snippet (sentence or paragraph, in the case of essay-level benchmarks)[7] and the output is a corrected version of that text.

Similarly, we do not focus on designing appropriate prompts for GEC, but build on earlier studies for the best prompts (Coyne et al., 2023; Katinskaia and Yangarber, 2024; Davis et al., 2024). We use zero-shot prompts (see Appendix D).

Appendix Table 8 compares our GPT-4 prompting results with those from earlier work. Our results are slightly higher for English (CoNLL) and German, and are better for Spanish and Russian, although we use the same prompt as Katinskaia and Yangarber (2024). It should be noted that Katinskaia and Yangarber (2024) used GPT-3.5, whereas we use GPT-4. We conjecture that the improved performance is due to GPT-4 being a stronger model.

Prompting Claude and Llama to create closest-gold references (CGs) In Section 4.3 we propose a novel approach to generating closest-gold references automatically via LLM prompting. We select Claude and Llama as high-performing LLMs that have multilingual capabilities. We use the same prompt as above, since CG generation is essentially the same task as GEC, and we expect that prompts optimized for GEC will also be effective for generating closest-gold references. Claude and Llama are queried using outputs produced by supervised models and GPT-4. We use claude-sonnet-4-20250514 and Llama-3.3-70b-instruct.

4 Experiments and Results

4.1 Evaluation with Fixed Reference Sets

We first evaluate GPT-4 and the supervised models using fixed references.[8] For each benchmark, we use a supervised model that achieved the best result on that benchmark and denote it as *SOTA*. The performance scores of the SOTA models are obtained from the respective papers that describe the models.[9] Table 2 shows the performance of SOTA models, and compares it to zero-shot prompting with the three LLMs. Appendix Table 7 lists additional top-performing supervised models for each benchmark and provides information on each model. As shown in Table 2, SOTA outperforms all LLMs on almost all benchmarks. Two notable exceptions are Russian and Slovene (both have the smallest gold training sets among all the benchmarks). Claude obtains the highest performance among the LLMs, with GPT-4 being close, while Llama exhibits the poorest results.

Since we use Llama and Claude to generate CG references, we focus our analysis on GPT-4 vs. SOTA. Figure 3 also shows precision and recall scores of SOTA and GPT-4. SOTA outperforms GPT-4 on almost all benchmarks (exceptions are Russian and Slovene that have the least amount of supervision). The largest gap occurs for Romanian and Ukrainian. Furthermore, SOTA models tend to exhibit higher precision than GPT-4.

4.2 Evaluation with Human Closest Golds

Closest golds As discussed in Section 2, an erroneous sentence can have many equally valid corrections, and evaluating against fixed references severely underestimates performance. Rozovskaya and Roth (2021) define *a closest-gold reference* as a reference that is as close as possible to the

[5] We use $F_{0.5}$ and *F-score* interchangeably to refer to $F_{0.5}$.

[6] https://openai.com/blog/openai-api

[7] We use paragraphs and not entire essays; an essay may contain multiple paragraphs but usually no more than 2-3.

[8] We also evaluate Claude and Llama for completeness.

[9] For the Russian benchmarks, we evaluate using the model of Palma Gomez and Rozovskaya (2024) and an enhanced set of 3 fixed references, Sorokin (2022) reports the best result on RULEC using a single reference, but we were not able to obtain the outputs to score them with three references

System	EN-B	EN-C	CZ	DE	ES	RO	RU-R	RU-L	UA	AR*	IT*	LV*	SL*	SV*	All
SOTA	**63.4**	**71.8**	**73.0**	**76.8**	**58.9**	**75.5**	64.8	62.1	**65.5**	**62.5**	**65.9**	**80.6**	51.2	56.9	**65.6**
GPT-4	40.7	57.9	66.9	66.5	52.1	44.1	**65.1**	68.2	29.7	47.1	46.8	65.1	54.5	52.3	54.1
Llama	37.7	56.6	55.8	65.5	43.6	42.9	55.7	60.0	26.1	47.6	38.0	39.3	36.4	40.6	46.1
Claude	45.3	62.9	67.3	68.0	53.5	64.4	63.5	**71.3**	31.4	52.3	50.3	72.6	**55.5**	**58.1**	58.3

Table 2: Performance ($F_{0.5}$) of top-performing supervised models (SOTA), and 3 LLMs via zero-shot prompting, evaluated using standard references (RGs). *EN-B*, *EN-C*, *RU-R*, and *RU-L* stand for EN (BEA), EN (CoNLL), RU (RULEC), and RU (Lang8), respectively. Datasets marked with a ⋆ are essay-level, the rest are sentence-level. The last column shows performance averaged over all benchmarks. Best result for each benchmark is in bold. Column *All* reports results averaged over 14 benchmarks. **Lesson from the table:** *Under standard reference-based evaluation (RGs), the LLMs substantially underperform SOTA models on most languages and benchmarks.*

system output within the space of all possible valid outputs for given input. To obtain such a reference, they construct closest-gold references by directly correcting system outputs. This contrasts with *fixed* reference golds, which are created independently of the system output.

Evaluating against closest golds arguably provides a more realistic evaluation of system performance: a closest-gold reference, on the one hand, directly quantifies the remaining errors in system output, and, on the other hand, incorporates all valid edits proposed by the system. The latter is made possible by correcting system output itself. Thus, evaluating against a closest-gold reference does not penalize valid corrections that are absent from an arbitrary fixed reference, resulting in greater overlap with the edits proposed by the system (see Figure 2). We argue that this is a more principled measure of correction quality, since it avoids penalizing legitimate edits, yet, reflects the remaining edits that the system had missed. Appendix F provides more detail on the evaluation with CGs.

Creating human CG references We have performed human annotation for four benchmarks in three languages – English (BEA and CoNLL), Arabic, and Russian (RULEC). We follow Rozovskaya and Roth (2021) and present the SOTA and GPT-4 outputs to the annotator together with the original sentence. The annotator is asked to correct the remaining errors, with a focus on maintaining the original meaning (see Appendix G for the annotation guidelines). The annotators are fluent native speakers of the target language. Two of the annotators are trained linguists, and two other annotators are students with background in Language Technology.

For each dataset, we select a random subset of inputs corrected by the corresponding SOTA model and GPT-4 and annotate these with human CGs.

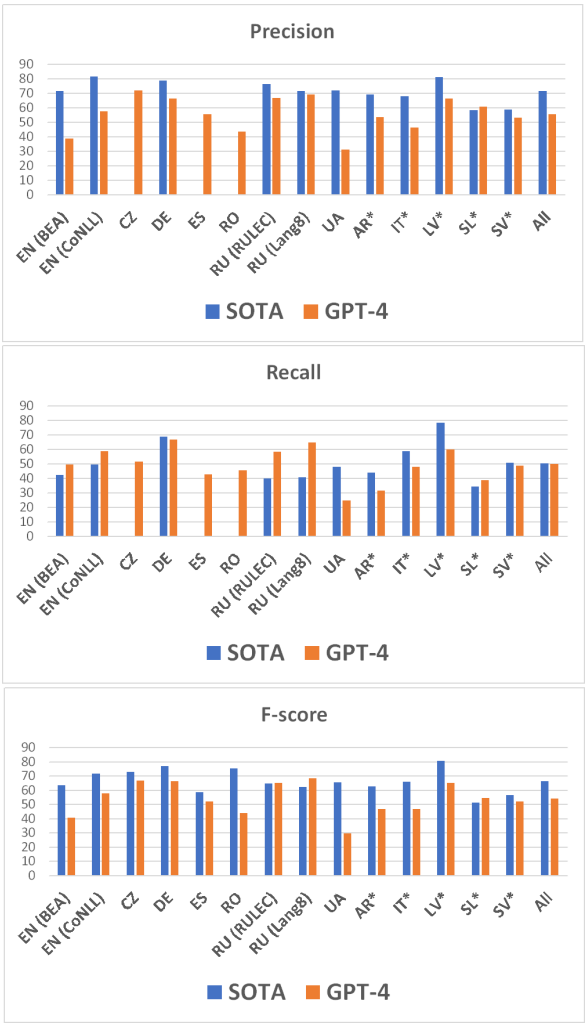

Figure 3: Performance of SOTA and GPT-4, using fixed references (RGs). *SOTA* is defined as the top-performing supervised model by language and benchmark. For Czech, Romanian and Spanish, precision and recall are not available in the published papers. ⋆ denotes essay-level benchmarks. *All* shows performance averaged over all benchmarks. **Lesson from the figure:** *Under standard reference-based evaluation (RGs), GPT-4 underperforms SOTA models on most benchmarks, with SOTA models exhibiting notably higher precision.*

System	References	EN (BEA)	EN (CoNLL)	RU (RULEC)	AR (QALB)
SOTA	CG-H vs. RG	0.14☆	0.23☆	0.3★	0.07
	CG-H vs. CG-C	0.6★	0.56★	0.45★	0.25☆
	CG-H vs. CG-L	0.6★	0.5★	0.34★	0.27☆
GPT-4	CG-H vs. RG	-0.002	0.10☆	0.02	0.35★
	CG-H vs. CG-C	0.23☆	0.30★	0.46✷	0.79✷
	CG-H vs. CG-L	0.29☆	0.32★	0.27✷	0.72✷

Table 3: Pearson **r** between human closest golds (CG-H) vs. Claude closest golds (CG-C), Llama closest golds (CG-L), and standard fixed reference golds (RG). ✷ denotes strong positive correlation; ★ denotes moderate to strong positive correlation. ✧ denotes moderate positive correlation; ☆ denotes weak positive correlation.

We use the following number of inputs: Russian (RULEC) – 200; English (BEA and CoNLL) – 150 each; and Arabic – 30. This amounts to a comparable number of annotated words in each dataset (3,500-4,000) (excluding inputs shorter than 10 tokens that are typically sentence fragments).[10] We refer to the resulting corrected outputs as human CGs. We use the resulting human CGs to create M2 files, by aligning each source sentence with its corrected version using ERRANT (Bryant et al., 2017). These M2 files are then used to score the corresponding outputs of the supervised models and GPT-4 with the M2 scorer or ERRANT.

Results of evaluation using human CGs Figure 4 shows the results of evaluating supervised models and GPT-4 on the subsets of the four benchmarks, using standard references (RGs) and human CGs. Using closest-gold references leads to higher scores across all models and benchmarks compared to standard evaluation. Notably, using the CG evaluation methodology, GPT-4 outperforms supervised models on all four benchmarks: With closest-gold references, *GPT-4 scores increase more substantially than those of supervised models*, suggesting that *GPT-4 performance is more severely underestimated under standard fixed-reference evaluation*. Furthermore, the gap is more pronounced for precision, i.e., closest-gold evaluation increases GPT-4 precision more than SOTA precision. This indicates that, compared to the SOTA models, GPT-4 proposes more valid corrections, which are not captured by the fixed references. In fact, closest-gold evaluation reveals that GPT-4 achieves precision comparable to SOTA models, in contrast to standard reference-based evaluation. Lastly, while CGs do not significantly affect the recall scores of the supervised models, they do raise the recall of GPT-4. Overall, *evaluation with CGs demonstrates that GPT-4 performance is more severely underestimated, when fixed sets of references are used.* Appendix Table 10 tabulates the results in Figure 4.

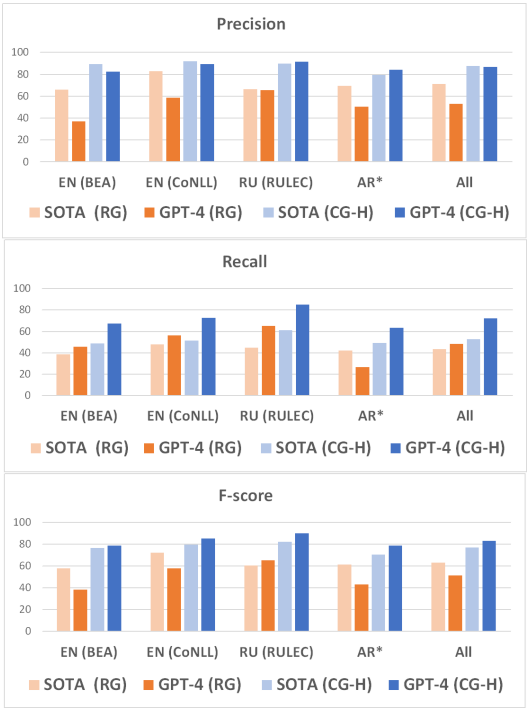

Figure 4: Evaluation with fixed reference sets (RGs) and human closest gold references (CG-H) on the subsets by language. *All* denotes scores averaged over the four subsets. **Lesson from the figure:** *Evaluating against human CGs leads to higher scores across all models and benchmarks, compared to standard evaluation. Notably, GPT-4 outperforms supervised models on all four benchmark subsets.*

4.3 Automatic CG Generation

We now create CGs automatically by prompting Claude and Llama on the outputs produced with SOTA models and GPT-4.

[10] A small proportion of inputs is excluded: AR: 0%; EN (CoNLL): 1.5%; EN (BEA): 9%;; RU (RULEC): 8%.

Comparing human and automatics CGs First, we establish the credibility of using LLMs for generating CGs, by verifying whether using automatic CGs produces results that are close to those produced with human CGs. To this end, we evaluate the performance on the subsets of the four benchmarks, for which we generated human CG references.

Figure 5 reports the results of this comparison. Scoring against Claude and Llama CGs produces results close to those obtained with human CGs. Crucially, *both Llama and Claude CGs produce the same system ranking as human CGs*, consistently ranking GPT-4 higher than the SOTA models. While evaluation with RGs suggests that SOTA are superior to GPT-4, both human and automatic CGs score GPT-4 higher for all the benchmarks. Further, while automatic and human CG scores are very close for precision, Claude and Llama scores are consistently higher for recall, compared to human CGs. This suggests that *Claude and Llama accept a similar number of edits proposed by the system, but the human annotators identify additional errors to be corrected in the outputs*. Finally, both Llama and Claude produce very close scores (within 1-2 points) for the four benchmarks, with Llama CGs resulting in slightly higher scores. Appendix Table 11 tabulates the results presented in Figure 5.

We compute Pearson correlation coefficient r (Freedman et al., 2007) between human and Claude CGs, between human and Llama CGs, and between human CGs and fixed references, by comparing input-level $F_{0.5}$ scores for the same output. The results shown in Table 3 indicate a positive correlation (mostly moderate, and weak in some cases, with Arabic QALB having a strong correlation) between human CGs and automatic CGs. Moreover, the correlation values with human CGs are very close for Claude and Llama. Lastly, fixed references have no correlation with human CGs.

The key points based on the evaluation with CGs are: (1) Using RGs does not produce the same system ranking as with human CGs, and RGs have no correlation with human CGs; (2) In contrast, Claude and Llama CG rankings always agree and have positive correlations with the human CGs; (3) Precision is consistently higher with Claude and Llama CGs, compared to human CGs.

4.4 Scoring with Automatic CGs

We use Claude and Llama to score GPT-4 and SOTA outputs. For the following benchmarks, the outputs for the SOTA models in Table 2 were not available: Czech, German, Spanish, and Romanian. In these cases, we train supervised models following Stahlberg and Kumar (2024) (see Appendix H). The performance of the models is very close to SOTA for all of these languages.

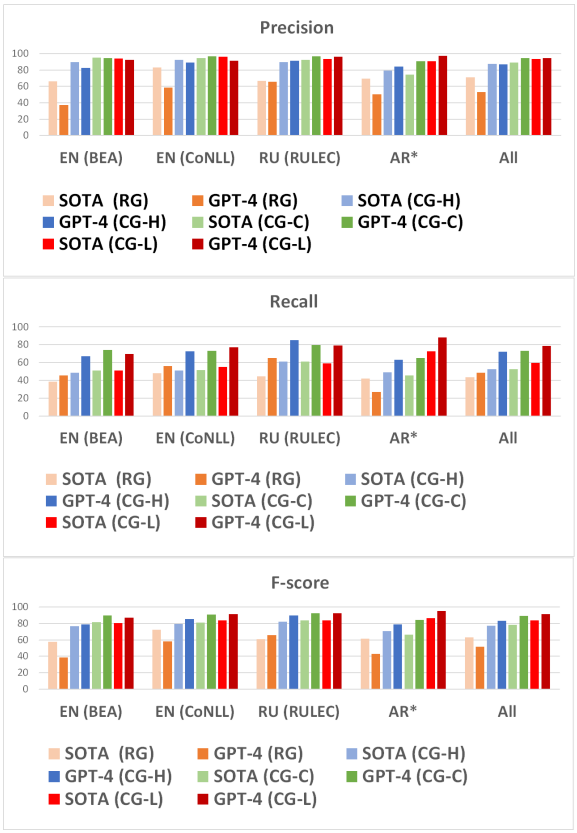

Figure 5: Evaluation with fixed references (RGs), human closest golds (CG-H), closest golds produced by Claude (CG-C) and by Llama (CG-L) (on the subsets by language). *All* denotes averaged scores over the four subsets. **Lesson from the figure:** *Scoring against Claude and Llama CGs produces results close to those obtained with human CGs: both Llama and Claude CGs produce the same system ranking as human CGs.*

Ranking of SOTA and GPT-4 Table 4 presents the results of the evaluations using RGs and automatic CGs. Figure 6 shows the differences in $F_{0.5}$ between SOTA and GPT-4, when using fixed references vs. Claude and Llama CGs. A positive value indicates that GPT-4 outperforms SOTA. Evaluation with both Llama and Claude CGs reveals that GPT-4 outperforms SOTA models on all sentence-level benchmarks, with the exception of Ukrainian, and on the essay-level Arabic benchmark. On Ukrainian and the remaining essay-level datasets, SOTA and GPT-4 are tied based on Llama CGs, with Clade CGs suggesting a slight advantage to GPT-4.

Performance gap between SOTA and GPT-4

Sys.	Refs.	EN-B	EN-C	CZ	DE	ES	RO	RU-R	RU-L	UA	AR*	IT*	LV*	SL*	SV*	All
SOTA	RG	62.9	71.1	72.0	73.6	57.8	72.1	64.8	62.1	65.1	62.5	65.9	80.6	51.2	56.9	65.6
	CG-C	79.0	82.0	82.5	79.1	78.5	84.7	62.2	62.8	72.01	63.1	78.1	88.3	82.5	90.2	77.5
	CG-L	73.2	76.3	82.7	81.0	73.5	69.6	65.4	65.6	73.7	86.5	93.1	92.4	91.7	89.7	79.6
GPT-4	RG	40.7	57.9	66.9	66.5	52.1	44.1	65.1	68.2	29.7	47.1	46.8	65.1	54.5	52.3	54.1
	CG-C	90.8	94.4	88.5	91.6	89.2	88.5	84.5	88.8	70.7	84.8	86.8	90.9	90.4	92.9	88.1
	CG-L	83.8	87.3	86.2	90.0	82.1	78.8	83.1	84.1	69.7	93.8	93.4	93.7	92.3	90.8	86.4

Table 4: Performance ($F_{0.5}$) of SOTA models and GPT-4, evaluated using standard references (RGs), Claude CGs (CG-C) and Llama CGs (CG-L). *EN-B*, *EN-C*, *RU-R*, and *RU-L* stand for EN (BEA), EN (CoNLL), RU (RULEC), and RU (Lang8), respectively. Datasets marked with a ⋆ are essay-level, the rest are sentence-level. The last column shows performance averaged over all benchmarks. **Lesson from the table:** *Claude exhibits stronger preference for GPT-4 outputs over SOTA, compared to Llama.*

The largest performance gaps are observed for the high-resource languages on both Llama and Claude CGs:[11] Russian, English, German and Spanish. Claude and Llama exhibit consistency on these languages not only for ranking but also for the degree of preference for GPT-4, indicating that both LLMs can score outputs equally confidently. However, other (mid- and low-resource) languages – Czech, Romanian – display smaller gaps indicating that the difference between SOTA and GPT-4 may not be as strong. Finally, on the lowest-resource languages – Ukrainian, Latvian, Slovene – as well as mid-resource languages (Swedish and Italian) GPT-4 and SOTA models perform similarly, according to both LLMs. These benchmarks (except for Ukrainian) are also essay-level, suggesting that GPT-4 might have difficulty with longer contexts.[12] Overall, mid-resource and low-resource languages exhibit less consistency between the two LLM rankings, compared to high-resource languages. Finally, Table 4 demonstrates that Llama scores tend to be higher than Claude's for SOTA outputs, whereas GPT-4 outputs are ranked higher by Claude. As such, *Claude exhibits stronger preference for GPT-4 outputs over SOTA, compared to Llama*.

Claude and Llama disagreements Claude and Llama CGs provide the same rankings for SOTA and GPT-4 on 9 out of 14 benchmarks. On Ukrainian, Llama prefers SOTA outputs, while Claude scores them equally. On the four essay-level benchmarks (Italian, Latvian, Slovene, and Swedish), Claude scores GPT-4 higher than SOTA outputs, whereas Llama scores them equally. We

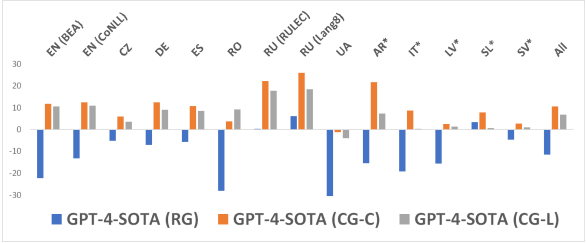

Figure 6: $F_{0.5}$ differences (score (GPT-4) – score (SOTA)). A positive difference indicates that GPT-4 outperforms SOTA, whereas a negative value indicates that SOTA surpasses GPT-4. **Lesson from the figure:** *Llama and Claude CGs show that GPT-4 outperforms SOTA models on nearly all sentence-level benchmarks. On the essay-level benchmarks results are tied with Llama CGs or slightly favor GPT-4 (Claude CGs).*

conjecture that Llama may be a weaker model for these low-resource languages (Benkirane et al., 2024) and struggles with longer inputs. The disagreements on Italian and Swedish suggest that even on mid-resource languages LLMs may not be reliable, when longer inputs are involved.

5 Analysis and Discussion

How human and automatic CGs differ Human and automatic CGs provide the same ranking for the outputs of SOTA and GPT-4 for the four language benchmarks in our study, reversing the results obtained with the standard fixed reference sets. Furthermore, precision scores obtained with Claude and Llama CGs are consistently higher than the scores based on human CGs, whereas recall scores when using auomatic CGs are similar to those obtained with human CGs (Figure 5).

To understand the reason for higher precision scores with automatic CGs, we inspect the outputs from the English CoNLL benchmark, and identify instances of disagreement between Claude and human CGs. We find that Claude may accept edits proposed in the system output that are not accepted

[11] Here, we define high-, mid-, and low-resource languages in terms of the amount of monolingual data available for LLM pre-training, e.g., Xue et al. (2021).

[12] We hypothesize that the combination of a low-resource language and longer contexts is the most challenging, since the Arabic essay-level benchmark exhibits stronger GPT-4 performance, as a mid-resource language.

Input	They were told that the wife 's family was carrying the polygenetic disorder and can pass only to boys rather than girls .
Output	They were told that the wife 's family was carrying the polygenetic disorder and it could be passed only to boys rather than girls .
CG-H	They were told that the wife 's family was carrying the polygenetic disorder and that it could only be passed to boys rather than girls .
CG-C	They were told that the wife 's family was carrying the polygenic disorder and it could be passed only to boys rather than girls .
Input	In conclusion , the shadow always exists when there is light .
Output	In conclusion , the shadow always exists when there is light .
CG-H	In conclusion , shadow always exists when there is light .
CG-C	In conclusion , the shadow always exists when there is light .

Table 5: Example of an original sentence (input); system output; the human closest gold (CG-H), and the Claude closest gold (CG-C). The parts of the sentence where the human and the Claude CGs disagree are underlined.

by human annotators. However, we do not observe cases where Claude accepts clearly incorrect suggestions. Table 5 provides such examples.

Over-correction One issue frequently observed with LLMs is the issue referred to as *over-correction* or *fluency edits*, where an LLM has the tendency to make edits beyond what is required and will modify output that may be acceptable (Katinskaia and Yangarber, 2024; Fang et al., 2023; Coyne et al., 2023). We distinguish two cases: (1) *fluency over-correction* – the model replaces an acceptable word with another word or expression that is as good or more fluent, without changing the meaning; (2) *incorrect over-editing* – proposing a change that modifies the meaning of the original sentence.

We have an annotator identify and classify the over-correction cases in the GPT-4 and SOTA outputs in the RULEC subset (see Table 15). GPT-4 has a higher frequency of fluency over-corrections (15% of sentences vs. 2% for the SOTA models). However, the SOTA model introduces more false positives by modifying well-formed input and changing the meaning of the sentence. Most of the over-corrections in the GPT-4 outputs are due to rephrasing (word change) or word reordering. In general, *GPT-4 over-correction improves the input*.

Model self-bias We compare the rankings of the four models using Claude and Llama CGs in Figure 7. While Claude and Llama agree in the rankings of SOTA and GPT-4, the LLMs are clearly biased towards their own outputs. Specifically, Claude outputs receive higher scores than Llama outputs when evaluated with Claude CGs, whereas Llama CGs indicate that Llama performs comparably to or better than Claude. This suggests model self-bias, and we advise against using the same LLM for grammar correction and for evaluation. See Appendix J for further discussion.

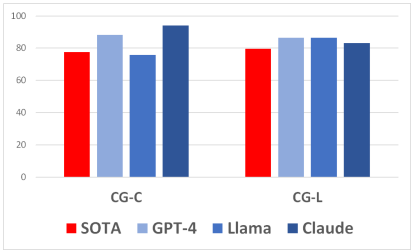

Figure 7: $F_{0.5}$ scores for the SOTA and the three LLMs (averaged over the 14 benchmarks) obtained with Claude CGs (CG-C) vs. LLama CGs (CG-L). **Lesson from the figure:** *Claude and Llama agree on the relative ranking of SOTA models and GPT-4, but differ in how they rank their own outputs, with each assigning higher scores to its own output than to that of the other model.*

6 Conclusion

We propose to automate the evaluation methodology that uses closest-gold references (CGs), for a more accurate evaluation of GEC system performance. The approach emulates human references, thereby providing *interpretable performance metrics*: precision, recall, and F-score. We generate CGs with two high-performing LLMs – Claude and Llama – and show that both have positive correlation with human rankings. We conduct a study across 14 GEC benchmarks. To our knowledge, this is the first study of such a large scope, providing insight into GEC LLM performance beyond English.

We demonstrate that the CG methodology can be automated and used reliably for a more accurate evaluation of GEC models, as long as the LLM used to generate the CGs is good enough in the target language. The proposed methodology should be valuable for all languages, due to the challenges of evaluation and the cost of creating human references. However, on low-resource languages the methodology may not be reliable yet, especially when longer contexts are involved.

Limitations

In this work, we have introduced the approach to automate the evaluation methodology based on the idea of closest-gold references. We have used two LLMs – a commercial one (Claude) and an open-source one (Llama) for generating CGs. We have not evaluated other LLMs, and we hypothesize that weaker LLMs would not do as well.

Further, we have used four benchmarks in three languages (English, Russian, and Arabic) to establish the credibility of the CG evaluation methodology. We further conjecture that the quality of the resulting closest-gold references depends on the target language and its representation in the monolingual pre-training data. Both English and Russian are considered high-resource in this regard, whereas Arabic can be viewed as mid-resource.

That said, for the low-resource languages that are not well-represented in the monolingual pre-training data, more research is needed to evaluate the quality of the proposed methodology. We plan this for future work.

One question that this raises is how can we determine whether an LLM is sufficiently strong to evaluate a given language? We recommend considering the model's performance on other NLP tasks, as well as the amount of monolingual data for that language used during pre-training, if possible. As noted above, we find the methodology to be reliable for high- and mid-resource languages. For low-resource languages, one may consider combining multiple LLMs and/or using models specifically tailored to the target language.

Another limitation of this work is that the generation of closest-gold references does not take into account the constraint that corrected texts should retain their original meaning. This may potentially raise a concern if the system dramatically changes the sentence meaning. That said, we have performed an analysis of system outputs for instances of over-correction in SOTA and GPT-4 (see Section 5) on the Russian RULEC benchmark to quantify how often sentence meaning is modified. Our manual analysis by a native annotator indicates that in the RULEC subset (200 sentences, 4.3K tokens) there were 16 and 12 over-editing changes in the SOTA and GPT-4 model outputs, respectively, which is quite infrequent.[13] Based on these results, we believe that for other high-resource languages over-editing that results in meaning changes would not be common in strong LLMs such as GPT-4 (and with strong SOTA supervised models). However, it is possible that it could be an issue for lower-resource languages that have little supervision and/or little monolingual data in LLM pre-training. As such, we recommend that similar analyses be conducted on other languages.

Another question that we have not investigated is LLM potential for homophily bias. Specifically, the CG methodology involves using one class of models (high-performing LLMs) to evaluate another LLM (GPT-4) against a different class of models (supervised SOTA). This raises a concern for a potential for an "LLM-style" bias, where the evaluator LLM might be more likely to approve of the kinds of stylistic or fluency-oriented corrections that another LLM produces. While we have discussed and evaluated self-bias in LLM evaluators, measuring and controlling homophily bias would require different approaches, and, arguably, would be harder to measure than self-bias. For example, one could generate human closest golds for a small subset of the inputs and measure whether LLM evaluator scores have the same distance from the human CGs when scoring LLMs and supervised SOTA. Another direction is to compare multiple LLM evaluators and assess their agreement when scoring LLMs versus supervised SOTA models. We believe this is an important and exciting direction for future work. That said, the preference bias is, arguably, also present with human raters who might have a preference for a specific type of correction style.

Why evaluation is necessary One may argue that the concept of automatically generating CG references is redundant, given that we already have sufficiently good GEC models. In such a scenario, why do we still need evaluation metrics? We believe that evaluation is still necessary and is needed for several reasons: (1) We want to compare model performance; which one is best to use (in a given use case); (2) In order to improve solutions, we need to have a reliable evaluation. No solution is perfect, and if we want to improve (on specific use cases, languages, phenomena), we need to be able to have a reliable evaluation method; (3) The CG methodology is useful in educational settings, to evaluate human performance.

Finally, we emphasize that this work does not advocate for discarding human references. Rather, we

[13] We hypothesize that this is because the prompts we use are optimized to perform minimal edits.

argue that the closest-gold evaluation methodology enables a more accurate assessment of GEC system performance, compared to using fixed references, and propose automating the generation of Closest Gold references (CGs), using LLMs, rather than using human experts.

Ethical Considerations

The human annotation presented in this work (for the subsets of the four benchmarks) was manually generated by two native English speakers, a native Russian speaker, and a native Arabic speaker that were hired to perform the annotation for a compensation. The amount was set according to a compensation that was offered for similar annotation efforts, and that pay was deemed acceptable by the annotators. The annotators were informed that the annotations will be used to perform research in computational linguistics. The annotations are available for research purposes.

The authors are not aware of any potential risks that could result from the use of the data and the annotations.

Acknowledgments

The authors thank the annotators for their annotation work. The authors are grateful to the anonymous ARR reviewers for their insightful comments.

References

Bashar Alhafni, Go Inoue, Christian Khairallah, and Nizar Habash. 2023. Advancements in Arabic grammatical error detection and correction: An empirical investigation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6430–6448, Singapore. Association for Computational Linguistics.

Anna Alsufieva, Olsya Yatsenko Kisselev, and Sandra G. Freels. 2012. Results 2012: Using flagship data to develop a russian learner corpus of academic writing. *Russian Language Journal*, 62:79–105.

Anthropic. 2024. Introducing the next generation of claude.

Hiroki Asano, Tomoya Mizumoto, and Kentaro Inui. 2017. Reference-based metrics can be replaced with reference-less metrics in evaluating grammatical error correction systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Kenza Benkirane, Laura Gongas, Shahar Pelles, Naomi Fuchs, Joshua Darmon, Pontus Stenetorp, David Ifeoluwa Adelani, and Eduardo Sánchez. 2024. Machine translation hallucination detection for low and high resource languages using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics.

Maksym Bondarenko, Artem Yushko, Andrii Shportko, and Andrii Fedorych. 2023. Comparative study of models trained on synthetic data for Ukrainian grammatical error correction. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*. Association for Computational Linguistics.

Adriane Boyd. 2018. Using wikipedia edits in low resource grammatical error correction. In *Proceedings of the 4th Workshop on Noisy User-generated Text*. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

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*. 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)*. Association for Computational Linguistics.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *ACL*.

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*, pages 643–701.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction . In *Proceedings of the AAAI*. Association for the Advancement of Artificial Intelligence.

Leshem Choshen and Omri Abend. 2018. Inherent biases in reference-based evaluation for grammatical error correction and text simplification. In *ACL*.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. Neural grammatical error correction for romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction. In *arXiv preprint arXiv:2303.14342*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. A beam-search decoder for grammatical error correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 568–578, Jeju Island, Korea. Association for Computational Linguistics.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. Developing NLP tools with a new corpus of learner Spanish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France. European Language Resources Association.

Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. Prompting open-source and commercial language models for grammatical error correction of English learner text. In *Findings of ACL 2024*.

Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, , and Fabio Petroni. 2022. Editeval: An instruction-based benchmark for text improvements. In *arXiv preprint abs/2209.13331*.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is ChatGPT a highly fluent grammatical error correction system? In *arXiv preprint arXiv:2304.01746*.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, Online. Association for Computational Linguistics.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. In *arXiv preprint abs/1901.05287*.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT*.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the ACL Workshop on Innovative Use of NLP for Building Educational Applications (BEA-19)*.

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*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, , and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *ACL*.

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

Anisia Katinskaia and Roman Yangarber. 2024. GPT-3.5 for grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. ELRA and ICCL.

Satoru Katsumata and Mamoru Komachi. 2019. (almost) unsupervised grammatical error correction using synthetic comparable corpus. In *Proceedings of the ACL Workshop on Innovative Use of NLP for Building Educational Applications (BEA-19)*.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics.

Yova Kementchedjhieva and Anders Søgaard. 2023. Grammatical error correction through round-trip machine translation. In *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference*

on Natural Language Processing (EMNLP-IJCNLP), pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. 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)*. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*. European Association for Machine Translation.

Piji Li and Shuming Shi. 2021. Tail-to-tail non-autoregressive sequence prediction for Chinese grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2024. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*.

Agnes Luhtaru, Taido Purason, Martin Vainikko, Maksym Del, and Mark Fishel. 2024. To err is human, but llamas can learn it too. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025a. The MultiGEC-2025 shared task on multilingual grammatical error correction at NLP4CALL. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*.

Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, Robert Östling, Kais Allkivi, Špela Arhar Holdt, Ilze Auzina, Roberts Darģis, Elena Drakonaki, Jennifer-Carmen Frey, Isidora Glišić, Pinelopi Kikilintza, Lionel Nicolas, Mariana Romanyshyn, Alexandr Rosen, Alla Rozovskaya, Kristjan Suluste, Oleksiy Syvokon, Alexandros Tantos, Despoina-Ourania Touriki, Konstantinos Tsiotskas, Eleni Tsourilla, Vassilis Varsamopoulos, Katrin Wisniewski, Aleš Žagar, and Torsten Zesch. 2025b. Towards better language representation in natural language processing : a multilingual dataset for text-level grammatical error correction. *INTERNATIONAL JOURNAL OF LEARNER CORPUS RESEARCH*, 11(2):309–335.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2024. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. 56(2).

Masato Mita, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, and Kentaro Inui. 2019. Cross-corpora evaluation and analysis of grammatical error correction models – is single-corpus evaluation enough? In *NAACL*.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. Czech grammar error correction with a large and diverse corpus. *Transactions of the Association for Computational Linguistics*, 10:452–467.

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

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.

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

Mihai Niculescu, Stefan Ruseti, and Mihai Dascalu. 2021. Rogpt2: Romanian gpt2 for text generation. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*.

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

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

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, St. Julian's, Malta. Association for Computational Linguistics.

Frank Palma Gomez, Alla Rozovskaya, and Dan Roth. 2023. A low-resource approach to the grammatical error correction of ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop in conjunction with EACL*.

Muhammad Reza Qorib, Alham Fikri Aji, and Hwee Tou Ng. 2024. Efficient and interpretable grammatical error correction with mixture of experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics.

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

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

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015. The second QALB shared task on automatic text correction for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35, Beijing, China. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Alla Rozovskaya and Dan Roth. 2021. How good (really) are grammatical error correction systems? In *EACL*.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. PEER: A collaborative language model. In *arXiv preprint arXiv:2208.11663*.

Alexey Sorokin. 2022. Improved grammatical error correction by ranking elementary edits. In *EMNLP*.

Felix Stahlberg and Shankar Kumar. 2024. Synthetic data generation for low-resource grammatical error correction with tagged corruption models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Ryszard Staruch. 2025. UAM-CSI at MultiGEC-2025: Parameter-efficient LLM fine-tuning for multilingual grammatical error correction. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*. University of Tartu Library.

Oleksiy Syvokon, Olena Nahorna, Pavlo Kuchmiichuk, and Nastasiia Osidach. 2023. UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 96–102, Dubrovnik, Croatia. Association for Computational Linguistics.

Oleksiy Syvokon and Mariana Romanyshyn. 2023. The UNLP 2023 shared task on grammatical error correction for Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop in conjunction with EACL*.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error

correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models.

Viet Anh Trinh and Alla Rozovskaya. 2021. New dataset and strong baselines for the grammatical error correction of russian. In *ACL Findings*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 619–628, New Orleans, Louisiana. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

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*. 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*. Association for Computational Linguistics.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *NAACL*.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

A Details on Related Work

Evaluating LLMs on English GEC Most of the previous work that assessed the performance of LLMs for GEC was carried out on English. Schick et al. (2022) employed a simple zero-shot prompt for GEC, while Dwivedi-Yu et al. (2022) used diverse zero-shot prompts. Coyne et al. (2023) analyzed the performance of GPT-3.5 and GPT-4 on two English benchmarks and compared several prompts in the zero-shot and few-shot settings. They found that, given a suitable prompt, LLMs behave reliably in the single-sentence prompt setting, generating no unexpected sequences. They also found that adding few-shot examples has a slight positive effect on GPT-3.5 but no or negative effect on GPT-4 performance. Davis et al. (2024) evaluated seven LLMs on four English benchmarks using zero-shot and few-shot prompts, with the goal of eliciting performance using minimal-edit style corrections. Although Davis et al. (2024) specifically aim to elicit minimal-style edits, they observe that LLMs are still prone to making fluency edits that score lower on the standard references created using the minimal-edit principle. Loem et al. (2023) evaluated the performance of GPT-3 on 3 English benchmarks, using both zero-shot and few-shot settings, and compared the use of several prompts against supervised and unsupervised baseline models. They found that GPT-3 outperforms only unsupervised baselines in the zero-shot setting. All of the above studies used standard reference-based evaluation to compare LLM performance with that of supervised models, and found that LLM performance is far below that of supervised models. They further suggested that this happens because LLMs tend to make fluency edits (Loem et al., 2023).

Prompting for GEC on Other Languages Evaluation of LLMs on languages other than English is quite limited. Fang et al. (2023) used zero-shot and few-shot chain-of-shot (CoT) prompts to evaluate

ChatGPT on English, German, and Chinese. Using standard reference-based evaluations, they found that ChatGPT performs worse compared to supervised models. Katinskaia and Yangarber (2024) used zero-shot and zero-shot CoT prompts to evaluate GPT-3.5-turbo on seven language benchmarks. In line with previous studies, they found supervised models to be superior, although the LLM showed significantly higher recall compared to supervised models. Additionally, while for English the corrected sentences remained semantically similar to the source, for languages other than English, the LLM substantially altered the source sentences, including their semantics, as well as tended to "over-correct" inputs that were already well-formed.

Reference-free evaluations To avoid the reliance on human-generated references, prior work has explored reference-less approaches for evaluating GEC outputs, based on grammaticality and fluency (Napoles et al., 2016; Yoshimura et al., 2020; Asano et al., 2017).

In the last few years, there have been studies on the use of LLMs as evaluators for text generation tasks such as summarization and machine translation (Kocmi and Federmann, 2023; Liu et al., 2024). LLMs have been used for assessing grammaticality for specific language phenomena such as subject-verb agreement (Linzen et al., 2016; Goldberg, 2019) or as components of supervised GEC systems (Yasunaga et al., 2021).

Studies that assess the use of LLMs as evaluators for GEC are focused on English, furthermore they generally do not provide an intuitive and interpretable evaluation of the system performance. Kobayashi et al. (2024) explore the extent to which LLMs operate as evaluation models in English GEC. They found that GPT-4 correlates well with human evaluations, but smaller LLMs tend not to do well. Islam and Magnani (2021) introduce Scribendi score metric, an LLM-based method that combines perplexity, a token sort ratio, and the Levenshtein distance. Although Scribendi was found to correlate with human rankings (in English), it has not been evaluated on non-English texts, and, furthermore, it provides a single score for a sentence.

In contrast, we use an LLM to emulate human references, which allows us to compute a more intuitive and interpretable evaluation of model performance in terms of precision, recall, and F-score. Furthermore, our study assesses the validity of the metric on 4 benchmarks in 3 languages.

Language	Dataset	Train	Dev
English (EN)	BEA	628K	63K
English (EN)	CoNLL	1M	-
Czech (CZ)	GECCC	818K	137K
German (DE)	Falko-Merlin	305K	39K
Spanish (ES)	COWS-L2H	132K	18K
Romanian (RO)	RONACC	79K	17K
Russian (RU)	RULEC	4,980	2,500
Russian (RU)	Lang8	-	1,968
Ukrainian (UA)	UA-GEC	32K	1,506
Arabic (AR⋆)	QALB	1M	24K
Italian (IT⋆)	Merlin	71K	9K
Latvian (LV⋆)	LaVA	121K	15K
Slovene (SL⋆)	Solar-EvalH	4.5K	27K
Swedish (SV⋆)	SweLL-gold	110K	12K

Table 6: Gold GEC datasets (training and development data). See Table 1 for statistics on the evaluation benchmarks. Datasets marked with a ⋆ are essay-level, the rest are sentence-level.

B GEC Benchmarks

The following benchmarks are used: English CoNLL-14 (Ng et al., 2014) and BEA W&I+LOCNESS (Bryant et al., 2019), the development partition; Arabic QALB, the learner partition (Rozovskaya et al., 2015); Czech GECCC (Náplava et al., 2022); German Falko-Merlin (Boyd, 2018); Romanian RONACC (Cotet et al., 2020); Russian RULEC (Alsufieva et al., 2012; Rozovskaya and Roth, 2019) and RU-Lang8 (Trinh and Rozovskaya, 2021); Spanish COWS-L2H (Davidson et al., 2020); Ukrainian UA-GEC (Syvokon et al., 2023), Italian, Slovene, Latvian, and Swedish (Masciolini et al., 2025a,b).

C Supervised Approaches to GEC

Supervised Approaches to GEC Supervised approaches to GEC can be broken down into sequence-to-sequence (seq2seq) (Chollampatt and Ng, 2018; Yuan and Briscoe, 2016; Grundkiewicz et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019; Kiyono et al., 2019; Zhao et al., 2019; Ji et al., 2017; Katsumata and Komachi., 2019; Xie et al., 2018; Sorokin, 2022; Rothe et al., 2021), and sequence-to-editing (seq2edits) (Omelianchuk et al., 2020; Awasthi et al., 2019; Li and Shi, 2021; Tarnavskyi et al., 2022). In the seq2seq approach, GEC is cast as a machine translation task with the erroneous sentences treated as the source and corrected

sentences treated as the target. In the seq2edits approach, GEC systems produce explicit text changes, called edits. Both approaches achieve state-of-the-art performance on English GEC that has plenty of gold training data. The edit-based framework (e.g., GECToR Omelianchuk et al. (2020)) was shown to be competitive on English, however, attempts to use it with other languages proved to be less successful (Syvokon and Romanyshyn, 2023), due to the fact that the approach requires language-specific knowledge to develop rules (Bryant et al., 2023). Recent works also achieve strong results with edit ranking (Sorokin, 2022) or ensembling (Omelianchuk et al., 2024). For an overview of approaches and methods in GEC, please see Bryant et al. (2023).

Table 7 lists the results of the top-performing supervised model(s) for each benchmark. The best models for English use ensembling techniques and combine 7 best single models following (Qorib et al., 2024; Omelianchuk et al., 2024).

For non-English GEC, the seq2seq framework has shown state-of-the-art performance (Syvokon et al., 2023; Zhang et al., 2022; Rothe et al., 2021). Seq2seq models are typically *pre-trained* on monolingual data where the source side has been corrupted with artificial noise. The pre-trained model is then further finetuned on gold-labeled data for the target language. Pre-trained language models can be used as a starting point (Kaneko et al., 2020; Malmi et al., 2019; Omelianchuk et al., 2020; Katsumata and Komachi, 2020). Rothe et al. (2021) make use of mT5 in a multilingual setting. Stahlberg and Kumar (2024) finetune mT5 on the monolingual data with synthetic spelling errors and on gold GEC data for each target language (Stahlberg and Kumar, 2024). Their approach produced SOTA results for Romanian, German, and Spanish. Palma Gomez et al. (2023) obtained SOTA results on Ukrainian following (Stahlberg and Kumar, 2024) and Palma Gomez and Rozovskaya (2024) adopt their approach for Russian but generate synthetic data based on morphological transformations instead. Náplava et al. (2022) present SOTA results in Czech by training a seq2seq model from scratch on synthetic data and finetuning it on gold data. The Arabic SOTA results are achieved by finetuning AraBART Alhafni et al. (2023) on the gold data.[14] For the other essay-level benchmarks (Italian, Latvian, and Slovene, and Swedish), we use a multilingual system of Staruch (2025) that finetunes Gemma 2 with QLoRA adapters on the gold data from multiple languages. This system scored first in the multilingual GEC shared task (Masciolini et al., 2025a).

Essay-level approaches operate on longer contexts (300-1000 tokens), whereas sentence-level context length is typically less than 200.[15]

D Details about the Prompts

We use the following zero-shot prompt, when querying GPT-4, Llama, and Claude:

`Provide a grammatical correction for the following text indicated by <input> ERROR </input> tag, making only necessary changes. If the input text is already correct, return it unchanged. Output the corrected version directly without any comments and explanations. Remember to format your corrected output with the tag <output> Your Corrected Version </output>. Please start: <input> ERROR </input>"`

We use the temperature of 0.1. We found that the overall results are quite consistent between the runs. The following information about the *role* is provided to the model (where LANGUAGE is replaced with the appropriate language):

`You are a LANGUAGE grammatical error correction tool that can identify and correct grammatical errors in a text.`

E Performance of Supervised State-of-the-Art Models

Table 8 reports performance of supervised SOTA models by language and benchmark, based on earlier studies. We show multiple top-performing models, when available.

F Evaluation with Closest Golds

Example comparing evaluation using reference golds and closest golds In Table 9 we show an example (in English) that shows two references (ref. 1 and ref. 2). Ref. 1 is the standard fixed reference (created by correcting the input sentence) and ref. 2 is generated by correcting the system hypothesis.

[14] Among all non-English benchmarks, Arabic has the largest training gold data – over 1 million tokens.

[15] Context length is defined in terms of tokens (after LLM tokenization). The length of 200 for sentence-level models is to accommodate longer sentences occurring in 1-2% of inputs.

Dataset	System	P	R	$F_{0.5}$
EN (BEA (dev))	Omelianchuk et al. (2024) ens majority-voting (best 7)	71.7	42.2	62.9
	Qorib and Ng (2023) ens best	-	-	63.4
	Omelianchuk et al. (2024) GPT-4-ZS single	42.5	45.0	43.0
	Omelianchuk et al. (2024) T5-11B single	60.9	51.1	58.6
	Omelianchuk et al. (2024) GECToR-2024 single	64.6	37.2	56.3
EN (CoNLL)	Qorib and Ng (2023) ens best	79.6	49.9	71.1
	Omelianchuk et al. (2024) ens majority-voting (best 7)	83.7	45.7	71.8
	Omelianchuk et al. (2024) ens GRECO-rank-w (best 7)	81.6	49.3	72.1
	Omelianchuk et al. (2024) GPT-4-ZS	59.0	55.4	58.2
	Omelianchuk et al. (2024) single GECToR-2024	75.0	44.7	66.0
CZ (GECCC)	Náplava et al. (2022)	-	-	73.0
DE (Falko-Merlin)	Rothe et al. (2021)	-	-	76.0
	Náplava et al. (2022)	-	-	73.7
	Stahlberg and Kumar (2024) mt5-base FT	-	-	70.5
	Stahlberg and Kumar (2024) mt5-xxl FT	-	-	75.5
	Luhtaru et al. (2024) Llama-2-7B FT	79.1	68.7	76.8
ES (COWS-L2H)	Stahlberg and Kumar (2024) mt5-base FT	-	-	50.1
	Stahlberg and Kumar (2024) mt5-xxl FT	-	-	58.9
	Flachs et al. (2021)	-	-	57.3
	Kementchedjhieva and Søgaard (2023)	-	-	55.2
RO (RONACC)	Stahlberg and Kumar (2024) mt5-base FT	-	-	68.1
	Stahlberg and Kumar (2024) mt5-xxl FT	-	-	75.5
	Kementchedjhieva and Søgaard (2023)	-	-	68.6
	Niculescu et al. (2021)	-	-	69.0
RU (RULEC, 1 ref.)	Sorokin (2022) ru-GPT-large edits, 'combined'	73.5	27.3	55.0
	Náplava and Straka (2019) Transformer	63.3	27.5	50.2
	Rothe et al. (2021) mT5-xxl	-	-	51.6
RU (RULEC, 3 refs.)	Palma Gomez and Rozovskaya (2024) mT5-large	76.7	39.9	64.8
RU (Lang8)	Palma Gomez and Rozovskaya (2024) mT5-large	71.6	40.5	62.1
UA (UA-GEC (dev))	Palma Gomez et al. (2023)	72.1	47.9	65.5
UA (UA-test)	Bondarenko et al. (2023)	79.1	43.9	68.2
	Palma Gomez et al. (2023) mt5-large	73.2	53.2	68.1
	Luhtaru et al. (2024) Llama-2-7B FT	82.0	53.4	74.1
AR⋆ (QALB)	Alhafni et al. (2023) AraBART finetuned best	69.3	43.9	62.5
IT⋆ (Merlin)	Staruch (2025) gemma2-9b-it with QLoRA adapters	68.1	58.6	65.9
LV⋆ (LaVA)	Staruch (2025) gemma2-9b-it with QLoRA adapters	81.3	78.2	80.6
SL⋆ (Solar-EvalH)	Staruch (2025) gemma2-9b-it with QLoRA adapters	58.4	34.2	51.2
SV⋆ (SweLL-gold)	Staruch (2025) gemma2-9b-it with QLoRA adapters	58.7	50.7	56.9

Table 7: Performance of top-performing supervised models by benchmark using standard references (RGs). Results are from the published literature. FT stands for finetuning. ZS stands for zero-shot. RULEC, 3 refs. refers to the same benchmark as RULEC 1., but includes an enhanced set of 3 references (Palma Gomez and Rozovskaya, 2024) vs. 1 gold reference in the original RULEC. Datasets marked with a ⋆ are essay-level, the rest are sentence-level.

Dataset	System	P	R	$F_{0.5}$
EN (CoNLL)	Katinskaia and Yangarber (2024) GPT-3.5 zero-shot	55.8	58.5	56.3
	Fang et al. (2023) ChatGPT zero-shot	50.2	59.0	51.7
	Davis et al. (2024) GPT-3.5	-	-	57.2
	Loem et al. (2023) GPT-3 zero-shot	-	-	56.1
	Loem et al. (2023) GPT-3 16-shot	-	-	57.1
	Fang et al. (2023) ChatGPT zero-shot with CoT			51.7
	GPT-4 (this work)	57.7	58.8	57.9
DE (Falko-Merlin)	Katinskaia and Yangarber (2024) GPT-3.5 zero shot	63.5	63.2	63.4
	GPT-4 (this work)	66.4	66.8	66.5
ES (COWS-L2H)	Katinskaia and Yangarber (2024)	28.1	40.8	29.9
	GPT-4 (this work)	55.7	42.5	52.1
RU (RULEC, 1 ref.)	Katinskaia and Yangarber (2024) GPT3.5 zero-shot	29.2	51.5	32.0
	Katinskaia and Yangarber (2024) GPT-3.5 rerank	39.5	43.7	40.3
	GPT-4 (this work)	47.4	46.7	47.2

Table 8: Comparison to earlier published results of LLM prompting.

Note that the scores (P, R, $F_{0.5}$) depend on the *overlap* between a reference and system hypothesis. Evaluation with closest golds attempts to generate a reference that is as close as possible to the system hypothesis (by producing a reference relative to system output, and not relative to the original sentence). Thus, evaluation with CGs provides a more realistic evaluation of system performance. Note that the overlap (number of correct edits) is larger for ref. 2, and thus the F0.5 score against ref. 2 is higher (91.0) vs. F0.5 score against ref. 1 (50.0). Although the evaluation framework that we described above computes precision, recall, and F-score for each reference independently, edits from multiple references can be combined for a more accurate evaluation (CLEME, Ye et al. (2023)). However, while this may help, it still does not solve the issue of performance underestimation since most GEC benchmarks that we use contain 1 reference per single text.

G Annotation Guidelines for Creating Human Closest-Gold References

We present the outputs from a SOTA model and from GPT-4 to a human annotator together with the original sentence. For annotation, we select SOTA models from Table 7 for which the outputs are publicly available. The following supervised models are used: EN (BEA): Omelianchuk et al. (2024); EN (CoNLL): Qorib and Ng (2023); RU (RULEC): Palma Gomez and Rozovskaya (2024)); AR (QALB): Alhafni et al. (2023). The annotator is asked to correct the remaining errors in the sentence, with a focus on maintaining the original meaning and ensuring the output is grammatical.

Input	The settings are very reallistic and the actors had a great performance.
System hypo	The settings are very realistic and the actors had great performance.
System edits	reallistic → realistic; had a great → had great
	Evaluation against original gold (RG)
Ref. 1 (RG)	The settings are very realistic and the actors gave a great performance.
Gold edits (RG)	(1) reallistic → realistic; (2) had → gave
Correct edits (RG)	(1) reallistic → realistic
Performance against RG	$P = 50.0$; $R = 50.0$; $F_{0.5} = 50.0$
	Evaluation against closest gold (CG)
Ref. 2 (CG)	The settings are very realistic and the actors had great performances .
Gold edits (CG)	(1) reallistic → realistic; (2) had great → had a great; (3) performance → performances
Performance against CG	$P = 100.0$; $R = 66.0$; $F_{0.5} =$ **91.0**

Table 9: Evaluation with a fixed reference (RG) and a closest-gold reference (CG).

Dataset	System	Ref.	P	R	$F_{0.5}$
EN (BEA)	SOTA	RG	66.0	38.7	57.8
	SOTA	CG	89.5	48.7	76.4
	GPT-4	RG	37.1	45.7	38.5
	GPT-4	CG	82.5	67.4	79.0
EN (CoNLL)	SOTA	RG	82.9	48.1	72.4
	SOTA	CG	92.1	51.3	79.5
	GPT-4	RG	58.7	56.2	58.2
	GPT-4	CG	89.3	72.8	85.4
RU (RULEC)	SOTA	RG	66.5	44.6	60.6
	SOTA	CG	89.9	61.1	82.2
	GPT-4	RG	65.7	65.3	65.6
	GPT-4	CG	91.3	85.0	90.0
AR* (QALB)	SOTA	RG	69.5	42.2	61.5
	SOTA	CG	79.4	49.2	70.7
	GPT-4	RG	50.6	26.8	43.0
	GPT-4	CG	84.0	63.2	78.8

Table 10: Evaluation with human closest golds (on a subset) by benchmark. The following supervised models (SOTA) are used: EN (CoNLL): Qorib and Ng (2023); EN (BEA): Omelianchuk et al. (2024); RU (RULEC): Palma Gomez and Rozovskaya (2024)); AR (QALB): Alhafni et al. (2023).

Dataset	System	Ref.	P	R	$F_{0.5}$
EN (BEA)	SOTA	RG	66.0	38.7	57.8
	SOTA	CG-H	89.5	48.7	76.6
	SOTA	CG-C	95.2	51.3	81.3
	SOTA	CG-L	93.8	51.4	80.5
	GPT-4	RG	37.1	45.7	38.5
	GPT-4	CG-H	82.5	67.4	79.0
	GPT-4	CG-C	94.6	74.0	89.6
	GPT-4	CG-L	92.5	69.6	86.8
EN (CoNLL)	SOTA	RG	82.9	48.1	72.4
	SOTA	CG-H	92.1	51.3	79.5
	SOTA	CG-C	94.4	51.8	81.1
	SOTA	CG-L	96.2	55.1	83.7
	GPT-4	RG	58.7	56.2	58.2
	GPT-4	CG-H	89.3	72.8	85.4
	GPT-4	CG-C	96.7	72.9	90.8
	GPT-4	CG-L	96.0	77.1	91.5
RU (RULEC)	SOTA	RG	66.5	44.6	60.6
	SOTA	CG-H	89.9	61.1	82.2
	SOTA	CG-C	92.4	61.3	83.9
	SOTA	CG-L	93.5	59.2	83.8
	GPT-4	RG	65.7	65.3	65.6
	GPT-4	CG-H	91.3	85.0	90.0
	GPT-4	CG-C	96.5	79.6	92.6
	GPT-4	CG-L	96.2	79.1	92.2
AR* (QALB)	SOTA	RG	69.5	42.2	61.5
	SOTA	CG-H	79.4	49.2	70.7
	SOTA	CG-C	74.4	45.7	66.1
	SOTA	CG-L	90.7	72.8	86.4
	GPT-4	RG	50.6	26.8	43.0
	GPT-4	CG-H	84.0	63.2	78.8
	GPT-4	CG-C	90.8	65.0	84.2
	GPT-4	CG-L	97.4	88.1	95.4

Table 11: Evaluation with closest golds produced with human annotators (CG-H) vs. automatic CGs produced with Claude (CG-C) and LLama (CG-L) (on a subset by the benchmark). The following supervised (SOTA) models are used: English (CoNLL): (Qorib and Ng, 2023) ; English (BEA): (Omelianchuk et al., 2024); Arabic: (Alhafni et al., 2023); Russian (RULEC): (Palma Gomez and Rozovskaya, 2024).

Annotation Instructions

You are given X text snippets/sentences in language Y (Y is English, Arabic, or Russian) that may contain grammatical errors. These snippets were corrected by two different automated systems. The total number of text snippets to be corrected is $2 * X$ (2 system outputs per snippet). X is set between 150-200 for Russian and English and and 30 for Arabic. Annotation Instructions: The provided snippets need to be checked for:
(1) Is the sentence grammatical? If not, correct all remaining errors to make sure it is grammatical (try to make as few changes as possible but ensure the resulting sentence is well-formed).
(2) Check that the automated system did not introduce other errors and/or change the meaning of the original sentence. If the meaning of the original sentence has been changed, the corrected sentence should be modified to keep the meaning of the original sentence.
The format of the file is as follows: Each sentence appears three times (line 1 – original sentence; line 2 – system corrected sentence; line 3 – same as line 2 but preceded by ***) You should not modify line 1 and line 2. Only line 3 (that starts with ***) needs to be modified as above.
Do not correct punctuation that is written separately from other words. Also, when correcting, simply delete the incorrect word(s) and replace them with the correct one(s).

H mT5-Large Models

We adopt the approach of Rothe et al. (2021) and Stahlberg and Kumar (2024) and make use of mT5 (Xue et al., 2021). mT5 has been pre-trained on mC4 corpus, covering 101 languages (Xue et al., 2021). Following Palma Gomez et al. (2023), we first pre-trained on target language monolingual data with synthetic spelling errors. In the second step, we finetune on the gold data for the target language. We use mt5-Large, with 1.2B parameters. Results are shown in Table 14.

Hyperparameters Experiments are performed on four A40 48GB GPUs. Table 13 shows training times per model and per epoch on the synthetic data (2M sentence pairs). Finetuning on gold data is fast due to the small sizes of the finetuning sets. We use 2 seeds with each model, and report results averaged over two runs. Hyperparameter values are shown in Table 12.

I Analysis of Over-Corrections

Table 15 shows results of the manual analysis of the instances of over-correction in SOTA and GPT-4 outputs in a sample of the Russian RULEC dataset (see Section 5).

J Model Self-Bias

To evaluate model self-bias, we first run Llama and Claude on the original inputs, and then generate CGs with both Claude and Llama on all the outputs. We first compare the rankings of the four systems by language: SOTA, GPT-4, Llama, and Claude, using RGs. Then we evaluate the four systems using Claude and Llama CGs. Results are shown in Figure 8. While Claude and Llama agree in the rankings of SOTA and GPT-4 for almost all benchmarks, the LLMs are clearly biased towards their own outputs. Specifically, Claude outputs are scored higher than those by Llama when using Claude CGs, whereas Llama CGs suggest that Llama may be as good or better than those of Claude. Averaged results over the 14 benchamarks show that Claude substantially outperforms Llama, whereas Llama CGs indiciate a reverse relation (albei with a smaller gap). This suggest model self-bias and we advise against using the same LLM for GEC and for evaluation of GEC models.

Hyperparameter	Value
Dropout	0.1
Learning rate	$1e-4$
Optimizer	Adam
Max epochs	5 (20)
Input/output lengths	128
Seeds	42 (1,42)

Table 12: Hyperparameter settings for mT5-Large models pre-training on synthetic data and finetuning on gold data. The seeds and the number of epochs are shown separately for the pre-training and the finetuning stages (in parentheses).

GPU	Training time
$A40 \times 1$	36hrs

Table 13: Training times *per epoch* for the pre-training stage on a single GPU (on synthetic data).

Performance ($F_{0.5}$)			
CZ	DE	ES	RO
72.0	73.5	57.8	72.1

Table 14: mT5-Large models pre-trained on 2M. sentence pairs with synthetic errors (spelling) and finetuned on target language gold data.

	Fluency over-correction				Incorrect over-editing			
Model	Word order	Lex.	Other	Total	Deletion	Lex.	Other	Total
SOTA	0	4	0	4	3	7	6	16
GPT-4	4	17	8	29	1	7	4	12

Table 15: Number of over-correction and over-editing occurrences by GPT-4 and the supervised model (SOTA) in the RULEC subset. *Fluency over-correction* refers to the cases where the meaning is preserved but the model replaces a word or expression that is acceptable with another one that is as good or better. *Incorrect over-editing* denotes cases that result in the change of meaning, when an acceptable word or expression is replaced.

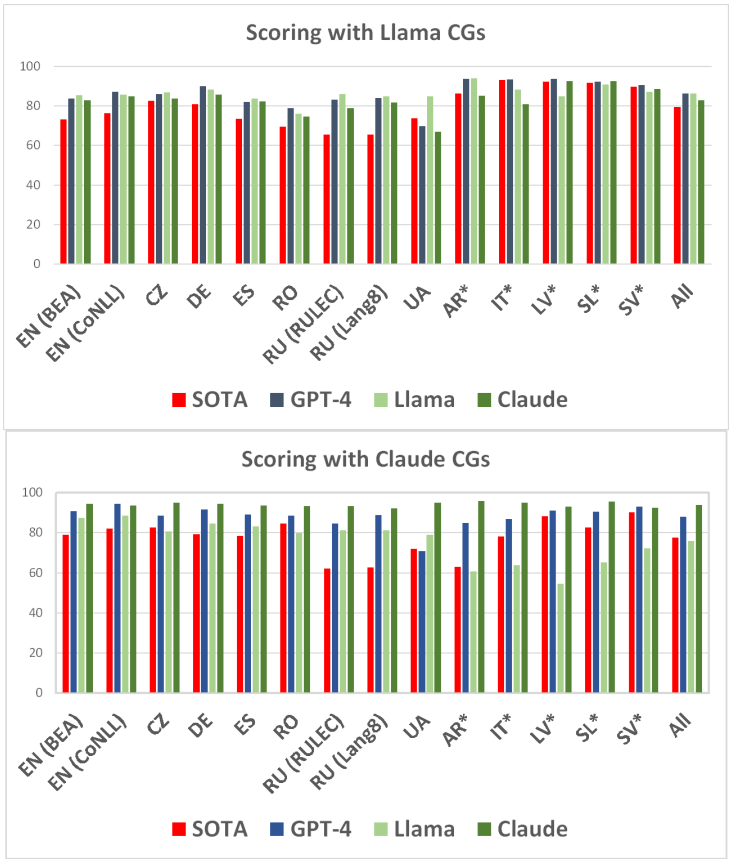

Figure 8: Scoring SOTA, GPT-4, Claude and Llama with Claude CGs (top) and Llama CGs (bottom). Results by the benchmark and averaged over all datasets.