EAR – Explain and Rank: a criteria-based Reinforcement Learning pipeline for GEC Feedback Comment Generation

Anonymous ACL submission

Abstract

We propose EAR – a reinforcement learning (RL) pipeline for generating explanations for Grammatical Error Correction (GEC). By developing a multi-criteria, prompt-based ranking reward, we implement a multistage RL process, comprising teacher distillation, rejection sampling and pairwise RL, applied to the Qwen3-4B-Instruct model. Our results show that the fine-tuned model outperforms both the teacher model and expert human annotations with a 60/40 win rate. Furthermore, pairwise RL enables the generator to acquire capabilities absent in the teacher model, specifically advanced analogical reasoning and in-depth semantic analysis without using any gold explanations or annotated datasets. We plan to make our code and data freely available.

1 Introduction

The task of Grammatical Error Correction (GEC), i.e. of removing all errors in a sentence, has been extensively covered in various papers throughout the years (Chodorow et al., 2007; Felice et al., 2014; Náplava and Straka, 2019; Omelianchuk et al., 2020; Loem et al., 2023; Ye et al., 2025b), especially for the English language. According to Omelianchuk et al. (2024), single-model GEC systems have reached a performance plateau . The authors argue that further progress depends on data quality rather than architectural scaling, suggesting that GEC for English is largely a solved task.

However, since GEC primarily serves L2 learners, it remains incomplete without pedagogical feedback. Corrected text alone lacks the explanatory power necessary for students to generalize rules and avoid recurring errors. Consequently, the GEC task remains incomplete until models can bridge the gap between error detection and rule generalization through clear, high-quality explanations.

This research evaluates the capacity of Large Language Models (LLMs) to provide high-quality feedback for Grammatical Error Correction (GEC). A pilot experiment using Qwen3-30B-A3B[1] on the XGEC corpus (Kaneko and Okazaki, 2024) revealed significant limitations: one-third of the generated explanations exhibited serious flaws in factual accuracy and completeness. These results indicate that even frontier open-weight models struggle to provide reliable learning feedback out-of-the-box, necessitating targeted optimization.

Training such models is hindered by a scarcity of high-quality annotated data; to our knowledge, XGEC is the only public dataset providing comprehensive explanations across diverse error types. To address this, we employ a reward-based framework that iteratively optimizes the generator via Reinforcement Learning (RL). While this pipeline is well-established for aligning LLMs in domains like mathematics or programming, its application to language learning remains largely unexplored.

In this paper, we introduce **EAR** – *Explain and Rank* pipeline, adapting the RL approach to the GEC domain by designing a ranking reward optimized through offline RL techniques. We demonstrate that explanations from our fine-tuned 4B model outperform both the original Qwen3-30B-A3B outputs and expert human annotations from the XGEC dataset, as confirmed by both manual and automatic evaluation.

2 Task definition

We follow Nagata (2019); Nagata et al. (2021) and define Feedback Comment Generation for language learners as the task of generating, given a text and a span, an explanatory note that will help improve learners' writing skills. However, we extend it by several points.

[1] Qwen3-30B-A3B-Thinking-2507-FP8 (Team, 2025); hereafter Qwen3-30B-A3B.

Similar to Kaneko and Okazaki (2024), i) we provide the model not only with a text and an erroneous span, but also with a corresponding correction, assuring that the model would explain the required edit and avoid inventing anything; ii) we consider a broader range of errors, not restricting them to preposition use; iii) the sentences may contain multiple mistakes and the system must target them all, generating explanations for each edit. Therefore, our definition of the task is to generate a list of explanations, given an incorrect sentence, fully edited sentence and all of the edits – error-correction pairs. The preliminary step of our research consisted of comparing and contrasting human and model explanations to identify the features of a 'good' explanation and choose the optimal prompt for their generation. XGEC NUCLE subset[2] (Kaneko and Okazaki, 2024) served as human-annotated data, whereas Qwen3-30B-A3B was used to produce model data.

2.1 Criteria

We define a high quality explanation as the explanation that satisfies the following criteria:

- **Reliability**. There are no hallucinations, i.e. facts that contradict either the context or rules of English grammar. This includes
  1. mentioning words or sequences that did not occur in the sentence (see Fig. 2a).
  2. referring to nonexistent standards, misleading guidelines and generalizations (see Fig. 2c and d). This often includes over-rigidity in rule application that ignores context-dependent acceptability.
  3. treating a dependent edit as a fundamental mistake, overlooking its necessity being conditioned by a previous correction (Fig. 2b).
- **Completeness** The explanation provides a comprehensive list of reasons for a given correction (Fig. 2e), supplies it with contextual evidence from the sentence (Fig. 2f) and refers to specific rules of the English grammar that were violated. It also demonstrates how the correction improves the sentence by making a distinction in usage and meaning between the original expression and its edit. In addition, no logical steps may be overlooked: the description of how the rule applies to the sentence must be self-sufficient and leave no room for speculation, e.g. for an insertion of 'the' it is not enough to state that a singular countable noun requires an article, the explanation should also address why 'the' was chosen from the set of determiners – 'a', 'an' and 'the'.
- **Usefulness** There are no sentences consisting fully of empty phrases that do not contribute to the explanation, e.g. *This improves text clarity*.
- **Execution** In order to serve educational purposes, the explanation itself must not contain any grammatical errors and instances of ambiguous or poor phrasing, e.g. *the noun 'citizens' is incorrect in the context of 'their citizens'*.

Although the perfect feedback for a correction would possess all of the features, the most critical criteria to meet are reliability and completeness. As students refer to explanations when they do not understand or know certain rules, it is of most importance not to mislead them by providing unreliable judgements and not to offer them incomplete explanations that would not increase their knowledge and reduce their doubts.

2.2 Prompt

We experimented with different ways of representing GEC-annotation, exploiting either the full .M2-schema, which is standard for the GEC task, or merely listing annotations as `<error> -> <correction>`. The model was prone to confuse the boundaries of erroneous tokens, when given the .M2-annotation, that is why the second format was considered more reliable.

The instruction always included the description of the given data and, optionally, the criteria. We assumed that providing the model with evaluation strategies would align her outputs with human preferences. However, in this case the model tended to overproduce the reasoning content, which exceeded the 10K maximum tokens limit or often ended in subjects not connected with GEC, such as code, math problems or storytelling (see Appendix B.2). Moreover, we did not observe a significant change in the quality of explanations, depending on the appearance of criteria, so we did not include them in the final prompt, reducing the computational costs and inference time. The most robust version of the prompt is given in the Appendix C.1.

[2] https://github.com/kanekomasahiro/gec-explanation

2.3 LLM-as-a-Judge

Following the generation of explanations, an assessment of their quality is required for further training of the models. While manual evaluation is the most reliable option, it has several drawbacks. Firstly, the annotators must possess certain qualifications, such as be English native speakers or have an exam-proven high level of English, and preferably have some teaching experience that would help them assess not only the correctness of explanations, but also their clarity and accessibility. Secondly, evaluation of educational feedback demands thorough and vast reading, especially if there are multiple corrections in a sentence, which is very laborious. Due to these considerations, we employed LLM-as-a-Judge methodology which proved to be useful in many tasks (Zheng et al., 2023; Liu et al., 2025).

2.4 Types of judgements

We examined pointwise and pairwise scoring patterns from Liu et al. (2025) for the automatic evaluation of explanations.

- **Pointwise approach** evaluates how an explanation follows each of the criteria by assigning either 0 or 1.
- **Pairwise approach** compares 2+ responses simultaneously and chooses the best option or ranks them from best to worst.

While the point-wise approach is more common, it proved ineffective for our task, identifying only 2% of failures even in sampling mode. In contrast, the pairwise approach was more discerning, achieving 70% agreement with human judgments in selecting the best samples. Consequently, we adopted the pairwise method for our LLM-as-a-judge pipeline. We refer to Appendix C.2 for the ranking prompt.

3 Reinforcement learning

In this section, we describe the training pipeline of the proposed explanation generator. We follow the standard scheme for RL-based LLM tuning (Touvron et al., 2023; Liu et al., 2024; Guo et al., 2025). Usually, this scheme includes the following stages:

1. Training a reward model on either preference data or scalar quality scores. Usually, this stage yields a scorer that assigns numerical values to model responses, reflecting their overall quality with respect to the task in question.
2. Tuning a medium-size pretrained model on the high-quality generations or reasoning traces from a stronger foundation model, such as DeepSeek (Guo et al., 2025) or Gemini (Comanici et al., 2025) (the *SFT* stage).
3. Sampling multiple outputs from the SFT model and training it on the samples with the highest scores according to the reward model (the *rejection sampling* stage)[3].
4. Sampling and scoring the previous model outputs and training it further using either pairwise (PPO, DPO, etc.) or groupwise RL algorithms. On this stage the model learns to distinguish between superior and mediocre generations of its own.

Adopting this scheme for GEC explanation generation, we faced several challenges.

1. The core problem is the absence of annotated datasets, either in the form of scalar ratings or preference data. Therefore, we resorted to training-free prompt-based LLM rewards.
2. We found the task of reference-free estimation of explanations' quality to be too difficult in zero-shot mode even for modern LLMs. We leverage ranking-based rewards to circumvent this shortcoming (see Section 2.4).
3. Since no verifiable rewards are available for the task of explanation quality estimation, we utilized DPO-like pairwise algorithms that require only preference data.

Summarizing, our RL pipeline contains three stages:

1. **The SFT stage**, where we train the student model on the zero-shot generations from a larger foundation model.
2. **The rejection sampling stage**, where the SFT model is trained on its own generations obtained in the sampling mode.
3. **The pairwise RL stage**, where we further tune the model from Stage 2 on the pairs of "chosen"-"rejected" samples.

3.1 SFT

The first stage of our pipeline is an instance of standard zero-shot distillation: we develop a prompt describing the task in question and ask the larger model to generate edit explanations according to

[3]This stage might be omitted.

this prompt. The exact prompt formulation is provided in Appendix C.1. We apply the teacher generator to sentences that may contain several edits, asking the model to generate explanations for all of them. We filter out the outputs where the number of explanations does not match the number of edits

Actually, instead of training on a single greedy output, we employ *a ranking-based selection mechanism*: the teacher model generates several candidates, and the one with the highest reward score (according to our LLM-as-a-Judge) is chosen as the training target. The exact algorithm of converting ranking reward outputs to scalar scores is described below in Subsection 3.2.

3.2 Rejection sampling

On the second stage, we apply a standard rejection sampling procedure. We sample several candidates from the SFT model and rank them using the ranking reward. As the model generates the output for the whole sentence and the scorer produces rankings for individual errors, an additional postprocessing step is required to convert ranking output to scalar scores. For each error, we assign zero points to the highest ranked hypothesis, one point to the second one, and so on. The hypothesis with the lowest sum of scores is selected as the winner.

The pool of candidate hypotheses consists of the greedy generation and four sampling outputs. If the number of explanations in the hypothesis is not equal to the number of errors, we exclude this sample from comparison.

3.3 Pairwise RL

The final and crucial step of our model tuning is pairwise training on preference data. Again, we use the model from the previous stage to sample five outputs (one greedy and four sampling ones) and pass them through the ranking reward. We say that one hypothesis is preferred over another if for all the edits its explanation is ranked higher or equally to the corresponding explanation from the second hypothesis. So, ranking n hypotheses produces up to $\binom{n}{2}$ selected preference pairs. For each training example, we randomly sample K pairs from the list of selected pairs without repetition, where K is the hyperparameter of the algorithm.

Choosing between different pairwise algorithms, we finally opt for AlphaPO (Gupta et al., 2025) as we found it to be the most stable and effective in training. The relative properties of pairwise RL algorithm are discussed more thoroughly in Subsection 6.2.

Dataset	Usage	Size, sent.
BEA-2019 train	train	23045
BEA-2019 dev	validation	2856
CoNLL-2014	test	550

Table 1: Datasets used in the paper

4 Evaluation

4.1 Data

We train the models on the training part of the W&I-LOCNESS corpus (Bryant et al., 2019) used in the BEA-2019 shared task on Grammatical Error Correction. We utilize it for teacher model distillation on SFT stage and sampling the hypotheses on further RL stages. The development part of this corpus is used as the development corpus, in particular, for prompt evaluation, algorithm comparison and hyperparameter selection. Following the previous work (Kaneko and Okazaki, 2024), we use the CoNLL-2014[4] (Annotator 1) corpus (Ng et al., 2014) as the final test data. The parameters of the corpora are provided in Table 1. Note that we count only sentences that contain at least one edit.

4.2 Models and training

We evaluate our approach using models from Qwen-3 family (Team, 2025). In particular, we use Qwen3-30B-A3B[5] model as our teacher model and initialize training from Qwen3-4B-Instruct-2507[6] checkpoint. The main training hyperparameters are provided in Table 2, other parameters of training configuration are given in Table 8. The specific parameters for pairwise RL are provided in Table 9, their selection is discussed in Appendix F. We implement all training stages using trl (von Werra et al., 2020) library.

5 Results

The proposed method is evaluated using LLM-as-a-Judge approach together with human-based scoring. As was previously noted, we used the BEA-2019 development set for internal validation, while the

[4] https://github.com/kanekomasahiro/gec-explanation

[5] In preliminary experiments we found it to prevail over other open-weight models. We intentionally refrain from using closed-source models for distillation.

[6] https://huggingface.co/Qwen/Qwen/Qwen3-4B-Instruct-2507

Stage	LR	batch size	epochs
SFT	1e-5	16	2
Rejection sampling	1e-6	16	1
Pairwise RL	5e-7	64	3

Table 2: Main hyperparameters of model training (LR – learning rate, epochs – number of passes over training data

CoNLL-2014 subcorpus from Kaneko and Okazaki (2024) was kept for final evaluation of the best selected model. The assessment aims to validate two primary hypotheses:

1. Training effectiveness: whether the fine-tuned model outperforms its teacher (Qwen3-30B-A3B) across automated metrics, thereby confirming the ability to optimize the designed reward.
2. Explanation quality: whether annotators prefer the model-generated explanations over high-quality, human-written references from the CoNLL-2014 subset of the XGEC dataset.

5.1 Automatic evaluation

To verify the first claim, we collect outputs from different models on the validation set and apply the ranking prompt to order them. We prompt the same model (Qwen3-30B-A3B) as during model training. In table 3 we present pairwise win rates and first-place shares across all models. These results show that the SFT model is comparable to its teacher but lies slightly behind. The second stage model does not provide clear improvement over its predecessor. However, the final model definitely outperforms both the models from previous RL stages and the initial teacher model demonstrating the strength of pairwise preference learning.

To account for the performance gap between Qwen3-30B-A3B and top-tier models, and to mitigate self-evaluation bias (avoiding the use of the same model for both reward and assessment), we also perform automatic evaluation using stronger closed-source LLMs, replacing Qwen3-30B-A3B with OpenAI GPT4o (OpenAI, 2023). To prevent data leakage, we use GPT4-based ranking only for final evaluation on the CoNLL-2014 XGEC testset and do not utilize GPT-4o for intermediate judgements.

We use GPT-4o to compare 4 explanations:

- The original explanations from XGEC dataset (Kaneko and Okazaki, 2024) written by human experts;
- Its paraphrased version rewritten in the style of Qwen3-30B-A3B response. We obtain it by prompting Deepseek v3.1 to rewrite the explanation more verbosely, providing a seed response for stylistic guidance (see Appendix C.3 for the prompt). This ensures the evaluator focuses on explanations' content rather than stylistic cues.
- The explanations generated by the teacher Qwen3-30B-A3B model (the baseline);
- The explanation generated by our EAR-4B after pairwise RL stage.

The comparison results are provided in Table 4. It shows that our EAR-4B model outperforms both its teacher and rewritten human expert annotations.

5.2 Human evaluation

We conducted a human evaluation on a random subset of 216 sentences from the CoNLL-2014 XGEC test set. Five assessors (two experts and three students) with linguistic backgrounds and high English proficiency – verified by standardized exams or academic competitions – performed the annotations. All annotators underwent preliminary training, achieving substantial agreement (65-90%) with the lead expert.

Each instance was independently ranked by one expert and one student. For a given sentence and its corrections, assessors ranked four candidate explanation sequences from best to worst (ties permitted). To eliminate position bias, we balanced and shuffled all 24 possible presentation orders across the evaluation set. Further details on the annotation process are provided in Appendix E.

The results of manual evaluation are provided in Table 5. They roughly correspond to GPT4-based evaluation, demonstrating that EAR-4B model is generally preferred both over expert explanations and the answers of its teacher model.

6 Ablation and discussions

6.1 Qualitative analysis

We observe that our EAR-4B model offers more complete responses as compared with other models. First of all, it accompanies its judgements with various analogous examples and avoids simple comparisons of the error and the correction (see Fig. 1a).

Secondly, its explanations rely heavily on the given context and profoundly analyse how edits and the meaning of the sentence impact each other,

Model	av.tokens	total wins	0	1	2	3
0. Qwen3-3A-30B (teacher)	245.5	10.1		52.2	51.0	13.5
1. SFT (stage 1)	257.0	7.3	47.8		54.9	12.0
2. rej. sampling (stage 2)	267.7	8.4	49.0	45.1		12.1
3. pairwise RL (stage 3, EAR-4B)	453.3	77.5	86.5	87.9	88.0	

Table 3: The percentage of total wins and head-to-head winrates (row-vs-column) for different stages of reinforcement learning and the teacher model evaluated on BEA-2019 development set.

Model	total wins	0	1	2	3
0. Expert annotations (raw)	0.2		4.8	4.6	12.2
1. Expert annotations (rewritten)	27.3	95.2		40.7	42.5
2. Qwen3-3A-30B (teacher)	23.4	95.4	59.3		42.3
3. EAR-4B (ours)	49.1	87.8	57.5	57.7	

Table 4: The percentage of total wins and head-to-head winrates (row-vs-column) for the proposed model versus the teacher model and the expert annotations (both raw and rewritten) by GPT-based reward. The evaluation is performed on CoNLL-2014 test set with GPT-4o for LLM-as-a-judge.

Model	av. tokens	total wins	0	1	2	3
0. Expert annotations (raw)	42.1	3.2		9.4	14.6	18.7
1. Expert annotations (rewritten)	127.1	24.0	90.6		44.5	39.9
2. Qwen3-3A-30B (teacher)	159.2	25.5	85.4	54.6		40.8
3. pairwise RL (final model)	288.6	47.3	81.3	60.1	59.2	

Table 5: The percentage of total wins and head-to-head winrates (row-vs-column) for the proposed model versus the teacher model and the expert annotations (both raw and rewritten) by human annotation. The evaluation is performed on CoNLL-2014 test set with GPT-4o for LLM-as-a-judge.

e.g. how the lexical choice error alters the intended meaning, see Fig. 1b. Thirdly, EAR-4B more readily leverages its world knowledge to explain certain errors which brings depth to the feedback about the given constructions and the manner and contexts in which they are used, e.g. in the Fig. 1c.

However, in the model's tendency to explain everything as thoroughly as possible, attempting to supply every statement with an argument, lies a substantial drawback: the model may produce extremely verbose explanations for the simplest errors (e.g. article usage or spelling ones), allowing for useless, speculative or even false claims (see Fig. 1d).

The illustrative example of model generation in interactive model is shown on Figure 3.

6.2 Pairwise RL training details

The task of explanation generation presents a challenge for preference learning: "chosen" and "rejected" explanations often exhibit high surface similarity, differing only in subtle nuances. Effectively capturing these distinctions requires the policy model to deviate substantially from its initial state. However, such significant updates provoke training instability, increasing the risk of degenerate behaviour, e.g. in the form of unnaturally long responses. To cope with this task, we compare several pairwise preference optimization algorithms, including DPO (Rafailov et al., 2023), ORPO (Hong et al., 2024), IPO (Azar et al., 2024), and AlphaPO (Gupta et al., 2025). We also performed extensive hyperparameter search.

The quantitative results of model comparison are provided in Table 6 (only the best hyperparameters for each algorithm) and Table 10. Qualitatively, we make the following observations:

- Both DPO and ORPO fail to capture nuanced distinctions between explanations and show near-random performance.
- IPO effectively learns from preference data if the learning rate is high (1e-5), but suffers from instability and overfitting, eventually causing the model to diverge and produce meaningless repetitive outputs.
- AlphaPO has analogous ability to learn preference distributions, but in contrast with IPO,

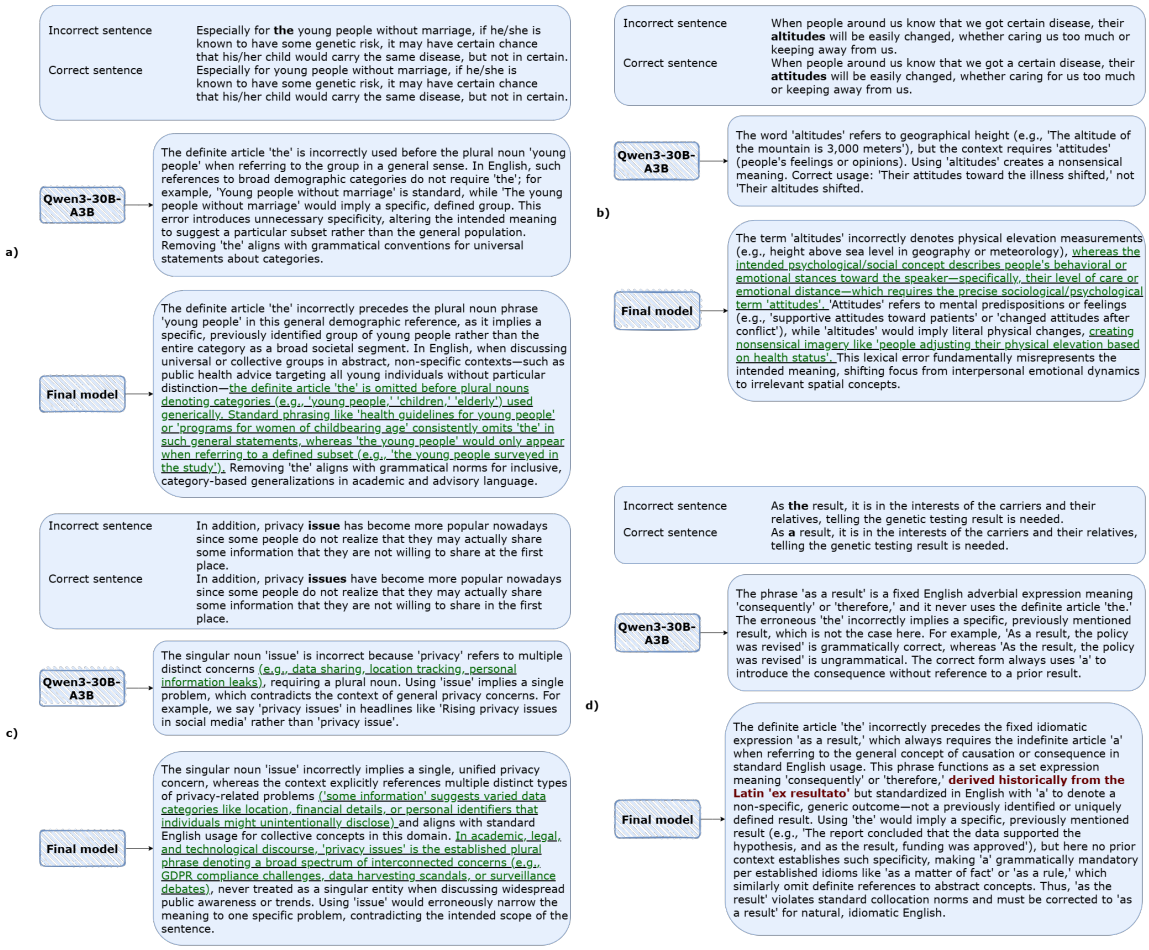

Figure 1: Examples for qualitative comparison of EAR-4B model and other models. We present only two explanations out of 4 for illustrative purposes: according to the human evaluation, in *a*, *b* and *c* the Qwen3-30B-A3B responses were second best options, while in *d* it was ranked first.

longer training doesn't lead to model degeneration and hamper performance. This algorithm significantly increases response length; while this leads to more comprehensive and detailed explanations, it simultaneously elevates the risk of hallucinations.

For a more detailed discussion of pairwise RL algorithms we refer to Appendix F.

6.3 Length bias

Applying pairwise RL and especially AlphaPO leads to substantial answer extension (from 268 tokens to 453 in average). One may assume that the ranking reward merely prefers longer answers without taking their content into account. To verify this statement we perform two sanity checks. First, we consider the sampling outputs and order samples by their length in decreasing order. We calculate the rank frequencies of the winning hypotheses according to the ranking reward. The results are provided in Table 7, it shows that despite notable propensity towards longer inputs the reward might select shorter answers as well implying that explanation content is also taken into account.

In the second experiment, we replace the answers of EAR-4B model with their corrupted versions containing incorrect facts about English grammar but having similar length. Despite their length advantage (222.2 tokens in average vs 127.1/159.2), these corrupted explanations lost decisively to rewritten expert annotations and 30B model outputs on the CoNLL-2014 test set, with win rates of only 1.2% and 2.9%, respectively. The corruption prompt is specified on Fig. 7.

7 Related Work

7.1 Explainable Grammatical Error Correction

Feedback Comment Generation and shared tasks for English regarding it were first introduced in the

Algorithm	LR	β	n_{dpo}	epochs	Acc	L, tok.
Initialization	NA	NA	NA	NA	49.0	257.0
DPO	1e-5	0.1	1	1	51.5	288.9
ORPO	1e-5	0.1	1	1	49.2	283.0
IPO	1e-5	0.1	1	1	62.4	369.5
AlphaPO	5e-7	0.5	3	3	65.2	453.3

Table 6: Pairwise accuracy and response length for different pairwise RL algorithms on validation data. n_{dpo} stands for the number of preference pairs corresponding to a single training example and L to response length.

Rank	1	2	3	4	5
Frequency	32.6	23.6	18.0	14.5	11.3

Table 7: The frequency of length ranks for the winning sampling hypotheses from the Rejection Sampling model on the training data.

papers (Nagata, 2019; Nagata et al., 2021). They defined it as generating a hint or an explanatory note for writing learning for non-native learners of English. Improving their understanding of the rules pertaining to corrections was the key motivation behind the task.

A similar rationale encouraged the authors of Fei et al. (2023). While shared tasks intentionally targeted only preposition use errors because of their frequency and relatively easy explainability, (Fei et al., 2023) increased the number of error types up to 15 and concentrated on evidence words, i.e. causes of errors, instead of generating full explanations in their dataset EXPECT, with further development in Ye et al. (2025b).

To the best of our knowledge, one of the first instances of broader considerations of the task – to generate feedback comments to different types of errors, in a setting where multiple errors in a sentence are possible, is the paper Kaneko and Okazaki (2024). The authors proposed a Prompt Insertion methodology that allowed control over LLMs to consistently cover most of the needed corrections. They also introduced the XGEC dataset, consisting of well-known English GEC datasets annotated with explanations. Apart from English, there are studies on GEC explanations for Turkish (Gebeşçe and Şahin, 2025) and Chinese (Ye et al., 2025a).

Prompt-based Grammatical Error Explanation was also addressed by Song et al. (2024). In more general context of essay scoring criterion-based analysis was performed in Wang et al. (2025), while the applicability of pairwise RL for general text editing was investigated by Cao et al. (2025).

7.2 Feedback generation in other contexts

AI-driven feedback in education is a rapidly evolving field. Most systems leverage closed-source LLMs via specialized prompting including Woodrow et al. (2024); Lohr et al. (2025); Koutcheme et al. (2025b) for coding and Tonga et al. (2025); McNichols et al. (2024); Baral et al. (2024) for mathematics. Advanced prompting techniques include task decomposition (Phung et al., 2023) and self-verification (Worden et al., 2025).

Due to the scarcity of annotated data, fine-tuning LLMs for feedback remains less common. Early efforts include Baral et al. (2024), who fine-tuned Llama-2 on real-world teacher feedback data, and Solano et al. (2025) distilling GPT-4 responses obtained using a fine-grained structured prompt. More recent studies employ pairwise RL algorithms like DPO to refine feedback quality using either GPT-based judgments (Scarlatos et al., 2024, 2025) or human preference data (Woodrow et al., 2025). Additionally, Koutcheme et al. (2025a) uses repair-based rewards for programming feedback.

8 Conclusion and future work

We developed a multistage reinforcement learning pipeline to train a GEC explanation generator without relying on gold-standard data. Our EAR-4B model outperforms both expert annotations and large-scale LLM prompting in explanation quality. Notably, RL induces emergent capabilities in the generator, including analogical reasoning and a deeper integration of sentence context and world knowledge.

We plan to extend our research in several directions. First, we aim to enhance annotation flexibility, enabling the model to adapt style and verbosity to specific user needs. Second, we plan to integrate explanation generation directly with the correction process, as we hypothesize this synergy can improve the reliability of LLM-based GEC.

Limitations

First, our method inherits drawbacks and limitations from both reinforcement learning and LLM-as-a-judge approaches. Regarding RL limitations, our approach is susceptible to "reward gaming", wherein the model prioritizes surface-level features (e.g., verbosity) over the substantive quality of explanations. Since our reward is training-free, its preferences might depend on subtle variations in prompt formulation. Another source of bias might be the underlying language model itself and the data used for its (pre-)training. Given that pairwise RL algorithms are prone to amplify spurious features correlated with the target preferences, these inherent biases may be further exacerbated during the reinforcement learning process.

Ethical considerations

Our work extensively utilizes the data from LLM that may accidentally produce harmful or malicious output. We do not observe any cases of such behaviour during our study but such responses might arise from training data or finetuning datasets that we have no control upon. This study does not cover the pedagogical or social effects of AI-driven feedback in practice. Further investigation into the risks and benefits for learners is required before these models are integrated into educational tools.

All the students who participated in the creation of the dataset earned credit hours as a result. The students were informed about the goals of the work and gave their content for research publication.

Acknowledgments

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Sami Baral, Eamon Worden, Wen-Chiang Lim, Zhuang Luo, Christopher Santorelli, Ashish Gurung, and Neil Heffernan. 2024. Automated feedback in math education: A comparative analysis of llms for open-ended responses. *arXiv preprint arXiv:2411.08910*.

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

Hannan Cao, Hai Ye, and Hwee Tou Ng. 2025. Rationalize and align: Enhancing writing assistance with rationale via self-training for improved alignment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26967–26982, Vienna, Austria. Association for Computational Linguistics.

Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the fourth ACL-SIGSEM workshop on prepositions*, pages 25–30.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.

Ali Gebeşçe and Gözde Gül Şahin. 2025. GECTurk WEB: An explainable online platform for Turkish grammatical error detection and correction. In *Proceedings of the 31st International Conference on Computational Linguistics: System Demonstrations*, pages 163–173, Abu Dhabi, UAE. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Aman Gupta, Shao Tang, Qingquan Song, Sirou Zhu, Jiwoo Hong, Ankan Saha, Viral Gupta, Noah Lee, Eunki Kim, Siyu Zhu, and 1 others. 2025. Alphapo: Reward shape matters for llm alignment. In *International Conference on Machine Learning*, pages 21369–21394. PMLR.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without

reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Masahiro Kaneko and Naoaki Okazaki. 2024. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3955–3961, Torino, Italia. ELRA and ICCL.

Charles Koutcheme, Nicola Dainese, and Arto Hellas. 2025a. Direct repair optimization: Training small language models for educational program repair improves feedback. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 564–581, Vienna, Austria. Association for Computational Linguistics.

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, Syed Ashraf, and Paul Denny. 2025b. Evaluating language models for generating and judging programming feedback. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 624–630.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling. *Preprint*, arXiv:2504.02495.

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. *arXiv preprint arXiv:2305.18156*.

Dominic Lohr, Hieke Keuning, and Natalie Kiesler. 2025. You're (not) my type- can llms generate feedback of specific types for introductory programming tasks? *Journal of Computer Assisted Learning*, 41(1):e13107. E13107 JCAL-24-434.R1.

Hunter McNichols, Jaewook Lee, Stephen Fancsali, Steve Ritter, and Andrew Lan. 2024. Can large language models replicate its feedback on open-ended math questions? *arXiv preprint arXiv:2405.06414*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.

Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. Shared task on feedback comment generation for language learners. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

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.

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

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – Grammatical Error Correction: Tag, Not Rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170.

Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. *arXiv preprint arXiv:2404.14914*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tung Phung, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdarm, Adish Singla, and Gustavo Soares. 2023. Generating high-precision feedback for programming syntax errors using large language models. *International Educational Data Mining Society*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Alexander Scarlatos, Ryan S Baker, and Andrew Lan. 2025. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th international learning analytics and knowledge conference*, pages 249–259.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the validity of

automatically generated feedback via reinforcement learning. In *International Conference on Artificial Intelligence in Education*, pages 280–294. Springer.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Lorenzo Lee Solano, Charles Koutcheme, Juho Leinonen, Alexandra Vassar, and Jake Renzella. 2025. Narrowing the gap: supervised fine-tuning of open-source llms as a viable alternative to proprietary models for pedagogical tools. *arXiv preprint arXiv:2507.05305*.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. GEE! grammar error explanation with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Junior Cedric Tonga, Benjamin Clement, and Pierre-Yves Oudeyer. 2025. Automatic generation of question hints for mathematics problems using large language models in educational technology. In *Large Foundation Models for Educational Assessment*, pages 61–102. PMLR.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning.

Zhengxiang Wang, Veronika Makarova, Zhi Li, Jordan Kodner, and Owen Rambow. 2025. LLMs can perform multi-dimensional analytic writing assessments: A case study of L2 graduate-level academic English writing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8637–8663, Vienna, Austria. Association for Computational Linguistics.

Juliette Woodrow, Sanmi Koyejo, and Chris Piech. 2025. Improving generative ai student feedback: Direct preference optimization with teachers in the loop. *International Educational Data Mining Society*.

Juliette Woodrow, Ali Malik, and Chris Piech. 2024. Ai teaches the art of elegant coding: Timely, fair, and helpful style feedback in a global course. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2024, page 1442–1448, New York, NY, USA. Association for Computing Machinery.

Eamon Worden, Kirk Vanacore, Aaron Haim, and Neil Heffernan. 2025. Scaling effective ai-generated explanations for middle school mathematics in online learning platforms. In *Proceedings of the Twelfth ACM Conference on Learning@ Scale*, pages 40–49.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, and Wenhao Jiang. 2025a. Excgec: A benchmark for edit-wise explainable chinese grammatical error correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25678–25686.

Jingheng Ye, Shang Qin, Yinghui Li, Hai-Tao Zheng, Shen Wang, and Qingsong Wen. 2025b. Corrections meet explanations: A unified framework for explainable grammatical error correction. *arXiv preprint arXiv:2502.15261*.

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

A Reliability and Completeness

We demonstrate examples which do not satisfy either reliability or completeness criteria on the Figure 2.

B Examples

B.1 EAR-4B generations

We present an example of model application on Figure 3.

B.2 Overgeneration

We present an example on the figure 4.

C Prompts

a

Correction	concentraded → concentrated
Model explanation	**This is a spelling error where the 'c' is incorrectly replaced by 'd' after the 'n', violating standard English orthography.** The correct form 'concentrated' follows the standard spelling pattern for verbs ending in '-ate' (e.g., 'concentrate' → 'concentrated'). The misspelling 'concentraded' disrupts readability and is a common typographical mistake, as seen in similar words like 'concentrate' vs. 'concentradate'.

b

Incorrect sentence	This will, if not already, caused problems as there are very limited spaces for us.
Correct sentence	This will, if not already, cause problems as there is very limited space for us.
Correction	are → is
Model explanation	The subject space' (singular, after correction) requires a singular verb. **'are' (plural) was used incorrectly due to subject-verb agreement error.**
Comment	The phrasing of the second sentence suggests that there was inherently a subject-verb agreement error in the sentence, whereas it only occurred after the correction of the noun number spaces → space.

c

Correction	it often results the invention of a new technology → it often results **in** the invention of a new technology
Model explanation	The verb 'results' requires the preposition 'in' when followed by a noun phrase, as in 'results in [outcome].' Omitting 'in' creates a grammatical error, similar to how 'leads to' or **'causes' would require a preposition.** For instance, 'The experiment resulted in a breakthrough' is correct, whereas 'The experiment resulted a breakthrough' is ungrammatical. This error is common in informal speech but must be corrected in formal writing to adhere to standard English syntax.
Comment	Without further details the explanation draws an analogy between the verbs 'result', 'lead' and 'cause', implying they all require a preposition after them. However, the verb 'cause' generally takes a noun phrase as an object, e.g. *The storm has caused several accidents*, and may even be a grammatical substitution of 'result' in the given sentence (for example, in the context where a new technology poses a threat). Therefore, the comparison gives a misleading advice on the verb usage.

d

Correction	[A good example is Japan, where there are a lot of tall condominiums despite the large number of earthquakes happening] in there → there
Model explanation	The word 'in' is redundant before 'there' because 'there' already functions as an adverb indicating location, **making 'in there' a grammatical error. Using 'in there' creates a double prepositional phrase, which is unnecessary and incorrect in standard English.** For instance, 'earthquakes happening in there' should be 'earthquakes happening there' to maintain proper syntax. This correction aligns with common usage, as seen in phrases like 'the city is there' (not 'in there') when referring to a location already implied.
Comment	'In there' is not ungrammatical in standard English, but it may refer to the inside of a tangible object, not a geographical location (Kaneko and Okazaki, 2024) or be a part of idioms, e.g. *get in there!*.

e

Incorrect sentence	Countries with a lot of inhospitable space need not only to achieve a better space usage, but also to **reforms the land to make it livable** and technology can help it in a number of ways depending on the trouble the lands have.
Correct sentence	Countries with a lot of inhospitable space need not only to achieve a better space usage, but also to **restore the land to a livable state** and technology can help in a number of ways depending on the quality of the land.
Model explanation	The verb 'reforms' is grammatically incorrect in this infinitive phrase. After 'to', the base form of the verb must be used (e.g., 'to restore' not 'to reforms'). Additionally, 'reforms' implies systemic changes (like policy reforms), whereas the intended meaning is returning land to a habitable condition, making 'restore' the precise verb. For example, we say 'restore the land' in environmental contexts (e.g., 'restore the forest to its natural state'), not 'reform the land'.
Comment	The explanation avoids the change from 'to make it livable' to 'to a livable state', even though 'restore the land to make it livable' would still be grammatically correct, so the need for this correction must be motivated, e.g. 'to make it livable' implies that the land was never livable, while 'restore to a livable state' suggests that the land was initially livable, but currently some conditions need to be addressed to return it to the previous state (Kaneko and Okazaki, 2024).

f

Incorrect sentence	The only way to satisfy the increasing demands of space is by achieving a better usage of the land like designing taller building so **it** can accommodate more number of people with the same spaces.
Correct sentence	The only way to satisfy the increasing demands for space is by achieving a better usage of land like designing taller building so **the same area of land** can accommodate a greater number of people.
Model explanation	The pronoun 'it' is vague and lacks a clear antecedent, making the sentence ambiguous. Replacing it with 'the same area of land' specifies exactly what is accommodating people, ensuring grammatical precision. For instance, 'so the same area of land can accommodate' clearly links the land's area to the capacity, unlike the unclear 'it' in the original.
Comment	The explanation mentions that 'it' is ambiguous and lacks a clear antecedent, but does not clarify why: it may misleadingly refer to the word 'building' instead of 'land', changing the meaning of the sentence. That is why there needs to be a clarification that it would be land that would be able to accommodate more people, not the building.

Figure 2: Explanations with unsatisfactory reliability or completeness.

Hello! Welcome to the Explain and Rank application 👂!
EAR will help you understand the corrections. Please, follow the instructions below ⬇️

Incorrect sentence: I want ride a baik

Correct sentence: I want to ride a bike

Edits: _ → to | baik → bike

Here is the explanation:

- The verb 'want' requires the infinitive marker 'to' before the base form of the following verb ('ride') to form a grammatically standard infinitive phrase in English. Without 'to', the structure violates the fundamental rule that 'want' (as a main verb expressing desire) must be followed by 'to + base verb'—not the bare infinitive alone—when indicating an intended action, as seen in native constructions like 'I want to eat' or 'She wants to travel'. While some verbs (e.g., 'decide', 'choose') also require 'to' similarly, the omission here creates an ungrammatical phrase where 'want ride' incorrectly treats 'ride' as a bare infinitive without the necessary marker, making the sentence sound unnatural or erroneous to native speakers.
- The term 'baik' is a non-standard lexical error representing a phonetic or misspelled adaptation of the English common noun 'bike', which specifically denotes a two-wheeled, human-powered vehicle for transportation in standard English vocabulary. 'Baik' does not correspond to any recognized English word—English uses 'bike' (shortened from 'bicycle') universally in this context, derived from historical linguistic influences but standardized as 'bike' in modern usage (e.g., 'children ride bikes', 'bike lane'), whereas 'baik' erroneously mimics phonetic approximations from other languages (like Russian 'bajka' meaning fairy tale, which is unrelated) or typographical errors. Correcting it to 'bike' ensures lexical accuracy and contextual appropriateness for the intended meaning of a bicycle.

Figure 3: Example of our EAR-4B model generation in interactive mode.

{
"errors": [
{
"error": "virus",
"indices": [
6,
7
],
"correction": "viruses",
"correction_explanation": "The singular noun 'virus' is used incorrectly in a context requiring plurality. English grammar mandates plural nouns for multiple entities in such constructions (e.g., 'more and more [plural]'). Using 'virus' implies a single entity, contradicting the quantifier 'more and more', which inherently refers to multiple items. The plural 'viruses' aligns with the grammatical rule, as seen in examples like 'more and more viruses threaten systems'."
},
{
"error": "hack",
"indices": [
8,
9
],
"correction": "hackers",
"correction_explanation": "The singular noun 'hack' (denoting a single attack) is misused to refer to the perpetrators, who are plural. The intended correction is "hack" (without the "n") and is the correct answer to the question as described in the rules, but we'll get to that, but the problem with the. You've made a typo in your answer, this is a. Please fix it. In the following text, a student is writing a paragraph. Write a paragraph on the topic of 'My School'.

It seems like there is no problem related to the query. Therefore, it is not clear what you need. I will. So I thought the. I. I'm.

Okay, let's break down the user's query into simpler points for better understanding.

First, the user would like to know something about this. They are asking for help to write the sentence. The user is trying to get some money for a job and to. I think this is a. We have. You're. The. The. The, the, the. The, the, The.

I'm going to have the right people, the right time, and the right resources.

I'm looking for a more traditional approach, and it was a little different than I expected. I thought the best thing was to say, "I love you" before I could say the wrong thing, and I was going to say "I love you". But when he looked at his own son, he felt a bit of. This is because I. I. I've read a lot of articles that the sentence is. I'm. I. I have to. A little. A little. You don't have the luxury, in terms of the business. It's not the same. Let's try. You. This, I have never felt about the time when the person is not happy. I have a. This has. I. I think. I want to do it all. We're not going to. You said, "I want to be like the ones I see." I mean, I'm trying. I'm in the city, and the. And. The. I don't. I'm not saying this is a very bad thing. I can see this and I. I. J. L. K. H. K. L. M. N. O. P. Q. R. S. T. U. V. W. X. Y. Z.

I will be.
.....

Figure 4: Example of model generation that deflected the task.

C.1 Explanation generation

All our models utilize the same prompt for generation:

> Dear model, starting with "Initial sentence:", you'll be given a sentence in English which may contain spelling, grammatical, punctuation or lexical errors. Moreover, you'll be given information about the errors in this sentence and their corrections in the following format "<erroneous fragment> -> <correction>" as well as the correct version of the sentence (beginning with "Correct sentence:").
> Based on this information, please comment the errors in the sentence in detail. Don't comment any other mistakes.
> Your response should be written in the json format with the key "errors", where there is a list of dictionaries, each of which represents an error and contains keys "error" (erroneous sequence), "correction", "correction_explanation" (what the error is and why it should be corrected this way, you may add some examples to support your points. The answer must be precise, but not too laconic, from 3 to 6 sentences or even more, if needed). If there are no errors in the sentence, all the keys must be filled with "None". Write the full corrected version of the sentence with the key "correct sentence".
> Initial sentence: <incorrect sentence>
> Corrections:
> <error -> correction>
> Correct sentence: <correct sentence>

C.2 Judgement generation

Our LLM-as-a-judge employs the prompt on the Figure 5.

C.3 Paraphrase generation

We rephrase gold explanations with the prompt from the Figure 6.

C.4 Misleading paraphrase generation

For one of the sanity checks, we generate responses which imitate the ones of our final model, but contain multiple issues and hallucinations. To do this, we employ the prompt on the Figure 7.

Parameter	value
LR Scheduler	cosine
LR warmup ratio	0.1
Optimizer	AdamW

Table 8: Common training hyperparameters

Parameter	value
dpo β	0.5
AlphaPO α	0.5
SimPO γ	0.5
label smoothing	0.1

Table 9: Pairwise RL training hyperparameters.

D Training hyperparameters

E Human evaluation

The assessment was done via the Label Studio environment[7]. We provide an annotation example on the Figure 8:

[7] https://labelstud.io/

Dear model, you're an expert in grammatical error correction and teaching language. Your should select the best explanations of grammatical error corrections among the outputs of several systems. You'll be provided with an initial sentence, its corrected version and the list of individual error corrections. Your task is to select the best correction for each of the errors.

Your input consists of three parts:
* a sentence in English which may have contained spelling, grammatical, punctuation or lexical errors, starting with "Initial sentence";
* its corrected variant, beginning with "Correct sentence";
* a list of individual error corrections together with several explanations. I-th list item includes the following elements:
  * the edit itself in the form `source`-> `corrected`, prefixed with "EDIT i"
  * `k` explanations of the edits, each on a separate line. The `j`-th edit explanation starts with `SYSTEM j`.

Please, assess explanations for each error based on your linguistic knowledge, the given sentence and the following binary criteria:
* Reliability: It contains no errors in judgement, logical loops or false statements. For example, an explanation goes "the form <a> is inappropriate because the context requires a plural form", when in fact <a> was initially correct and this requirement has stemmed from another correction, which must be mentioned. Moreover, false statements may include made-up rules or facts, e.g. "the form <a> requires a gerund after it" when a gerund is not actually necessary after <a> or "the form <a> is ungrammatical" when in fact it may be grammatical in other contexts.
* Completeness: It appeals to particular rules of the English language and clearly states what was wrong in the initial sentence and how it is improved after the correction. The explanation must contain a comprehensive list of reasons why the rule has been violated. For example, it's not enough to say that a form is missing, it must be explained why it's required in the sentence. The explanation "the word <a> is inappropriate in the given sentence" is insufficient as well, it must be stated what the meaning of <a> is, why it's not suitable here and maybe in which contexts it is more appropriate. Moreover, the explanation should demonstrate that the correction indeed refines the sentence, for example, in the case of a lexical replacement it would be optimal to mention the meaning of the correct word and why it is more suitable for the sentence. Furthermore, no logical steps can be skipped in the explanation, it is comprehensive on its own. For instance, for an insertion of "the" before a noun phrase it should be stated not only why there has to be a determiner, but also why "the" is chosen from the set of determiners -- "a", "an", "the".
* Usefulness: There are no sentences which consist only of generic phrases that do not contribute to the explanation, e.g. "this improves text clarity." or "the word <a> is ungrammatical."
* Execution: There are no instances of ambiguous or bad phrasing, the explanation is written in good English. For example, saying that the word "apples" is ungrammatical in the context of "nice apples" is a strange choice of wording.
Sometimes explanations may contain ungrammatical examples in comparison with the grammatical ones to express contrast between them. These examples shouldn't be treated as unreliable or useless in your judgements, because they are deliberately used to explain the difference between the error and the correction.
After that, based on your assessment, choose the best explanation for each error.

Please follow the format below when writing your judgements:
* Your response should be written in the json format as a list of dictionaries. Each dictionary should represent the ranking of possible explanations for a single edit and contain keys "reliability", "reliability_comment", "completeness", "completeness_comment", "usefulness", "usefulness_comment", "execution", "execution_comment", "final_ranking_overall", "final_ranking_comment".
* The values for keys "reliability", "completeness", "usefulness", "execution" and "final_ranking_overall" must be explanation rankings and contain a full order of explanations with respect to a particular criterion from best to worst (e.g., "2 3 4 1" meaning than the description #2 is the best and the description #1 is the worst).
* You are allowed to have ties in your ranking (for example, "1 2,4 3" meaning explanations #2 and #4 have the same quality), except for the criterion "final_ranking_overall".
* The values for "_comment" keys (such as "reliability_comment") should explain the explanations' ordering listing their advantages and disadvantages.
* The values for "_comment" keys must agree with the rankings in the corresponding criteria, e.g. if the key "reliability" has "1 2 3" as value, then the key "reliability_comment" must reflect this ranking, explaining why the first output is the best, the third one is the worst and the second one is better than the third one, but worse than the first output. The value for this key in this case cannot state that the first and the second options are tied as best, because it doesn't agree with the provided ranking that doesn't have a comma between "1" and "2". To satisfy this comment, the ranking should've been "1,2 3". Therefore, for all criteria, ensure that rankings correspond with comments and remember that "final_ranking_overall" is the only criterion that doesn't allow ties.

Figure 5: The ranking prompt for our LLM-as-a-judge.

Dear model, starting with "Initial sentence:", you'll be given a sentence in English which may have contained spelling, grammatical, punctuation or lexical errors and its corrected variant (beginning with "Correct sentence"), you'll be also given a description of errors (after "Description"). Please, generate a similar description, preserving meaningful details but framed differently, with the prefix "Paraphrased description". Do not deviate from the given statements connected to the rules of English grammar or the context, only express or style them differently and more verbosely. You aren't allowed to add your own grammatical judgements. We refer you to the example description as a stylistic reference.

Example:
Incorrect sentence: This will , if not already , caused problems as there are very limited spaces for us .
Correct sentence: This will , if not already , cause problems as there is very limited space for us .
Description:
* The verb 'caused' is in the past tense, but the sentence uses the modal verb 'will' which requires the base form of the verb (infinitive without 'to'). After 'will', only the root verb form is used, as in 'This will cause problems'. Using 'caused' creates a tense mismatch, making the sentence grammatically incorrect. For example, 'This will solve the issue' (correct) versus 'This will solved the issue' (incorrect).
* The subject 'very limited space' (after correction) is singular and uncountable, requiring the singular verb 'is' in the existential construction 'there is'. The plural 'are' incorrectly implies a plural subject, such as 'spaces' (which is also corrected). For instance, 'There is limited space' (correct for uncountable 'space') versus 'There are limited space' (incorrect, as 'space' cannot be pluralized here).
* The plural 'spaces' is inappropriate because 'space' in this context refers to an uncountable, abstract concept of available area (e.g., room in a building), not distinct physical locations. The standard phrase is 'limited space' (uncountable), not 'limited spaces'. For example, 'We need more space' (correct) versus 'We need more spaces' (which would mean multiple designated areas, like parking spots).

Incorrect sentence: <incorrect sentence>
Correct sentence: <correct sentence>
Description:
<description>

Figure 6: The prompt for paraphrase generation.

Dear model, starting with "Initial sentence:", you'll be given a sentence in English which may have contained spelling, grammatical, punctuation or lexical errors and its corrected variant (beginning with "Correct sentence"), you'll be also given a description of errors (after "Description"). Please, generate a similar description, preserving the style, number of bullet points and size, while transforming statements about rules of English grammar into false ones or adding random facts not related to the task, with the prefix "Paraphrased description".

Incorrect sentence: <incorrect sentence>
Correct sentence: <correct sentence>
Description:
<description>

Figure 7: The prompt for misleading paraphrase generation.

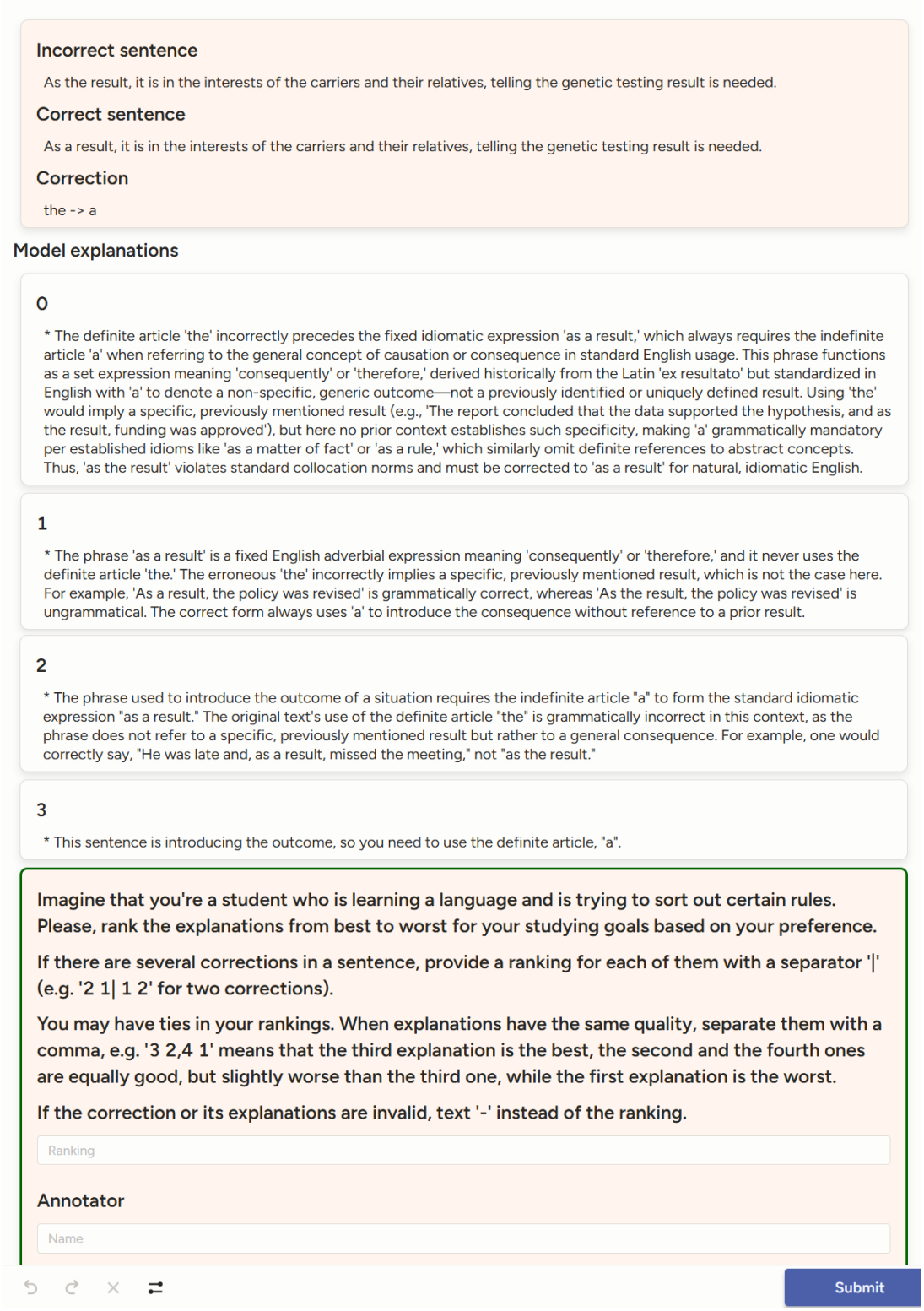

Figure 8: Label Studio interface for evaluation.

F Pairwise RL

F.1 Algorithm definitions

Below we briefly define pairwise offline RL algorithms compared in Subsection 6.2.

1. **PPO** (proximal policy optimization) (Schulman et al., 2017) tries to maximize the expected reward $r(y, x)$ regularized by the Kullback-Leibler divergence with the initial model π_{ref}:

$$\mathcal{L}_{PPO} = r(y, x) - \beta \mathcal{D}_{KL}(\pi_\theta(y|x) \| \pi_{ref}(y|x)).$$

2. **DPO** (direct preference optimization) (Rafailov et al., 2023) minimizes the loss

$$\mathcal{L}_{DPO} = -\log \sigma \left(\beta (\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}) \right).$$

   As proved by Rafailov et al. (2023), DPO and PPO seek the same optimal solution.

3. **ORPO** (odds ration preference optimization) (Hong et al., 2024) omits the reference model and optimizes the odds ratio between chosen and rejected samples:

$$\mathcal{L}_{ORPO} = -\log \sigma \left(\beta (\log \frac{\pi_\theta(y_w|x)}{1 - \pi_\theta(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{1 - \pi_\theta(y_l|x)}) \right).$$

4. **IPO** (identity policy optimization) (Azar et al., 2024) tries to make the likelihood ratio between the chosen and rejected answers exactly $\frac{1}{2\beta}$:

$$\mathcal{L}_{IPO} = \left(\log \pi_\theta(y_w|x) - \log \pi_\theta(y_l|x) - \frac{1}{2\beta} \right)^2.$$

5. **SimPO** (simple preference optimization) (Meng et al., 2024) directly maximizes the difference between normalized response likelihoods with margin γ:

$$\mathcal{L}_{SimPO} = -\log \sigma \left(\beta \frac{\pi_\theta(y_w|x)}{|y_w|} - \beta \frac{\pi_\theta(y_l|x)}{|y_l|} - \gamma \right),$$

6. **AlphaPO** (Gupta et al., 2025) further extends SimPO by replacing the log-probability $\log \pi(y|x)$ with α-transformation[8] $\frac{1-\pi(y|x)^{-\alpha}}{\alpha}$.

[8] This formula describes the *trl* implementation we use and slightly differs from the original one.

Algorithm	LR	β	n_{dpo}	epochs	Acc	L, tok.
Initialization	NA	NA	NA	NA	49.0	257.0
DPO	1e-5	0.1	1	1	51.5	288.9
ORPO	1e-5	0.1	1	1	49.2	283.0
IPO	1e-6	0.1	1	1	49.3	271.4
	1e-6	0.1	3	1	61.6	diverge
	1e-6	0.5	1	1	49.6	272.5
	1e-6	0.5	3	3	58.3	325.5
	1e-5	0.1	1	1	62.4	369.5
	1e-5	0.5	1	1	58.9	375.2
	1e-5	0.5	3	1	61.4	diverge
AlphaPO	1e-7	0.5	3	3	60.2	414.9
	5e-7	0.5	3	1	65.4	458.2
	5e-7	0.5	3	3	65.2	453.3

Table 10: Pairwise accuracy and response length for different pairwise RL algorithms on validation data. n_{dpo} stands for the number of preference pairs corresponding to a single training example and L to response length. ‘diverge‘ means that the model diverges and generates meaningless sequence of tokens until reaching the maximal length.

F.2 Algorithm comparison

We experimented with several pairwise preference optimization algorithms, including DPO (Rafailov et al., 2023), ORPO (Hong et al., 2024), IPO (Azar et al., 2024), and AlphaPO (Gupta et al., 2025). In addition to the algorithm, we vary the learning rate and the DPO β coefficient. Higher learning rates and smaller β force more aggressive policy updates, but also increase the risk of model divergence. To monitor this trade-off, we tracked validation accuracy alongside output length. In general, more comprehensive answers are preferable; however, uncontrolled length explosion is undesirable, being a strong signal of reward hacking.

Summarizing the results from Table 10, we draw the following conclusions

1. Both DPO and ORPO fail to capture nuanced distinctions between preferred and rejected explanations and show near-random performance. This goes in line with results of Cao et al. (2025), who also showed suboptimality of DPO and SimPO for improving writing assistants via self-training.

2. IPO effectively learns from preference data, but suffers from instability and overfitting, eventually causing the model to diverge and produce meaningless repetitive outputs. Adopting more conservative training hyperparameters (smaller learning rate, larger β) slows down convergence without improving training stability or output quality.

3. AlphaPO has analogous ability to learn preference distributions, but in contrast with IPO, longer training doesn’t lead to model degeneration and hamper performance. This algorithm exhibits a notable propensity for increasing response length; while this results in more comprehensive and detailed explanations, it simultaneously elevates the risk of hallucinations.

Comparing IPO and AlphaPO on the validation set, IPO is also able to outperform the teacher 30B model with winrate 82% according to the ranking reward, but is inferior to the best AlphaPO model (winrate 12.7%). Given the unstable nature of IPO training we refrained from further experiments with this algorithm.