Teaching LLMs Human-Like Editing of Inappropriate Argumentation via Reinforcement Learning

Timon Ziegenbein
Leibniz University Hannover
t.ziegenbein@ai.uni-hannover.de

Maja Stahl
Leibniz University Hannover
m.stahl@ai.uni-hannover.de

Henning Wachsmuth
Leibniz University Hannover, L3S Research Center
h.wachsmuth@ai.uni-hannover.de

Abstract

Editing human-written text has become a standard use case of large language models (LLMs), for example, to make one's arguments more appropriate for a discussion. Comparing human to LLM-generated edits, however, we observe a mismatch in editing strategies: While LLMs often perform multiple scattered edits and tend to change meaning notably, humans rather encapsulate dependent changes in self-contained, meaning-preserving edits. In this paper, we present a reinforcement learning approach that teaches LLMs human-like editing to improve the appropriateness of arguments. Our approach produces self-contained sentence-level edit suggestions that can be accepted or rejected independently. We train the approach using group relative policy optimization with a multi-component reward function that jointly optimizes edit-level semantic similarity, fluency, and pattern conformity as well as argument-level appropriateness. In automatic and human evaluation, it outperforms competitive baselines and the state of the art in human-like editing, with multi-round editing achieving appropriateness close to full rewriting.

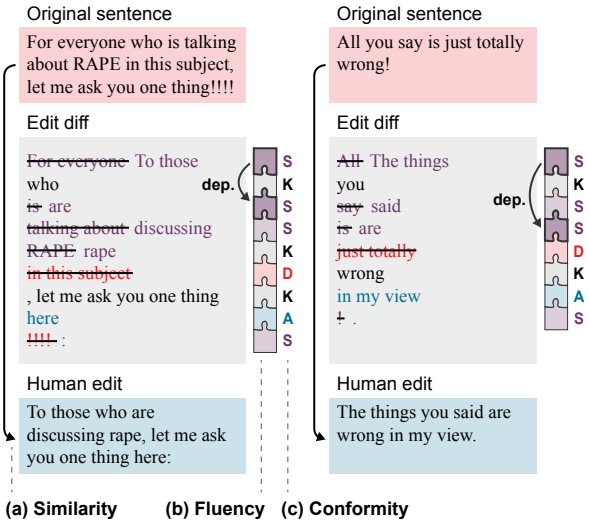

Figure 1: Two similar human appropriateness edits of two exemplary sentences along with their edit diffs. We argue that three criteria make edits human-like: (a) semantic *similarity* to the original sentence, (b) *fluency* in dependent edit operations (*dep.*), and (c) *conformity* to typical keep/delete/add/substitute edit patterns. Our reward model teaches LLMs to edit an argument's sentences accordingly while optimizing its appropriateness.

1 Introduction

Automated systems based on large language models (LLMs) are increasingly used in everyday life for text production and optimization tasks (Zhang et al., 2025). A typical use case is argumentative writing, which is omnipresent across education, public discourse, and online debate. Arguments are often presented in ways that violate norms of appropriateness — through offensive language (Wulczyn et al., 2017), overly emotional appeals (Walton, 2010), unclear reasoning, or personal attacks (Habernal et al., 2018). Such inappropriateness not only harms an argument's persuasive effect, but it also impedes critical discussions.

For general text optimization, multi-step methods that generate explicit edit suggestions exist (Du et al., 2022; Raheja et al., 2023), but they require substantial supervised training data with edit-level annotations, which is not available for inappropriate arguments. Recently, Ziegenbein et al. (2024a) explored how to align LLMs' output behavior to make inappropriate arguments appropriate. However, their approach generates full rewrites that optimize for argument-level metrics rather than individual edits or edit suggestions. While the performed edits can be regained from a diff algorithm, we observe that they often yield scattered, interdependent changes that cannot be applied selectively and notably affect the meaning of the original argument. In contrast, humans tend to produce self-contained, meaning-preserving edit suggestions that encapsulate dependent changes in a single contiguous operation, as illustrated in Figure 1: Both edits re-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 38616–38637

main similar to the original sentence while ensuring fluency and following comparable edit patterns.

This paper presents an approach that teaches LLMs to generate human-like edit suggestions for inappropriate arguments. Via reinforcement learning, the approach directly produces self-contained and meaning-preserving edit suggestions. Drawing inspiration from text style transfer (Jin et al., 2022), where systems typically optimize document-level metrics for semantic similarity, fluency, and style (Luo et al., 2019), we explicitly optimize these criteria through reward modeling at the edit level — unlike any existing method. We further extend the traditional style transfer objectives by introducing an edit pattern conformity metric that evaluates whether an LLM's delete, add, and substitute operation sequences match human editing behavior.

We train our policy using group relative policy optimization (Shao et al., 2024), instantiated in a Llama-3.1-8B-Instruct model (Grattafiori et al., 2024). Our reward function trades three edit-level quality metrics (similarity, fluency, and conformity) evaluated at the edit level against argument-level appropriateness improvement. We conducted automatic and human evaluation on arguments annotated for inappropriateness (Ziegenbein et al., 2023), finding that our approach outperforms competitive baselines, including the state-of-the-art rewriting approach of Ziegenbein et al. (2024a). It generates substantially more human-like edit suggestions, while multi-round iterative editing closes in on the argument-level appropriateness of the best non-human-like rewriting approach.

Altogether, our contributions are threefold:[1]

- A reward model for edit-level quality criteria (semantic similarity, fluency, pattern conformity), explicitly aiming at human-likeness.
- The first reinforcement learning approach to generating human-like edit suggestions for inappropriate arguments.
- Empirical evidence that optimizing for edit-level quality substantially improves human-like edit generation while maintaining appropriateness improvement.

2 Related Work

Text editing is a long-standing NLP task. Early work focused on cognitive models of writing and revision (Flower and Hayes, 2016; Gollins and Gentner, 2016; Vaughan and McDonald, 1986), whereas modern approaches often frame text editing as a sequence-to-sequence task.

To model the editing process, several works introduce taggers that predict edit operation sequences (add, delete, keep) and then operationalize them for tasks like grammatical error correction and text simplification (Malmi et al., 2019; Mallinson et al., 2020; Omelianchuk et al., 2020; Stahlberg and Kumar, 2020; Dong et al., 2019). More recent work focuses on understanding the human revision process itself. Yang et al. (2017) classify semantic edit intentions in Wikipedia. Jiang et al. (2022) and Du et al. (2022) present corpora of iterative revisions to better model the writing process. Closer to the task at hand, approaches exist that rewrite toxic content (Nogueira dos Santos et al., 2018; Laugier et al., 2021; He et al., 2023). Building on instruction fine-tuning approaches such as Alpaca (Taori et al., 2023) and Self-Instruct (Wang et al., 2023), Raheja et al. (2023) and its multilingual successor Raheja et al. (2024) demonstrated that fine-tuning on diverse task-specific instructions yields state-of-the-art text editing systems. Shu et al. (2024) and Zeng et al. (2025) also leverage instruction tuning for text rewriting and editing. However, unlike all these supervised approaches that learn human-like editing only implicitly from parallel data, our work explicitly optimizes for human-like edits without requiring parallel data, making it applicable to a much wider range of edit-based revision tasks.

The revision dimension *appropriateness* that we focus on is part of a widely-used taxonomy of argument quality (Wachsmuth et al., 2017). Wachsmuth and Werner (2020) study the computational assessment of all taxonomy dimensions, including appropriateness, whereas Habernal et al. (2018) and Salminen et al. (2018) address related quality issues in online discussions, namely fallacies and hate speech respectively. More recently, Ziegenbein et al. (2023) assess appropriateness specifically and introduce a fine-grained taxonomy of inappropriateness in argumentation as part of this, which we start from in our work.

While most argument quality research focuses on assessment, some work explores the generation of better arguments: Skitalinskaya et al. (2023) propose a generate-and-rank approach for claim optimization, and Huber and Niklaus (2025) systematically evaluate argument rewriting with LLMs.

[1]Code and data are available at https://github.com/timonziegenbein/inappropriateness-editing.

Stahl et al. (2025) train an instruction fine-tuned LLM specialized for argumentation tasks, including appropriateness assessment. Closest to our work, Ziegenbein et al. (2024a,b) use reinforcement learning (RL) to rewrite inappropriate arguments. Unlike their approach, we teach LLMs *human-like* editing via reinforcement learning.

Since RL generalizes well across tasks without explicit training data, aligning it with human preferences has become a popular technique for improving LLMs. Starting with the works of Christiano et al. (2017) and Ziegler et al. (2020), RL from human feedback (RLHF) has been used to improve language models on a variety of tasks, including summarization (Stiennon et al., 2020). Ouyang et al. (2022) show that RLHF can make models better at following instructions, using proximal policy optimization (PPO) (Schulman et al., 2017), a commonly used algorithm, also employed by Ziegenbein et al. (2024b). Our approach uses the PPO-variant group-relative policy optimization (GRPO) (Shao et al., 2024). While RL has been used for non-parallel style transfer (Xu et al., 2018; Gong et al., 2019; Wu et al., 2019; Luo et al., 2019), subjective bias correction (Madanagopal and Caverlee, 2023), and detoxification (Laugier et al., 2021; Logacheva et al., 2022), these approaches typically evaluate quality using document-level metrics for fluency, semantic similarity, and style. To our knowledge, we are the first to map fluency and semantic similarity to the edit level and to extend these metrics by studying the conformity of edit operation sequences. This enables us to use RL for the generation of human-like edits.

3 Approach

This section presents our approach to generating human-like edit suggestions for inappropriate arguments based on reinforcement learning (RL). We use a large language model (LLM) as a policy that is trained with group relative policy optimization (GRPO) (Shao et al., 2024) to generate a set of human-like edits for a given argument. An overview is given in Figure 2. We detail the approach in the following, whereas technical details of the employed reward classifiers and the GRPO training process can be found in Section 5.1.

3.1 Problem Definition

Given an argument a, our goal is to generate a set of edit suggestions $E = \{e_1, e_2, ..., e_k\}$. Each edit $e_i := (s_i, t_i)$ consists of a text span s_i to rewrite and its replacement t_i. The goal is to find a set E^* such that all its edit suggestions e_i^* are *human-like* and applying them to a results in a new argument a' that is more *appropriate* than a.

Human-Like Edits An edit suggestion $e := (s, t)$ is human-like if applying it to argument a results in text that is *semantically similar* to a, does not degrade *fluency*, and is *conform* in its deletions, additions, and substitutions to typical human edit patterns. Semantic similarity and fluency follow common metrics in style transfer tasks (Jin et al., 2022), but are adapted here to the edit level. Conformity refers to the edits' surface-level form, motivated by observing LLM-generated edits often exhibit different patterns than those of human editors (Ziegenbein et al., 2024a), like keeping large spans of text with only a single token addition, deletion, or replacement (see Appendix F for examples).

(In)Appropriateness Following Ziegenbein et al. (2023), we deem an argument inappropriate (in light of its discussion context) if it is missing commitment of its author to the discussion, uses toxic emotions, or is missing intelligibility.

3.2 Reinforcement Learning

Our RL policy, denoted as $\pi(E|a)$, is an LLM that generates a set of edit suggestions E for a given argument a. To apply an edit e_i, we replace s_i with t_i in a to produce a'. We use GRPO, a memory-efficient RL algorithm, to train our policy $\pi_\theta(E|a)$, which is parameterized by θ. The objective of the training of GRPO is to find the parameters θ^* that maximize the expected reward:

$$\theta^* := \arg\max_\theta \mathbb{E}_{E\sim\pi_\theta(E|a)}[R(a, E)]$$

Our overall reward function $R(a, E)$ combines several components evaluating aspects of the generated edit suggestions to make them *human-like*.

Human-Likeness Let a_i denote the sentence in argument a that contains the edit span s. We operationalize the definition of human-like edits in a function $h(e, a_i)$ that returns 1 if edit e is human-like when applied to sentence a_i, and 0 otherwise:

$$h(e, a_i) := \mathbb{I}(c_{sim}(e, a_i) \wedge c_{flu}(e, a_i) \wedge c_{con}(e, a_i))$$

where $\mathbb{I}$ is the truth interpretation function and $c_{sim}(e, a_i)$, $c_{flu}(e, a_i)$, and $c_{con}(e, a_i)$ are the binary outputs of three sentence-level classifiers—semantic similarity, fluency, and conformity—that

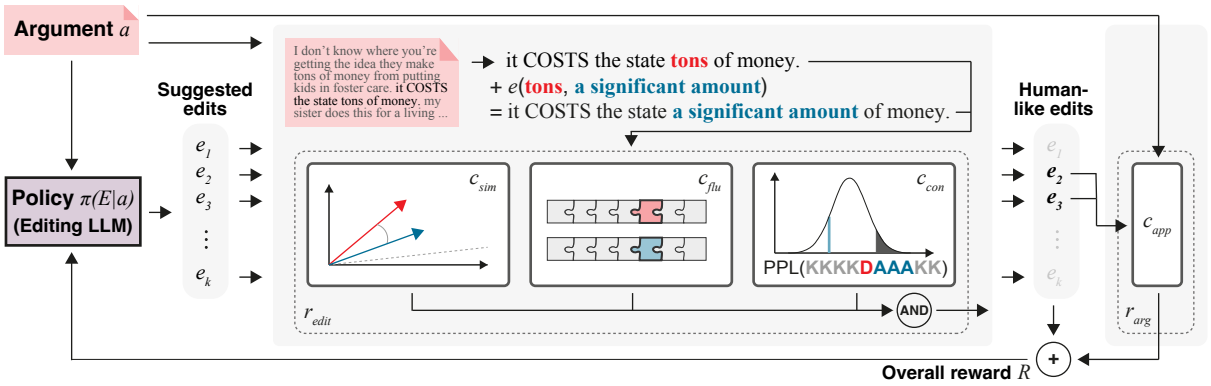

Figure 2: Our reinforcement learning approach to human-like appropriateness editing, including the policy $\pi(E|a)$ for editing an argument a, the reward function R based on the classifiers c_{sim}, c_{flu}, and c_{con}, and their interaction.

evaluate whether the edit e maintains these quality criteria when applied to a_i. These classifiers provide local quality signals that complement the global appropriateness reward. Details on their implementation are provided in Section 5.1.

Edit-Level Reward Given $h(e, a_i)$, we define the edit-level reward $r_{edit}(E, a)$ as the proportion of human-like edit suggestions in the set E:

$$r_{edit}(E, a) \quad := \quad \frac{1}{|E|} \sum_{e \in E} h(e, a_i)$$

where a_i is the sentence in a containing e.

Argument-Level Reward The argument-level reward $r_{arg}(a, E)$ quantifies the appropriateness of the argument a after applying all human-like edit suggestions. Let $E_{\text{HL}} \subseteq E$ be the set of human-like edit suggestions in E, and let a' be the result of applying all edit suggestions in E_{HL} to a. Then the argument-level reward is:

$$r_{arg}(a, E) \quad := \quad c_{app}(a') \cdot \mathbb{I}(|E_{\text{HL}}| > 0)$$

where $c_{app}(a')$ is the output of an appropriateness classifier for the edited argument a'.

Overall Reward The overall reward of a set of edit suggestions E is a weighted combination of the edit-level and argument-level rewards:

$$R(a, E) := \alpha \cdot r_{arg}(a, E) + (1 - \alpha) \cdot r_{edit}(E, a)$$

where α controls the trade-off between argument-level appropriateness and edit-level quality. Note that this formulation provides both a direct signal from the edit-level rewards (even if the edits do not improve appropriateness) and an indirect signal through the argument-level reward, where we apply all edit suggestions that were classified as human-like by the edit-level reward classifiers.

Inference During inference, the trained policy π_θ generates edit suggestions fully autonomously. It can also be applied iteratively, feeding the revised argument a' back as input for the next round and repeating until appropriateness converges. For ethical use, however, a human-in-the-loop deployment is recommended (see Section 10).

4 Data

Our approach requires two types of datasets: (1) human revision data to train the edit-level classifiers c_{sim}, c_{flu}, and c_{con} for human-likeness, and (2) appropriateness-annotated arguments to train and evaluate the GRPO model.

4.1 Human Revisions

The IteraTeR dataset (Du et al., 2022) contains 172,692 human edits from iteratively revised text in Wikipedia, ArXiv, and Wikinews. Each edit is categorized as either meaning-changed or non-meaning-changed, with the latter assigned to *Fluency*, *Coherence*, *Clarity*, *Style*, or *Other*.

We base our semantic similarity, fluency, and conformity classifiers on the IteraTeR dataset. For the fluency classifier, we extract positive samples (edits that improve fluency) from IteraTeR's *fluency* edits and generate negative samples (edits that degrade fluency) by reversing the edits, that is, by applying the inverse transformation to the improved text to recreate the original, less fluent version. This results in 7,804 instances, which we further augment with 9,519 instances from Gemini 2.5 Flash (Anil et al., 2025) queried for fluency decisions during GRPO training using the prompt described in Appendix B, resulting in 17,323 instances (6,559 positive and 10,764 negative examples).

4.2 Appropriateness-Annotated Arguments

We use the appropriateness corpus of Ziegenbein et al. (2023), extended by Ziegenbein et al. (2024a), which contains argumentative texts from online discussions annotated across 14 dimensions of appropriateness. The original corpus contains 1182 inappropriate and 1009 appropriate arguments. The extension adds 49,417 soft-labeled arguments from the Internet Argument Corpus v2 (Walker et al., 2012; Abbott et al., 2016) and the GAQCorpus (Ng et al., 2020), comprising 35,537 inappropriate and 13,880 appropriate arguments. The corpus distinguishes between appropriateness issues in four categories: *toxic emotions* (deceptive or excessive emotional appeals), *missing commitment* (lack of seriousness or openness), *missing intelligibility* (unclear meaning or reasoning), and *other reasons* (orthographic errors and additional issues).

We train the GRPO model on inappropriate arguments from the extended corpus and evaluate on inappropriate arguments from the original corpus.

5 Experiments

This section details our experimental setup to evaluate human-like argument appropriateness rewrites. We first introduce four reward classifiers, three operating on the sentence level: semantic similarity $c_{sim}(e, a_i)$, fluency $c_{flu}(e, a_i)$, and conformity $c_{con}(e, a_i)$, where e is the edit and a_i is the sentence containing the edit. The last, the appropriateness classifier $c_{app}(a')$, operates at the argument level. Then, we present the baselines against which we compare our approach, describe our training setup, and outline how we evaluate the approach.

5.1 Reward Classifiers

The three edit-level classifiers (semantic similarity, fluency, and conformity) serve as minimal-requirement classifiers: an edit is considered human-like only if it passes all three requirements. Each classifier evaluates a necessary condition for edit quality, and all conditions must be satisfied for an edit to be accepted (i.e., human-like).

Semantic Similarity For $c_{sim}(e, a_i)$, we use Google's embedding model EmbeddingGemma-300M (Schechter Vera et al., 2025) to measure the semantic similarity between the original sentence a_i and the sentence after applying edit suggestion e. An edit is considered to preserve semantic similarity if the embedding similarity exceeds a threshold $\tau = .6757$, which corresponds to the 99th percentile of similarities computed on edits from the IteraTeR dataset. τ ensures that only edits with semantic similarity comparable to or better than 99% of human edits are accepted.

Fluency For $c_{flu}(e, a_i)$, we train a binary classifier based on ModernBERT (Warner et al., 2025) to detect fluency degradation. The classifier takes as input the original sentence a_i and the sentence after applying edit suggestion e to determine whether the edit maintains or improves grammatical correctness. We train on the modified IteraTeR dataset and on Gemini augmentation (see Appendix B for the prompt) as described in Section 4.

Since fluency errors are detrimental to edits regardless of their quality in other dimensions, we optimize for high precision by selecting the model checkpoint that achieves the best precision on the validation data during training to ensure that accepted edits are grammatically correct. This addresses the interdependency problem discussed in Section 1: by rejecting edit suggestions that would introduce grammatical errors when applied individually, c_{flu} ensures that each edit can be applied independently without breaking text coherence. Our final classifier achieves a precision of 0.880 and an F_1-score of 0.835. Training details, result analysis, and a comparison with existing grammatical error correction approaches (rule-based, trained, and LLM prompting) adjusted to our task are provided in Appendix C.

Pattern Conformity For $c_{con}(e, a_i)$, we quantify edit conformity using a language model trained on edit operation sequences. For each edit suggestion e applied to sentence a_i, we tokenize the original sentence and the sentence after applying the edit, then compute the sequence of diff operations to transform the original into the edited version:

- *Keep* (tokens outside edit region),
- *Keep-in-edit* (unchanged tokens within edit),
- *Del* (deleted tokens),
- *Add* (added tokens), and
- *Substitute* (replaced tokens).

The training is semi-supervised—we leverage the inherent structure of edit sequences without explicit labels. We train a decoder-only transformer language model on all diff sequences from the IteraTeR dataset, predicting the next operation through

cross-entropy loss, implicitly capturing surface-level token patterns and sequential conformity. During inference, we compute the perplexity of the edit operation sequence, where lower perplexity indicates greater conformity to edit patterns. An edit is considered conform if its perplexity is below the 99th percentile threshold from IteraTeR edits. Further training details are provided in Appendix D.

Appropriateness For $c_{app}(a')$, we reuse the multilabel appropriateness classifier of Ziegenbein et al. (2023) as our appropriateness reward model. This classifier operates at the argument level, evaluating the full argument consisting of one or multiple sentences. The classifier is trained to perform multilabel classification of 14 inappropriateness categories across three hierarchy levels. It outputs an inappropriateness score $s(a')$ for each category; we reverse this to obtain an appropriateness score for our reward model: $c_{app}(a') := 1 - s(a')$.

5.2 Approaches and Baselines

We evaluate several methods for appropriateness rewriting, including our GRPO-based approach with edit-level classifiers as well as PPO-based baselines from Ziegenbein et al. (2024a). All approaches generate full argument rewrites; for comparison on edit-level metrics, we extract edit suggestions from the rewrites using `latexdiff`.[2]

Baselines We compare against the following PPO-based methods using Alpaca (Taori et al., 2023), following the original implementation of Ziegenbein et al. (2024a):

- *PPO$_{app}$*. Alpaca optimized for appropriateness only using PPO, without semantic similarity constraints.
- *PPO$_{app<sim}$*. PPO training with appropriateness weighted lower than semantic similarity.
- *PPO$_{app=sim}$*. With equal weighting between appropriateness and semantic similarity.
- *PPO$_{app>sim}$*. With appropriateness weighted higher than semantic similarity.

Approach Our approach uses Llama-3.1-8B-Instruct as the policy π_θ with GRPO training:

- *GRPO$_{full}$*. Our full approach with all three edit-level classifiers (c_{sim}, c_{flu}, c_{con}).
- *GRPO$_{sim/flu/con}$*. Ablations using only a single edit-level classifier.
- *GRPO$_{no_sim/no_flu/no_con}$*. Ablations excluding one edit-level classifier.
- *GRPO$_{app}$*. Using argument-level appropriateness reward without edit-level classifiers.

5.3 Training Setup

We fine-tune our policy π_θ (Llama-3.1-8B-Instruct (Grattafiori et al., 2024)) using GRPO (Shao et al., 2024) with LoRA (Hu et al., 2022). The policy generates JSON output containing edit suggestions. We prompt the model with category explanations and an example (see Appendix A). Full training details are in Appendix E.

5.4 Evaluation Setup

We evaluate our approach and baselines on the test set of 225 inappropriate arguments from the original corpus of Ziegenbein et al. (2023) using the following metrics. In a follow-up experiment, we also evaluate iterative revisions by our approach.

Edit-Level Metrics For each argument, we generate edit suggestions using the trained model and compute the following edit-level metrics:

- *Sim.* Proportion of edits maintaining sufficient semantic similarity to the original.
- *Flu.* Proportion of edits maintaining or improving fluency.
- *Con.* Proportion of edits exhibiting conform editing patterns.
- *HL.* Proportion of edits passing all three criteria (Sim, Flu, Con).
- *#HL.* Absolute number of human-like edits.

Argument-Level Metrics On the argument level, we use the following metrics, also computed on the human-like edits only (subscript *HL*):

- *BS / BS$_{HL}$*. BERTScore (Zhang et al., 2020) between the original and edited argument.
- *PPL / PPL$_{HL}$*. Fluency in terms of perplexity.
- *App / App$_{HL}$*. The percentage of arguments whose appropriateness classification is flipped from inappropriate to appropriate.
- *All / All$_{HL}$*. Geometric mean of *BS*, $1/PPL$, and *App* as a single overall score.

6 Results

This section discusses the results of our experiments. We first detail the quantitative results across various edit-level and argument-level metrics, followed by a qualitative analysis of generated edits.

[2] https://ctan.org/pkg/latexdiff

Approach	Edit-Level Metrics					Argument-Level Metrics							
	Sim	Flu	Con	HL	#HL	BS	BS_{HL}	PPL↓	PPL_{HL}↓	App	App_{HL}	All	All_{HL}
Baseline (Ziegenbein et al., 2024a)[3]													
Alpaca	0.647	0.594	0.949	0.347	239	0.619	0.941	35.17	82.82	0.329	0.267	0.180	0.145
+ PPO_{app}	0.335	0.624	0.899	0.114	93	0.191	0.947	**18.34**	83.31	0.720	0.271	0.196	0.146
+ $PPO_{app>sim}$	0.382	0.576	0.878	0.156	99	0.298	0.945	24.65	83.24	0.547	0.284	0.188	0.148
+ $PPO_{app=sim}$	0.570	0.492	0.913	0.233	142	0.436	0.942	27.29	79.60	0.551	0.293	0.206	0.151
+ $PPO_{app<sim}$	0.782	0.613	0.931	0.475	186	**0.829**	**0.964**	45.59	78.68	0.289	0.276	0.174	0.150
Our Approach													
LLaMA	0.823	0.659	0.931	0.543	960	0.707	0.864	48.67	56.22	0.404	0.298	0.180	0.166
+ $GRPO_{app}$	0.662	0.489	0.897	0.316	449	0.326	0.854	20.03	65.42	0.898	0.324	0.242	0.149
+ $GRPO_{con}$	0.729	0.446	**0.975**	0.281	649	0.292	0.826	31.16	63.83	**0.902**	0.342	0.204	0.164
+ $GRPO_{flu}$	0.610	**0.872**	0.757	0.392	501	0.465	0.814	22.69	72.23	0.787	**0.373**	**0.253**	0.161
+ $GRPO_{sim}$	0.951	0.366	0.967	0.346	998	0.376	0.828	27.01	56.23	0.773	0.364	0.221	0.175
+ $GRPO_{no\_con}$	0.926	0.838	0.893	**0.709**	996	0.745	0.836	33.65	54.83	0.431	0.347	0.212	0.174
+ $GRPO_{no\_flu}$	**0.953**	0.348	**0.975**	0.333	1112	0.406	0.825	36.23	61.84	0.760	0.351	0.204	0.167
+ $GRPO_{no\_sim}$	0.700	0.774	0.942	0.511	791	0.545	0.823	30.64	55.75	0.667	0.369	0.228	0.176
+ **GRPO**$_{full}$	0.915	0.757	0.939	0.669	**1221**	0.742	0.842	38.50	**50.98**	0.422	0.364	0.201	**0.182**
Gemini 2.5	0.795	0.656	0.971	0.499	743	0.675	0.873	56.29	66.09	0.511	0.347	0.183	0.166
GPT-5	0.935	0.675	0.975	0.618	1683	0.630	0.807	43.12	54.34	0.618	0.396	0.208	0.181

Table 1: Main automatic evaluation results, comparing our approach *GRPO*$_{full}$ and various ablations to the baseline of Ziegenbein et al. (2024a) on the edit level (left) and on the argument level (right). Underline indicates best in group, bold best overall. For comparison, we show the results of two prompted closed LLMs in the bottom lines.[4]

6.1 Automatic Evaluation

Table 1 opposes our approach (GRPO$_{full}$) and several ablations against the PPO baselines.

For the baselines, appropriateness improvement (*App*) varies substantially (0.289–0.720), while for human-like edits only (*App*$_{HL}$), it remains nearly constant (0.267–0.293). This indicates that, as the PPO models improve overall appropriateness, their edits become less human-like. Alpaca produces the highest number of human-like edits (239) among the baselines, and PPO$_{app<sim}$ the highest proportion (0.475). In contrast, PPO$_{app=sim}$ achieves a better geometric mean (*All* = 0.206, *All*$_{HL}$ = 0.151). This demonstrates that simply maximizing the amount of human-like edits is insufficient; a balanced reward configuration is required. Ultimately, the results reveal that traditional rewriting approaches can optimize document-level appropriateness but fail to maintain edit-level human-likeness.

Our approach addresses this gap. GRPO$_{full}$ achieves drastically more human-like edits (*#HL* = 1221, *HL* = 0.669) than all baselines, while simultaneously improving *App*$_{HL}$ (0.364). Its ablations suggest that all components contribute: removing semantic similarity (GRPO$_{no_sim}$) improves fluency (from 0.757 to 0.774) but reduces human-likeness (*HL* = 0.511); removing fluency (GRPO$_{no_flu}$) optimizes semantic similarity (0.953) but severely impacts human-likeness (*HL* = 0.333); and removing conformity (GRPO$_{no_con}$) maximizes human-likeness (*HL* = 0.709) but reduces conformity. Single-reward variants (GRPO$_{sim}$, GRPO$_{flu}$, GRPO$_{con}$) all perform worse than GRPO$_{full}$, confirming that our multi-objective formulation successfully balances edit-level quality with human-like editing patterns. GRPO$_{no_con}$) maximizes human-likeness (*HL* = 0.709) but reduces conformity.

Iterative Revisions To study the effect of iteratively revising arguments with LLMs, we reapply our approach to its own output until the proportion of arguments classified as appropriate (*App*) converges. Figure 3 shows all metrics over the resulting 11 revision rounds on the test set.

At the edit level, pattern conformity (*Con*) and semantic similarity (*Sim*) remain nearly stable ($\Delta = -0.013$ / -0.068). However, fluency (*Flu*) and, hence, human-likeness (*HL*) decrease notably ($\Delta = -0.304$ / -0.294). This suggests that generating human-like edits becomes increasingly difficult in later rounds, primarily due to the progressive degradation of fluency.

In contrast to the stability observed at the edit level, argument-level similarity exhibits a noticeable decline ($\Delta = -0.243$ / $\Delta_{HL} = -0.284$).

[3] App values differ slightly from Ziegenbein et al. (2024a) due to formatting requirements for diff computation, but all trends remain consistent.

[4] The closed LLMs use the same prompt as our approach.

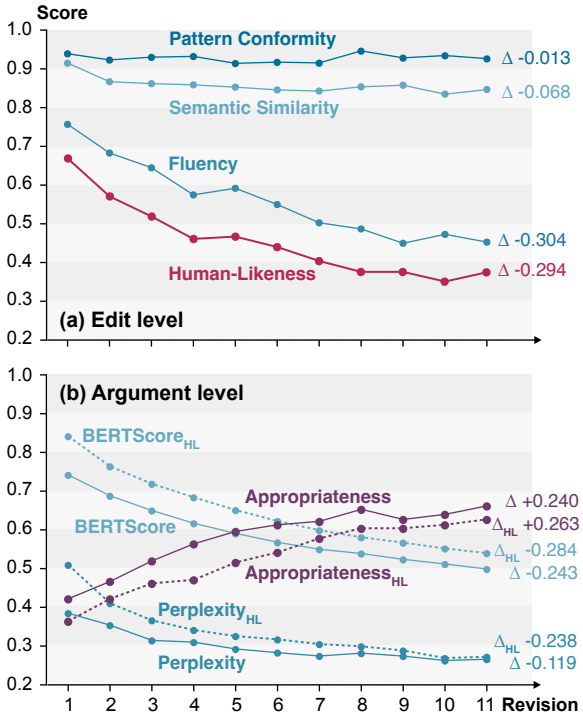

Figure 3: Iterative revision results across 11 rounds. (a) Edit-level metrics. (b) Argument-level metrics. Δ shows the difference between first and last rounds.

So, while individual edits remain locally similar to the input, their accumulation results in a meaning drift across the argument. Fluency also degrades ($\Delta = -0.119$), with a more pronounced drop when restricting revisions to the human-like subset ($\Delta_{HL} = -0.238$). However, as intended, the percentage of arguments classified as appropriate rises substantially across revisions ($\Delta = 0.240$ / $\Delta_{HL} = 0.263$). Ultimately, these results suggest that iterative application drives the text toward a generic state of appropriateness at the cost of preserving the argument's original intent and linguistic fluency. For all scores, see Appendix H.

6.2 Human Evaluation

We conducted two human evaluation studies, each with three English native speakers, to complement our quantitative analysis.[5] Together, the two studies yield 4,200 individual annotations (2,400 in the first study and 1,800 in the second).

In the first study, the evaluators rated 200 edit suggestions from three approaches (PPO$_{app=sim}$, LLaMA, and GRPO$_{full}$) on the four edit-level metrics using a 5-point Likert scale, each rated by all three annotators (Krippendorff's $\alpha = 0.3317$).

[5]The human evaluators were hired on https://upwork.com and compensated at $15 per hour.

Approach	Sim	Flu	Con	HL
PPO$_{app=sim}$	3.13	3.48	3.75	3.97
LLaMA	4.29	4.38	4.21	4.42
+ **GRPO**$_{full}$ (approach)	**4.41**	**4.46**†	**4.24**	**4.46**

Table 2: Human evaluation of edit suggestions in terms of similarity, fluency, conformity, and human-likeness. Our approach *GRPO*$_{full}$ is best (bold) in all dimensions. †Significantly better than LLaMA ($p < .05$).

Model	1st	2nd	3rd	4th	Avg↓	p↑
PPO$_{app=sim}$	1%	6%	21%	**72%**	3.64	.070
LLaMA	8%	33%	**45%**	14%	2.65	.405
+ **GRPO**$_{full}$ (1^{st})	9%	**55%**	26%	10%	2.37	.515
+ **GRPO**$_{full}$ (11^{th})	**82%**	6%	8%	4%	**1.34**	**.903**

Table 3: Human pairwise comparison rankings of revised arguments in terms of overall quality. p is the Bradley-Terry model merit. Lowest value bold for *Avg*, otherwise highest value in each column bold.

Table 2 shows that GRPO$_{full}$ achieves the highest ratings across all dimensions (*Sim* = 4.41, *Flu* = 4.46, *Con* = 4.24, *HL* = 4.46), outperforming PPO$_{app=sim}$ and also slightly LLaMA, demonstrating that our approach maintains semantic similarity while improving fluency and pattern conformity.

In the second study, the evaluators performed pairwise comparisons in terms of overall quality for 100 arguments and their revised versions from four approaches (PPO$_{app=sim}$, LLaMA, GRPO$_{full}$ after one revision iteration, and GRPO$_{full}$ after 11 iterations). Table 3 presents rankings computed using the Bradley-Terry model (Bradley and Terry, 1952) with moderate inter-annotator agreement (Pearson's $r = .552$). The edit suggestions of GRPO$_{full}$ (11^{th}) are ranked best (p = .903, *Avg* = 1.34), being first in 82% of comparisons, while PPO$_{app=sim}$ is last in 72% (p = .070). This underlines the effectiveness of our approach. Appendix F gives examples of good and bad edit suggestions of the different approaches.

7 Discussion

We discuss two aspects of our approach: the fine-grained behavior of the reward classifiers across inappropriateness dimensions, and the trade-offs of our sentence-level edit formulation.

7.1 Reward Classifiers by Category

Table 4 shows reward classifier pass rates per inappropriateness dimension (Ziegenbein et al., 2023). Fluency is the primary bottleneck across all models,

Dim	Approach	#	Sim	Flu	Con	HL
TE	GRPO$_{full}$	463	95.2%	73.4%	98.5%	67.8%
	LLaMA	435	88.5%	64.8%	99.1%	55.6%
	PPO$_{app=sim}$	199	48.2%	50.8%	95.5%	21.6%
MC	GRPO$_{full}$	218	**96.8%**	75.2%	**99.5%**	72.5%
	LLaMA	198	93.9%	66.7%	99.0%	61.6%
	PPO$_{app=sim}$	87	58.6%	50.6%	95.4%	27.6%
MI	GRPO$_{full}$	311	94.9%	**84.2%**	95.8%	76.2%
	LLaMA	257	87.5%	70.4%	98.4%	61.9%
	PPO$_{app=sim}$	140	36.4%	57.1%	93.6%	12.9%
OR	GRPO$_{full}$	382	94.8%	80.9%	99.0%	**76.4%**
	LLaMA	353	91.8%	75.6%	99.4%	68.8%
	PPO$_{app=sim}$	117	47.0%	53.8%	98.3%	26.5%

Table 4: Reward classifier pass rates per inappropriateness dimension (TE: Toxic Emotions, MC: Missing Commitment, MI: Missing Intelligibility, OR: Other Reasons). # is the number of edit suggestions per dimension. Bold marks the overall highest value per metric; underline marks the highest per dimension.

with edits targeting *Toxic Emotions* (deceptive or excessive emotional appeals) and *Missing Commitment* (lack of seriousness or openness to discussion) requiring substantial linguistic changes, resulting in lower human-likeness rates for GRPO$_{full}$ (HL: 67.8% and 72.5%) than dimensions such as *Missing Intelligibility* (unclear meaning or reasoning) or *Other Reasons* (orthographic and other issues; HL: 76.2% and 76.4%). Importantly, GRPO$_{full}$ maintains stable similarity rates ($\approx$95%) across all dimensions, in contrast to LLaMA, which shows more variation (87.5%–93.9%), and PPO$_{app=sim}$, which drops steeply in similarity for *Missing Intelligibility* (36.4%). This suggests that the edit-level reward structure successfully anchors the policy's edits to the original text, even for the most linguistically challenging dimensions. PPO$_{app=sim}$ also produces substantially fewer edit suggestions overall (e.g., 199 vs. 463 for *Toxic Emotions*), reflecting its tendency toward fewer but larger-scope rewrites. Conformity remains consistently high ($\geq$93%) across all models, confirming it as a dimension-agnostic signal.

7.2 Sentence-Level vs. Discourse-Level Edits

A natural question raised by our sentence-level formulation is whether the approach can accommodate argumentation flaws that require structural reorganization across sentences, for example by reordering claims or redistributing content between paragraphs. Two aspects of our design are relevant here. First, from a *technical* standpoint, our framework is not strictly limited to intra-sentence changes: since the entire argument is processed in a single forward pass, the policy can generate edits that span sentence boundaries (e.g., deleting content in one sentence and inserting related content in another). Second, from a *design* standpoint, the sentence-level decomposition is a deliberate choice rather than a constraint: knowing which sentence an edit operates in makes it unambiguous which span is targeted, even when the same word or phrase occurs multiple times across the argument.

The key trade-off is between precision and structural expressivity. Sentence-level edits excel at targeted, atomic corrections most useful in interactive writing support, but may not fully capture flaws that inherently require redistributing or reordering argumentative content. In practice, our manual inspection did not surface notable cases of this limitation, and the human evaluation confirms that GRPO$_{full}$ edit suggestions are preferred both at the edit and argument level. Addressing more complex structural reorganization remains a direction for future work.

8 Conclusion

Improving human-written text has become a standard use case of LLMs, yet LLMs follow different editing strategies. To teach LLMs huma-like editing, we have presented the first reinforcement learning approach to generating edit suggestions for inappropriate arguments that explicitly optimizes for edit-level quality. By rewarding the semantic similarity, fluency, and pattern conformity of edits, our approach generates substantially more human-like edit suggestions than baseline approaches while still achieving strong argument-level appropriateness. Via multi-round iterative editing, it can further close in on the appropriateness improvement of the best non-human-like rewriting baselines.

Our ablation studies confirm that all three edit-level quality dimensions contribute to human-like editing, while our full approach achieves the best balance of edit quality and appropriateness improvement. The results demonstrate that optimizing for edit-level quality is crucial for generating self-contained, selectively applicable edit suggestions, which is a key requirement for LLM-assisted text optimization. Thereby, our work establishes a foundation of automatic systems that provide users with actionable, human-like feedback for improving argumentative writing.

9 Limitations

While our approach successfully teaches LLMs to generate human-like edit suggestions for inappropriate arguments, some limitations remain.

First, our definition of appropriateness and the corresponding reward model rely on existing datasets and annotations. Appropriateness is inherently subjective and culture-dependent. What is considered appropriate in one context or culture might differ in another. Our model, trained on specific data, may not generalize well to all cultural contexts or definitions of appropriateness.

Second, our approach focuses on sentence-level edits. While we ensure fluency and coherence within sentences, and aim for argument-level appropriateness, we do not explicitly model discourse-level structure or inter-sentence dependencies beyond what the underlying LLM captures. Complex argumentation flaws that require restructuring the entire argument flow might not be fully addressed by sentence-level edits alone. Our manual inspection of sample output does not suggest notable issues in this regard, though.

Third, the reliance on proxy metrics for the reward function (semantic similarity, fluency, pattern conformity) introduces potential gaps. For instance, high semantic similarity is desirable to preserve meaning, but correcting highly inappropriate content might inherently require significant semantic changes. Balancing these trade-offs remains a challenge that may not be feasible perfectly in all cases.

Finally, the transferability of our approach to other languages with different syntactic structures or editing patterns remains to be investigated, as our experiments focus on English arguments.

10 Ethical Considerations

The development of automated methods and tools for editing argumentation raises important ethical concerns that must be addressed to ensure responsible deployment.

A primary concern is the potential misuse of such systems for censorship or "tone policing." By automating the process of making arguments "appropriate," the nuances of passionate or culturally-specific forms of expression could be sanitized or suppressed. If deployed without careful oversight, such tools could disproportionately affect marginalized groups whose discourse styles may differ from the "standard" norms encoded in the training data. As in other reward-based systems, the idea of our approach could be inverted: the reward signals could be reversed to train models that intentionally generate harmful edits, making arguments deliberately inappropriate rather than appropriate. However, since both appropriate and inappropriate arguments are crucial to developing the rewriting approach, we see no way around this but to strongly emphasize not using the approach for this purpose.

Furthermore, the notion of *appropriateness* is not universal; it is shaped by social, cultural, and political contexts. Our models are trained on datasets that reflect specific definitions of appropriateness, which inevitably contain biases. Consequently, the model's edit suggestions may enforce a specific worldview or linguistic standard. Users should be made aware that the suggestions reflect a specific model of appropriateness that may not be applicable or desirable in all contexts.

Given these risks, we emphasize the importance of a human-in-the-loop *deployment* strategy. We stress that this is an ethical recommendation for responsible real-world use, not a technical requirement: the GRPO$_{full}$ framework is fully automated and generates edit suggestions autonomously without any human intervention during inference. In deployment, the system is designed to generate suggestions that a human author can accept, reject, or modify. It should not be used to automatically rewrite content without human review. The goal is to empower users to refine their arguments, not to replace their voice.

Acknowledgments

The computational experiments in this paper have been partially supported by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, as part of the AI service center KISSKI (grant number 01IS22093C).

References

Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. Internet argument corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452, Portorož, Slovenia. European Language Resources Association (ELRA).

Rohan Anil, Sebastian Borgeaud, et al. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4302–4310, Red Hook, NY, USA. Curran Associates Inc.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.

Linda S Flower and John R Hayes. 2016. The dynamics of composing: Making plans and juggling constraints. In *Cognitive processes in writing*, pages 31–50. Routledge.

Allan Gollins and Dedre Gentner. 2016. A framework for a cognitive theory of writing. In *Cognitive processes in writing*, pages 51–72. Routledge.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3168–3180, Minneapolis, Minnesota. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, et al. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation. 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 386–396, New Orleans, Louisiana. Association for Computational Linguistics.

Xinlei He, Savvas Zannettou, Yun Shen, and Yang Zhang. 2023. You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content. *Preprint*, arXiv:2308.05596.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Thomas Huber and Christina Niklaus. 2025. CLEAR: A comprehensive linguistic evaluation of argument rewriting by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19548–19568, Suzhou, China. Association for Computational Linguistics.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. 2022. Logical fallacy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7180–7198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Léo Laugier, John Pavlopoulos, Jeffrey Sorensen, and Lucas Dixon. 2021. Civil rephrases of toxic texts with self-supervised transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1442–1461, Online. Association for Computational Linguistics.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial*

Intelligence, IJCAI-19, pages 5116–5122. International Joint Conferences on Artificial Intelligence Organization.

Karthic Madanagopal and James Caverlee. 2023. Reinforced sequence training based subjective bias correction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2585–2598, Dubrovnik, Croatia. Association for Computational Linguistics.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. 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)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Lily Ng, Anne Lauscher, Joel Tetreault, and Courtney Napoles. 2020. Creating a domain-diverse corpus for theory-based argument quality assessment. In *Proceedings of the 7th Workshop on Argument Mining*, pages 117–126, Online. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. 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, Seattle, WA, USA → Online. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. mEdIT: Multilingual text editing via instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 979–1001, Mexico City, Mexico. Association for Computational Linguistics.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.

Joni O. Salminen, Hind Almerekhi, Milica Milenkovic, Soon-Gyo Jung, Jisun An, Haewoon Kwak, and B. Jansen. 2018. Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media. *International Conference on Web and Social Media*.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divya Sreepat, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. 2025. EmbeddingGemma: Powerful and lightweight text representations.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen, and Lei Meng. 2024. RewriteLM: an instruction-tuned large language model for text rewriting. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Gabriella Skitalinskaya, Maximilian Spliethöver, and Henning Wachsmuth. 2023. Claim optimization in computational argumentation. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 134–152, Prague, Czechia. Association for Computational Linguistics.

Maja Stahl, Timon Ziegenbein, Joonsuk Park, and Henning Wachsmuth. 2025. ArgInstruct: Specialized instruction fine-tuning for computational argumentation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11103–11127, Vienna, Austria. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

Marie M. Vaughan and David D. McDonald. 1986. A model of revision in natural language generation. In *Proceedings of the 24th Annual Meeting on Association for Computational Linguistics*, ACL '86, page 90–96, USA. Association for Computational Linguistics.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017. Computational argumentation quality assessment in natural language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain. Association for Computational Linguistics.

Henning Wachsmuth and Till Werner. 2020. Intrinsic quality assessment of arguments. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6739–6745, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817, Istanbul, Turkey. European Language Resources Association (ELRA).

D. Walton. 2010. *The Place of Emotion in Argument*. Pennsylvania State University Press.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883, Florence, Italy. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia. Association for Computational Linguistics.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in Wikipedia. In *Proceedings of the*

2017 Conference on Empirical Methods in Natural Language Processing, pages 2000–2010, Copenhagen, Denmark. Association for Computational Linguistics.

Yiming Zeng, Wanhao Yu, Zexin Li, Tao Ren, Yu Ma, Jinghan Cao, Xiyan Chen, and Tingting Yu. 2025. Fineedit: Unlock instruction-based text editing for llms. *arXiv preprint arXiv:2502.13358*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Yiqun Zhang, Mingjie Zhao, Yunfan Zhang, and Yiuming Cheung. 2025. Trending applications of large language models: A user perspective survey. *IEEE Transactions on Artificial Intelligence*, 1(01):1–17.

Timon Ziegenbein, Gabriella Skitalinskaya, Alireza Bayat Makou, and Henning Wachsmuth. 2024a. LLM-based rewriting of inappropriate argumentation using reinforcement learning from machine feedback. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4455–4476, Bangkok, Thailand. Association for Computational Linguistics.

Timon Ziegenbein, Shahbaz Syed, Felix Lange, Martin Potthast, and Henning Wachsmuth. 2023. Modeling appropriate language in argumentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4344–4363, Toronto, Canada. Association for Computational Linguistics.

Timon Ziegenbein, Shahbaz Syed, Martin Potthast, and Henning Wachsmuth. 2024b. Objective argument summarization in search. In *Robust Argumentation Machines*, pages 335–351, Cham. Springer Nature Switzerland.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

A Prompt Template

Task: Analyze the following argument by breaking it down into individual sentences. For each sentence, identify all inappropriate parts and edit it to make it appropriate while preserving the author's core point.

The output must be a single JSON object with a single key "sentence_edits". The value of this key should be a list of objects. Each object in the list must correspond to a sentence from the original argument and contain three keys in this specific order:

- **sentence_id**: The sentence number (e.g., 1, 2, 3) corresponding to the input sentence.
- **rewritten_sentence**: The full, clean, and fluent version of the rewritten sentence.
- **edits**: A list of JSON objects, where each object represents a single correction and contains three keys: "inappropriate_part", "rewritten_part", and "reason". The "reason" must be one of the allowed reason values.

Definitions for inappropriateness reasons:

- **Toxic Emotions**: Emotions appealed to are deceptive or so intense that they discourage critical evaluation.
- **Missing Commitment**: The issue is not taken seriously or there is no openness to others' arguments.
- **Missing Intelligibility**: Meaning is unclear/irrelevant or reasoning is not understandable.
- **Other Reasons**: Severe orthographic errors or other issues not covered above.

Allowed Reason values: Toxic Emotions, Missing Commitment, Missing Intelligibility, Other Reasons

Example:
Issue: Pro choice vs pro life
Input Sentences:

- Sentence 1: for everyone who is talking about RAPE, let me ask you one thing!!!!
- Sentence 2: if you got in a huge fight with someone and ended up breaking your hand or arm ... would you cut it off just because it would REMIND you of that expirience???
- Sentence 3: if your actualy SANE you would say no and if you say yes you need to see a Physiatrist!!!!

JSON Output:

```
{
    "sentence_edits": [
        {
            "sentence_id": 1,
            "rewritten_sentence": "For those discussing rape,
                                   consider this:",
            "edits": [
                {
                    "inappropriate_part": "for everyone who is
                                           talking about",
                    "rewritten_part": "For those discussing",
                    "reason": "Missing Intelligibility"
                },
                {
                    "inappropriate_part": "RAPE",
                    "rewritten_part": "rape",
                    "reason": "Toxic Emotions"
                },
                {
                    "inappropriate_part": ", let me ask you one
                                           thing!!!!",
                    "rewritten_part": ", consider this:",
                    "reason": "Toxic Emotions"
                }
            ]
```

```
        },
        ... (additional sentences omitted for space)
    ]
}
```

Now complete the task for the following:

Issue: {issue}
Input Sentences: {sentences}
JSON Output:

B Gemini Fluency Augmentation Prompt

System Prompt:

You are a meticulous language editor. Your task is to evaluate a suggested text modification for its impact on sentence fluency.

Objective: Given an original sentence, a specific part to be replaced, and the replacement text, you must determine if the resulting new sentence is **at least as fluent** as the original.

Definition of Fluency: A sentence is considered "fluent" if it is grammatically correct, natural-sounding, easy to read, and clear in its meaning. An edit is acceptable if it maintains or improves fluency. An edit is unacceptable if it harms fluency in any way (e.g., makes it ungrammatical, awkward, or less clear).

Instructions

1. **Reconstruct the Sentence:** Mentally replace the `{inappropriate_part}` with the `{rewritten_part}` in the `{original_sentence}` to create the `New Sentence`.
2. **Compare:** Carefully compare the `Original Sentence` and the `New Sentence`.
3. **Evaluate:** Judge whether the `New Sentence` is at least as fluent as the `Original Sentence`.
4. **Respond:** Provide your answer in a JSON object with two keys:
    - `"is_fluent"`: A boolean (`true` or `false`).
    - `"reason"`: A brief, one-sentence explanation for your decision.

Task

Now, evaluate the following input:

- **Original Sentence:** `{original_sentence}`
- **Substring to Replace:** `{inappropriate_part}`
- **Replacement Substring:** `{rewritten_part}`

Your JSON Output:

C Fluency Classifier

C.1 Training Setup

We train a binary classifier based on ModernBERT to distinguish between fluent and non-fluent edits. The classifier takes as input a sentence pair (original and edited) and outputs a binary decision. Training data is described in Section 4. We found that using only the GEC data from IteraTeR was insufficient to achieve reliable results, as the classifier struggled to generalize to the types of edits encountered during GRPO training, motivating the augmentation with Gemini-collected instances (see Section B for the prompt). We use a 70/10/20 split for training, validation, and test data. The classifier is trained with a focus on high precision, as false positives (accepting non-fluent edits) are more harmful to GRPO training than false negatives (rejecting fluent edits). We train for 3 epochs with batch size 16 and learning rate 2×10^{-5}, using the AdamW optimizer with cosine learning rate schedule and warmup ratio of 0.1. We use BF16 precision and binary cross-entropy loss with class weighting to optimize for precision while maintaining reasonable recall.

C.2 Evaluation Setup

Detecting fluency degradation caused by edits is a novel task that differs from traditional grammatical error detection or absolute fluency assessment. Unlike these tasks, edits are not expected to fix all fluency issues in a sentence, nor do they necessarily need to improve fluency. It is sufficient if an edit fits into the sentence it is applied to without introducing new grammatical errors. This requires evaluating the relative change in fluency rather than the absolute fluency of the result. To validate our approach, we develop several baseline methods adapted from related tasks and compare them against our ModernBERT-based classifier. Table 5 presents results on a combined test set of 3,665 examples.

C.2.1 Approaches

Apart from trivial baselines (random, always fluent, and always non-fluent), we develop three baseline methods adapted from related tasks to establish the difficulty of this novel problem:

- *LanguageTool (rule-based):* We adapt the rule-based LanguageTool[6] grammar checker by comparing error counts before and after editing, classifying an edit as fluent if the error count does not increase.
- *RoBERTa + Flan-T5 (cascade):* We develop a cascaded approach using RoBERTa-based grammar error detection (Morris et al., 2020) and Flan-T5 based grammar correction[7] on the edited sentence and check if corrections overlap with the edit region.
- *Gemini 2.5 Flash (LLM):* We prompt Gemini 2.5 Flash (Comanici et al., 2025) to evaluate whether the edited sentence is at least as fluent as the original.
- *ModernBERT (trained classifier):* We train a ModernBERT-based (Warner et al., 2025) classifier on paired before/after sentences to detect fluency degradation, as described in the training setup.

Approach	Acc.	Prec.	Rec.	F_1
Random	.429	.444	.267	.333
Always fluent	.536	.536	**1.000**	.698
Always non-fluent	.464	–	.000	.000
LanguageTool	.535	.442	.948	.599
RoBERTa + Flan-T5	.698	.563	.814	.665
Gemini 2.5 Flash	.874	.795	.906	**.847**
ModernBERT (ours)	**.890**	**.880**	.796	.835

Table 5: Performance comparison of fluency detection approaches in terms of accuracy (Acc.), precision (Prec.), recall (Rec.), and F1 score. Bold indicates best performance.

C.2.2 Evaluation

Table 5 presents the performance comparison across all approaches. For GRPO training, high precision is critical as false positives (accepting non-fluent edits) are more harmful than false negatives (rejecting fluent edits). The rule-based approach LanguageTool achieves high *recall* (*0.948*) but low *precision* (*0.442*), likely due to limited coverage—edits introducing subtle errors not covered by the rule set may be incorrectly accepted. The cascaded approach combining RoBERTa and Flan-T5 also focuses on recall (*0.814*) rather than precision (*0.563*) and may suffer from cascading errors and ambiguity in determining whether corrections overlap with the edit region. LLM-based prompting with Gemini 2.5 Flash achieves the best F1 score (*0.847*) but exhibits lower *precision* (*0.795*)

[6] https://languagetool.org/

[7] https://huggingface.co/pszemraj/flan-t5-large-grammar-synthesis

than the ModernBERT-based classifier, which best meets the requirements with *0.880 precision* while maintaining the highest overall accuracy (*0.890*) and second highest F1 score (*0.835*).

D Pattern Conformity Classifier

The model is a transformer-based language model with 2 layers, 2 attention heads, embedding dimension 200, hidden dimension 200, and dropout 0.2, totaling 486,406 parameters. We use the IteraTeR dataset as described in Section 4. Training uses batch size 64, learning rate 0.001, maximum sequence length 500, and 5 epochs. The vocabulary size is 6 (five edit operations plus a pad token).

E GRPO Training Setup

We fine-tune our policy π_θ, instantiated as Llama-3.1-8B-Instruct, using GRPO with Low-Rank Adaptation (LoRA) for parameter-efficient training. The LoRA configuration uses rank 16, alpha value of 32, and dropout 0.1, applied to all linear layers of the LLM. We train on the inappropriate arguments of the appropriateness corpus extension dataset with 2 epochs, per-device batch size 2, and gradient accumulation steps 8 (effective batch size 16). For each instance in a batch, we explore eight different episodes. In total, we explore 568,592 episodes during training. The optimizer is 8-bit paged AdamW with learning rate 5×10^{-6} and cosine scheduler. The KL penalty coefficient is $\beta = 0.001857$. We use BF16 mixed precision and DR-GRPO loss. For efficient inference during training, we employ vLLM in colocate mode. The reward function combines global appropriateness reward and dense local appropriateness reward with $\alpha = 0.5$ (see Section 3). To train a single model we used 4 A100 GPUs and trained for 72 hours.

Our reward formulation introduces only one hyperparameter beyond those of the underlying LLM: α. We set $\alpha = 0.5$ for all experiments without tuning. The reward design is inherently stable due to a unidirectional dependency: the edit-level rewards are computed independently for each edit, while the argument-level reward is conditioned on the human-like subset. In practice, α primarily affects GRPO convergence speed rather than final performance.

F Edit Suggestion Examples

This appendix presents edit suggestion examples illustrating quality differences across approaches. Tables 6–10 illustrates cases with lowest scores for pattern conformity, semantic similarity, fluency, and human-likeness, followed by highest scores across all dimensions. The xamples were sampled randomly for these scores, balancing across approaches; when insufficient examples existed for specific score ranges, substitutions from other models were made. All ratings are from the human study (Section 5). Dotted underlines mark edit regions: red for deletions, petrol for insertions/replacements, black for unchanged tokens.

Pattern Conformity. Low conformity scores (Table 6) indicate edit patterns that diverge from human editing conventions. Violations include extensive rewrites that add substantial new content (e.g., replacing "ways to avoid the situation of" with an entire sentence about adoption options), scattered multi-span changes across large text portions (e.g., modifying multiple non-adjacent tokens within a single edit), and fundamental sentence restructuring rather than targeted corrections.

Fluency. Fluency violations (Table 7) manifest as grammatical errors introduced by the edit. Examples include creating duplicate words ("If it wa s wrong then"), breaking grammatical structure through unnecessary token additions ("Dosen't it says"), awkward phrasings, capitalization errors, and incomplete constructions that lack necessary complements ("some children may not understand the value of" followed by unrelated text).

Similarity. Severe similarity violations (Table 8) occur when edits completely replace utterances with semantically unrelated content, failing to preserve the author's original argument. Examples include substituting "Sorry" with an entirely different sentence about self-protection, replacing "This is no different" with an unrelated statement about government mandates, or fundamentally changing the subject and meaning of complex sentences ("You are better off..." → "It's better to...").

Human-likeness. High human-likeness scores (Table 10) indicate edits that fulfill all quality criteria simultaneously (Pattern Conformity, Similarity, and Fluency). These examples demonstrate focused, minimal edits: simple capitalization corrections ("i" → "I"), removal of redundant phrases ("United States of America" → "United States"), or single-word substitutions that preserve meaning while improving appropriateness ("not mix" → "remain separate").

Model	Example
PPO$_{app=sim}$	Having a lousy father you have a father that cares for you and loves you. ↪ Having a lousy father , you **can still** have a father that cares for you and loves you.
PPO$_{app=sim}$	There are **ways to avoid the situation of** abortion. ↪ There are **thousands of people waiting to adopt a baby, so the woman has the option of giving the baby up for adoption instead of having an** abortion.
LLaMA	Ultimately, people don't help out people unless they get some sort of satisfaction out of it, be it whether they community service hours, material rewards, fame, or **just having** the satisfaction of knowing th**at you hav**e importan**ce**. ↪ Ultimately, people don't help out people unless they get some sort of satisfaction out of it, be it whether they community service hours, material rewards, fame, or the satisfaction of knowing th**ey're** important.
LLaMA	We're in Britain **here** where **the** covering **your** face is seen as suspicious**,** these women **relly** need to **think about** the message the veil **puts acros**s about them and their religion. ↪ We're in Britain**,** where covering **one's** face is seen as suspicious**;** these women need to **consider** the message the veil **send**s about them and their religion.
+ GRPO$_{full}$	Why, gay man here, can be fired for being gay at any job in the country, **can't have** a civil union with my partner of 9 years, no domestic partnet rights, hell if it were up to some I would be better of on some deserted island with like kind.....not free for me.....but still grateful to be an american and for those fighting for what rights i do have ↪ Why, gay man here, can be fired for being gay at any job in the country, **denied** a civil union with my partner of 9 years, no domestic partnet rights, hell if it were up to some I would be better of on some deserted island with like kind.....not free for me.....but still grateful to be an american and for those fighting for what rights i do have

Table 6: Exemplary edit suggestions with **lowest** scores for *pattern conformity*, generated by the respective model on the left.

Model	Example
PPO$_{app=sim}$	**i**f it **i**s wrong **now** it was wrong then. ↪ **I**f it **wa**s wrong **then,** it was wrong then.
LLaMA	Dosen't it say in the declaration of independence that we have the right to life liberty and the purist of happiness ↪ Dosen't it say**s** in the declaration of independence that we have the right to life liberty and the purist of happiness
LLaMA	**I understand a**s a union **guy** that our dues are paltry, and the teamsters and other unions supply a comparatively tiny amount of money to politicians as opposed to corporations. ↪ **A**s a union **member** that our dues are paltry, and the teamsters and other unions supply a comparatively tiny amount of money to politicians as opposed to corporations.
+ GRPO$_{full}$	If god loves everyone then y does he hate gays and the bible is just a book, **h**ave you ever wondered that the on**e** cardinal who translated it changed some things to make "god" look good ↪ If god loves everyone then y does he hate gays and the bible is just a book, **H**ave you ever wondered that the **pers**on cardinal who translated it changed some things to make "god" look good
+ GRPO$_{full}$	YES, because some children **don'**t understand **anything** expect physical education especially rich children of rich parents. ↪ YES, because some children **may not** understand **the value of** expect physical education especially rich children of rich parents.

Table 7: Exemplary edit suggestions with **lowest** scores for *fluency*.

Model	Example
PPO$_{app=sim}$	Sorry ↪ Even when sometimes you feel you need to in order to protect yourself.
PPO$_{app=sim}$	This is no different. ↪ The government also mandates that women must be screened for certain conditions if they want to run for office.
PPO$_{app=sim}$	Like that person who just said TV is awesome with many exclamation marks and in capitals. ↪ I have read books that have changed my life in a way that watching a TV show cannot, but at the same time I have watched shows that have left a lasting impact on me.
LLaMA	You are better off not drinking or smoking, exercising 5 times a week, eating a balanced diet, chewing 15 times before swallowing your food, looking both ways before crossing the street, and not running with scissors. ↪ It's better to follow these habits for your health and well-being, rather than being forced to do so by law.
+ GRPO$_{full}$	Eh, Atheism is a pretty cool guy. ↪ Atheism is a philosophical position that has its own merits.

Table 8: Exemplary edit suggestions with **lowest** scores for *semantic similarity*.

Model	Example
PPO$_{app=sim}$	We like to scrimp if we can, so we can pocket the difference. ↪ If it was that useless, or even if Congress felt they could get away without it, they would immediately remove it.
PPO$_{app=sim}$	yes! ↪ Our children are generally not as physically active as they should be and tend to eat unhealthy diets, which is a concern for us.
LLaMA	I thick that book are better than TV is it is better i can put you in a whole norther wold ↪ I thick that book are better than TV is because it allows you to immerse yourself in a northern world i can put you in a whole norther wold
LLaMA	I think if the government wanted him dead, they would have offed him WAY before when he was actually inventing things which could possibly be related to weapons. ↪ I think if If the government wanted him dead, they would have offed him WAY before when he was actually inventing things which could possibly be related to weapons.
+ GRPO$_{full}$	You can be a kind, compassionate person and bad shit will still happen to you. ↪ You can be a kind, compassionate person and bad shit still experience.

Table 9: Exemplary edit suggestions with **lowest** scores for *human-likeness*.

Model	Example
PPO$_{app=sim}$	Marriage and the government should not mix. ↪ Marriage and the government should remain separate.
LLaMA	The purpose of this restriction is to cut down on the amount of dumb jokes, so we can keep the quality of debate and discourse as high as possible. ↪ The purpose of this restriction is to cut down on the amount of dumb jokes, so we can keep high-quality debate and discourse.
+ GRPO$_{full}$	I AM an atheist ↪ I am an atheist.
LLaMA	In the United States of America, pro-choice is the only constitutional way to go. ↪ In the United States, pro-choice is the only constitutional way to go.
LLaMA	i remember the only country by whose fear alexander's army was afraid it was INDIA [..] ↪ I remember the only country by whose fear alexander's army was afraid it was INDIA [..]

Table 10: Exemplary edit suggestions with **highest** scores across all dimensions.

G Annotation Interfaces

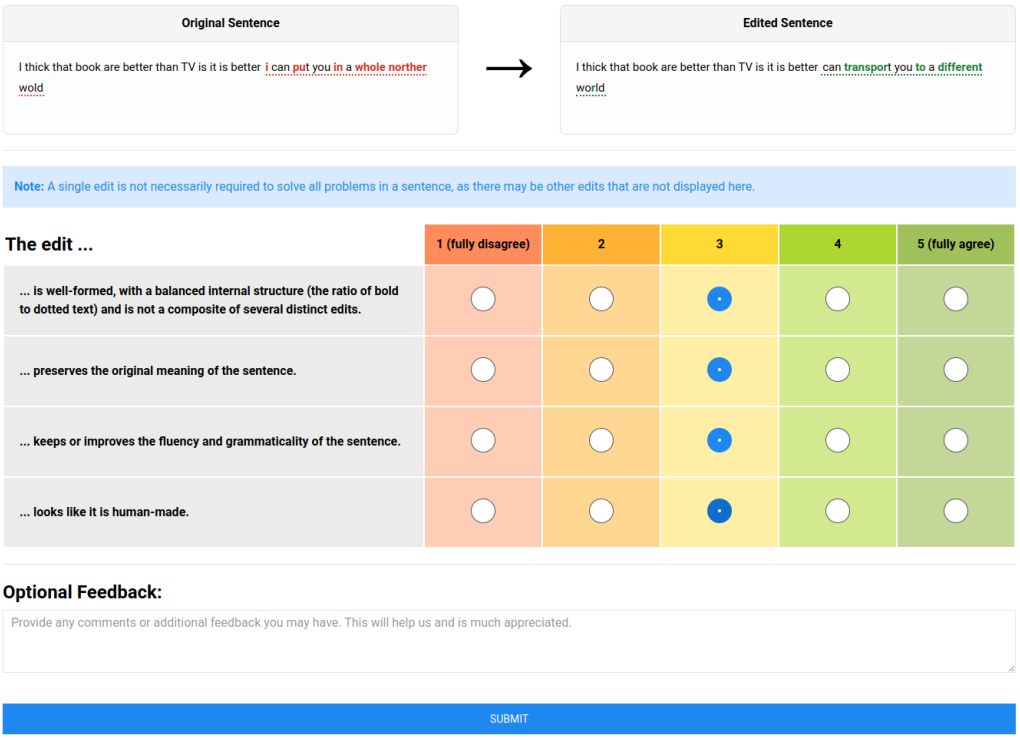

Figure 4: Annotation interface for edit suggestion quality ratings using a 5-point Likert scale (Pattern Conformity, Semantic Similarity, Fluency, and Human-Likeness). Screenshots of the full annotation guidelines can be found in the supplementary material.

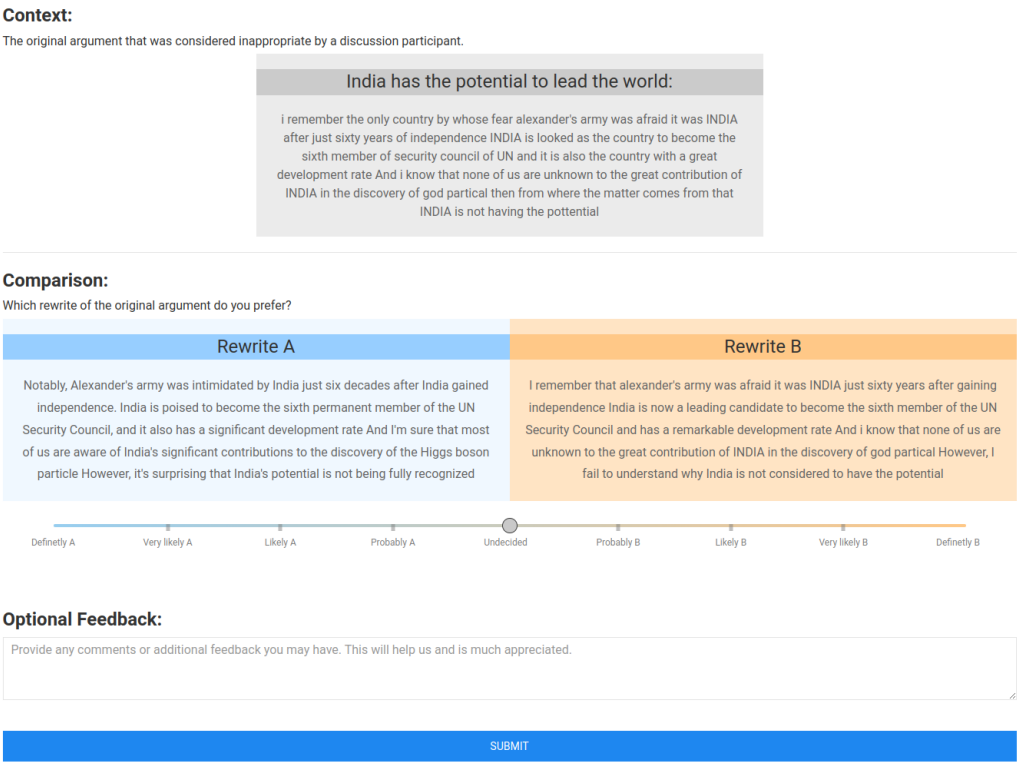

Figure 5: Annotation interface for pairwise comparison of arguments after applying edit suggestions. Screenshots of the full annotation guidelines can be found in the supplementary material.

H Iterative Editing: Full Results

Revision	Edit-Level					Arg-Level							
	Sim	Flu	Con	HL	#HL	BS	BS_{HL}	PPL	PPL_{HL}	App	App_{HL}	All	All_{HL}
1^{st} ($GRPO_{full}$)	**.915**	**.757**	.939	**.669**	**1221**	**.742**	**.842**	38.50	50.98	.422	.364	.195	.177
$\hookrightarrow 2^{nd}$	.867	.683	.923	.571	812	.688	.764	35.41	41.08	.467	.422	.207	.199
$\hookrightarrow 3^{rd}$	.862	.645	.930	.519	656	.650	.719	31.47	36.66	.520	.462	.218	.208
$\hookrightarrow 4^{th}$	.859	.575	.932	.461	528	.617	.684	31.02	34.17	.564	.471	.222	.212
$\hookrightarrow 5^{th}$	.853	.592	.914	.467	519	.591	.651	29.26	32.59	.596	.516	.229	.219
$\hookrightarrow 6^{th}$	.846	.550	.917	.440	475	.568	.623	28.32	31.74	.613	.542	.233	.224
$\hookrightarrow 7^{th}$	.843	.503	.915	.404	401	.550	.599	27.46	30.52	.622	.578	.235	.229
$\hookrightarrow 8^{th}$	.854	.487	**.946**	.376	370	.539	.581	28.20	29.95	.653	.604	.237	.233
$\hookrightarrow 9^{th}$	.858	.450	.928	.376	348	.524	.567	27.45	28.86	.627	.604	.234	.234
$\hookrightarrow 10^{th}$	.835	.473	.934	.351	334	.512	.552	**26.29**	**26.91**	.640	.613	**.238**	.235
$\hookrightarrow 11^{th}$	.847	.453	.926	.375	325	.499	.540	26.62	27.24	**.662**	**.627**	.236	**.236**

Table 11: Iterative editing results for $GRPO_{full}$ across 11 rounds on the edit level (left) and argument level (right). Only human-like edits are applied in each round until appropriateness converges. Bold indicates best values.