Can LLMs Clarify? Investigation and Enhancement of Large Language Models on Argument Claim Optimization

Yiran Wang[1,2], **Ben He**[1,2], **Xuanang Chen**[2*], **Le Sun**[2*]
[1]School of Computer Science and Technology,
University of Chinese Academy of Sciences, Beijing, China
[2]Chinese Information Processing Laboratory,
Institute of Software, Chinese Academy of Sciences, Beijing, China
wangyiran20@mails.ucas.ac.cn, benhe@ucas.ac.cn
chenxuanang@iscas.ac.cn, sunle@iscas.ac.cn

Abstract

In argumentation, the claim is the foundational proposition that underpins the argument, serving as the central pillar upon which the argument is constructed. It guides the subsequent presentation of evidence, reasoning, and analysis, thereby facilitating the audience's understanding of the core issue. Therefore, ensuring that the claim is precise and unambiguous is crucial for constructing a coherent and persuasive argument. While Large Language Models (LLMs) have demonstrated proficiency in text rewriting tasks such as style transfer and query rewriting, their application to claim optimization remains unexplored. Unlike other rewriting tasks, claim clarification requires the model to rewrite ambiguous or unclear segments of the claim, enhance the content by adding omitted key details, and eliminate redundant or verbose elements. Addressing this gap, this paper evaluates the performance of LLMs on the claim clarification task across various settings. While popular rewriting evaluation methods such as BLEU and ROUGE rely on exact word matching, this paper introduces a novel semantic evaluation approach based on a sliding window mechanism. Three distinct LLMs, including LLama2, Mistral, and Qwen2, are assessed for their ability to clarify arguments through zero-shot or few-shot prompting, and supervised fine-tuning (SFT). Additionally, we propose a reinforcement learning-based clarification approach that optimally balances content preservation with claim clarity, thereby augmenting the performance of LLMs on the claim clarification task.

1 Introduction

In argumentation, the claim holds fundamental importance as it constitutes the central assertion or proposition that the argument seeks to validate or substantiate (El Baff et al., 2019). It forms the foundation upon which the entire argumentative structure is built. Without a clearly articulated claim, the argument lacks coherence and direction, rendering it ineffective. The claim serves to focus the argument, guiding the selection and presentation of evidence, reasoning and analysis (Walton, 1996). Even for humans without training, clearly expressing one's claim is a challenging task, so for large language models, this is a difficult task (Osborne, 2010).

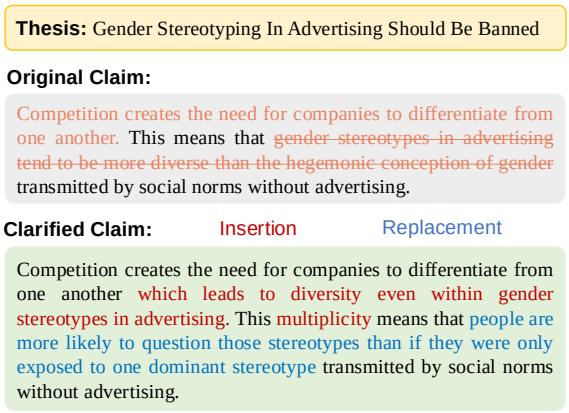

Figure 1: An example of clarified claim and its original version from the ClaimRev corpus (Skitalinskaya and Wachsmuth, 2023). The red part of clarified claim means the insertion of texts. The dashed part of original claim is replaced by the blue part in clarified claim.

Various tasks and datasets have been introduced to discuss the clarity in argumentation (Persing and Ng, 2013; Park et al., 2015; Sundriyal et al., 2023). The concept of clarity emphasizes the precision of arguments, avoiding ambiguity and vagueness, and using language suited to the audience. Wachsmuth et al. (2017a); Ghosh et al. (2016); Wachsmuth and Werner (2020); Li and Ng (2024) suggest that clarity reflects an important aspect in terms of argument quality. Thus, the task of rewriting to improve claim clarity is crucial in the field of computational argumentation.

Several studies have laid the groundwork for

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 4066–4077
January 19–24, 2025.

understanding clarity in argumentation and its computational modeling. Daxenberger et al. (2017) explore the varying conceptualizations of claims across datasets in argument mining and demonstrates that shared lexical features and specific system configurations can mitigate cross-domain classification challenges. Skitalinskaya and Wachsmuth (2023) study collaborative editing behaviors in online debates, aiming to uncover revision patterns that can guide writers on whether their claims need further revision. Skitalinskaya et al. (2023) introduce the task of claim optimization, focusing on improving the delivery of argumentative claims for better persuasion using the BART model (Lewis et al., 2019). Claim optimization task aims to improve the overall quality by rewriting claims.

Inspired by Ziegenbein et al. (2024) using a reinforcement learning-based rewriting approach to mitigate the inappropriateness of arguments, we propose a method based on Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) for argument claim clarification. Our study involves experiments using three distinct large models: LLama2-7b-chat, Mistral-7b-instruct, and Qwen2-7b-chat, each selected for their unique architectural capabilities, potentially influencing their effectiveness in processing and clarifying complex argumentative structures. The experimental results show that the RLHF-based method outperforms the few-shot and SFT methods.

Figure 1 illustrates a clarification process where an original claim is clarified by making specific insertions and replacements. In the original claim, the writer discusses how competition causes companies to differentiate, with a vague reference to diversity in gender stereotypes in advertising. The clarified claim adds specific explanations about how this diversity within gender stereotypes prompts individuals to question those stereotypes more critically, offering a clearer and more precise argument. Clarifying a claim involves refining the language while ensuring that the original intent or argumentative force remains intact. Skitalinskaya et al. (2023) uses BLEU as an important metric to evaluate the effectiveness of argument clarification. However, Claim clarification often involves restructuring arguments or rephrasing sentences to erase ambiguity or vagueness. BLEU heavily relies on matching exact words or phrases (Papineni et al., 2002). In claim clarification, synonymous expressions or rephrasing that improve clarity would be penalized because BLEU does not recognize synonyms as equivalent. To address this issue, we propose a semantic matching evaluation criterion based on a sliding window algorithm, focusing on assessing the changes in the revised text. Additionally, we annotate the differences between 837 pairs of claims derived from ClaimRev dataset (Skitalinskaya and Wachsmuth, 2023) and their revised versions to implement this evaluation metric.

Building on the premise that clarity is pivotal in argumentation, this paper presents a focused investigation into how large language models (LLMs) perform in clarifying argumentative claims. The challenge for LLMs lies in their capacity to distill complex, potentially ambiguous or vague language into a form that is precise, coherent, and tailored to the intended audience. In summary, this paper's main contributions are[1]:

1. We are the first to investigate the argument claim clarification task using large language models (LLMs).

2. We propose an RL-based method to enhance the performance of large language models on the claim clarification task.

3. We propose a new metric based on a sliding window algorithm to precisely evaluate the effect of argument clarification.

2 Related Work

2.1 Argument Claim Quality Assessment

The origins of creating a convincing argument can be traced back to ancient Greece, where the persuasiveness of arguments was explored through dialectic and rhetoric (Aristotle and Kennedy, 2006). An important research goal in the current field is to enable large-scale language models to generate claims that are clear and unambiguous.

With recent advancements in natural language processing, AQ has been studied and applied across various domains, including student essays (Wachsmuth et al., 2016), news editorials (El Baff et al., 2020), internet forum post (Wang et al., 2023) and social media discussions (Wachsmuth et al., 2017c; Skitalinskaya et al., 2021).

Current research on AQ assessment primarily focuses on its role as a sub-task within Argument

[1]Our code is available at https://github.com/ucasYW/Argument-clarification

Mining (AM). However, due to the inherently subjective nature of AQ, a clear definition remains elusive. It is widely believed that numerous factors influence AQ. Wachsmuth et al. (2017b) summarized 15 such factors, categorizing them into logical, rhetorical, and dialectical aspects.

Various studies have attempted to assess argument quality through different factors, which including clarity. For example, Gurcke et al. (2021) evaluated argument sufficiency with human input, positing that a sufficient argument's conclusion can be derived from its premises. Li et al. (2020) assessed argument persuasiveness by analyzing argument structure using a factor graph model. Meanwhile, Singh et al. (2021) focused on elucidating implicit reasoning (warrants) in arguments with the aid of trained experts. Falk and Lapesa (2023) sought to improve AQ assessment by incorporating knowledge from various dimensions into the prediction process via multi-task learning. Although this approach showed some improvement in specific dimensions, its overall performance regarding general quality remains limited.

2.2 Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a reinforcement learning algorithm that has gained popularity in natural language processing (NLP) for fine-tuning models in tasks such as text generation, dialogue systems, and summarization (Han et al., 2023; De Rosa and Papa, 2021; Amouzgar, 2019). Developed by OpenAI, PPO is designed to strike a balance between effective policy updates and stable training. Unlike traditional policy gradient methods, PPO prevents drastic changes to the policy during updates by using a clipped objective function. This makes it a highly stable and reliable method for optimizing complex models like large language models, which are sensitive to abrupt policy shifts.

In the context of NLP, PPO is particularly useful for enhancing specific aspects of text generation, such as fluency, clarity, and coherence (Zhu et al., 2022). For example, when applied to dialogue generation tasks, PPO allows the model to learn from feedback and improve over time (Rohmatillah and Chien, 2021). This feedback-driven approach helps the model better align with human preferences or task-specific goals. The reinforcement learning mechanism rewards the model for generating desirable outputs while penalizing it for producing incoherent or irrelevant responses. As a result, the model can gradually refine its ability to generate high-quality, context-appropriate text.

One of the key advantages of PPO in NLP is its ability to optimize model behavior without sacrificing stability. Traditional reinforcement learning methods can lead to significant variability in model performance due to the sensitivity of language models to policy updates. PPO mitigates this issue by constraining how much the policy can change with each update, ensuring smoother and more reliable improvements. This has made PPO a favored approach for fine-tuning large models, especially when the goal is to generate clear, unambiguous, and contextually relevant text.

3 Methodology

To evaluate and enhance the capability of LLMs in clarifying argumentative claims, an evaluation dataset with claim difference annotations is first constructed by GPT-4 model and human efforts, and then a training method using reinforcement learning from machine feedback is proposed to further enhance the clarify performance of LLMs. Besides, an automatic evaluation metric based on similarity at sliding window level is introduced to present more appropriate and accurate assessment.

3.1 Dataset Construction

Our evaluation dataset is derived from the ClaimRev dataset (Skitalinskaya and Wachsmuth, 2023), which encompasses 124,312 claim revision histories from the debate platform Kialo. Each revision history is formatted as a sequence $(c_1, \ldots, c_m)$, where each claim c_i represents an enhancement over the previous claim c_{i-1}, with $1 < i \leq m$, thus improving its overall quality. The ClaimRev dataset categorizes revision actions into four types: typographical or grammatical corrections, corrections or additions of links, clarifications, and other modifications.

As our research focuses on the ability of LLMs to clarify claims, we extract 21,237 claim pairs (c_{m-1}, c_m) where the revision action involves clarification, totaling 42,474 claims. These are subsequently distributed into training, validation, and test sets in an $80\%, 10\%, 10\%$ ratio, facilitating a structured evaluation of model performance. To examine the capabilities of LLMs in handling complex clarification cases, we implement a detailed data annotation procedure on 837 pairs of claims selected from the test set. This dataset is named as difference dataset. These pairs are specifically

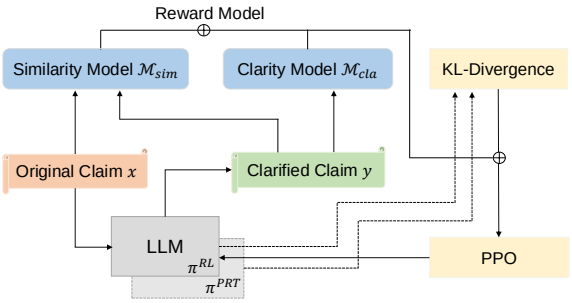

Figure 2: Method for clarifying argument claims involves optimizing the policy π^{RL} using PPO. The goal is to generate a clarified claim from the original claim, while maintain the semantic meaning, and simultaneously enhancing the claim's clarity. This optimization is driven by a reward model, which is determined by a weighted combination of the scalar outputs from similarity and clarify model, and the KL-divergence between the initial policy π^{PRT} and the current policy π^{RL}. The dashed lines represent the LLM's output as a probability distribution over the tokens.

chosen because the modifications between claims involve clauses containing at least 5 words. In oreder to achieve fine-grained substring extraction, We employ GPT-4 for the initial annotation phase, inputting each pair and instructing the model to identify and extract the differing substring between the two claims.

To ensure the accuracy of annotations, we conduct a manual calibration process. We engage annotators who hold graduate degrees taught in English, thereby having the ability to understand and clarify English claims effectively. During this phase, the human annotators carefully review and verify the results, correcting any inaccuracies found in the model's outputs. This combination of automated detection followed by human verification ensures the accurate and reliable identification of differences within the claim pairs.

3.2 PPO for Argument Claim Clarification

The task of argumentative claim clarification is defined as follows. Let x denote an argument claim and y represent its clarified version. The objective is to develop a function $f : x \rightarrow y$ that ensures y preserves as much of the original semantic content of x as possible. Simultaneously, the function must disambiguate the text and eliminate any vagueness, thereby enhancing the clarity and precision of the argument claim. When directly using LLMs as this function, it can demonstrate certain abilities in clarification, but they may not truly know what modifications would be clearer. Therefore, we adopt the reinforcement learning framework to steer an LLM that has already learned to solve the claim clarification task to a certain known extent instead of learning this task from scratch.

Specifically, we design a reward function that encourages large language models (LLMs) to enhance clarity while preserving the original semantics as closely as possible. As depicted in Figure 2, the reward model comprises two components. The first component is the clarity model $\mathcal{M}_{cla}$, a binary classification model that determines whether the input claim is clarified. This model is trained for five epochs on the Roberta-Large model (Liu et al., 2019) using our training data, achieving an F1-score of 0.601. We calculate the clarity score s_{cla} by evaluating the difference in probabilities that the model classifies x and y.

The second component is the semantic similarity model $\mathcal{M}_{sim}$, which evaluates how well the output text y maintains the original semantics of x. For this purpose, we employ Sentence-BERT (Reimers and Gurevych, 2019) as our semantic model. After obtaining the embeddings for x and y, we compute their cosine similarity to determine the similarity score s_{sim}. The final reward score is then defined as follows:

$$r(x, y) = \alpha \cdot s_{cla}(x, y) + (1 - \alpha) \cdot s_{sim}(x, y) \quad (1)$$

where $\alpha \in [0, 1]$ serves as a hyperparameter that measures the balance between maintaining semantic similarity and enhancing clarity, allowing for a nuanced adjustment of the model's focus, catering to specific needs for clarity or fidelity in the modified text.

Following Stiennon et al. (2020), the final reward model R is obtained by penalizing r with the KL-divergence between the initial policy π^{PRT} and the learned policy π^{RL}, thereby discouraging excessive divergence from π^{PRT}.

$$R(x, y) := r(x, y) - \beta \log \left[\frac{\pi^{\text{RL}}(y|x)}{\pi^{\text{PRT}}(y|x)} \right] \quad (2)$$

where $\beta \in \mathbb{R}$ is a hyperparameter that regulates the intensity of the KL-divergence.

3.3 Sliding Window Evaluation

To evaluate the quality of text generation, prior studies commonly use BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and SARI (Xu et al., 2016) as the automatic evaluation metrics (Skitalinskaya

Algorithm 1 Sliding Window Evaluation

Input: Text A of length n, window size k, difference text D, similarity model Sim
Output: Difference semantic similarity S_d
Initialize empty list $SubList$
Initialize $window\_sum = 0$
Initialize $start = 0$
for $end = 0$ to $n - 1$ **do**
    $window\_sum = window\_sum + A[end]$
    if $end \geq k - 1$ **then**
        Append $window\_sum$ to Sub_list
        $window\_sum = window\_sum - A[start]$
        $start = start + 1$
    end if
end for
Get $Sim(D, sub_i)$ for sub_i in $SubList$, $S_d = max(Sim(D, sub_i)$
Return S_d

and Wachsmuth, 2023). However, in the claim clarification task, a large number of insertion edits and replacement edits are involved. Therefore, methods that rely on exact matches between words and sequences cannot accurately assess the performance of claim clarification.

Another popular evaluation metric is the semantic similarity between the generated text and the ground-truth text (Ziegenbein et al., 2024). However, as shown in Figure 3, not all of claims will have complex clarification process as in Figure 1, an argument may process with only minor changes. Thus, when testing samples include cases like those in Figure 3, measuring the overall semantic similarity between the ground truth and the modified claim, one model can achieve high similarity score even it does not provide any modification.

To address this issue, we propose a new evaluation method. In order to focus on verifying whether the modifications to the claims were correctly made, we propose a sliding window-based evaluation method. This algorithm essentially breaks the text A into sliding windows of size k and computes a similarity score between each window and the difference text D. The final result is the maximum similarity score from all windows. If the difference is more than 1, we take the average of them as the final score. We apply this method to the different datasets. Algorithm 1 shows the detailed procedure.

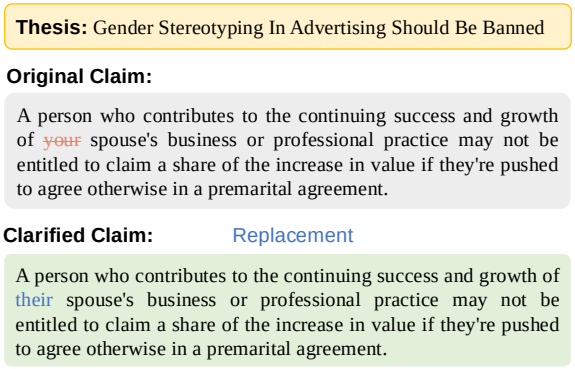

Figure 3: An example case of few changes made to clarify a claim.

4 Experimental Setup

4.1 Evaluation Metrics

As ChatGPT has demonstrated a level of performance comparable to human annotators in coarse-grained annotation (Huang et al., 2024), making it a proper tool for evaluating the text clarity and semantic fidelity in modified claims, in this paper, we primarily conduct automatic evaluation through ChatGPT's win rate, clarity score, the similarity score between the generated text and the ground truth, and the semantic similarity of the differing parts.

Win rate using ChatGPT. We present the claim clarified by the large language models (LLMs) and the ground truth claim (c_m) to ChatGPT-3.5-turbo, asking which claim is clearer and more accurate, without ambiguity or causing vagueness. We use the win rate over the ground truth as a metric to evaluate the large model's performance on this task.

Clarity score. We choose to use our fine-tuned RoBERTa-large model to generate the clarity score. The probability that the claim rewritten by the large model is recognized as clarified is used as the clarity score.

Overall similarity score. Due to its outstanding performance in the STS task, we choose to use the bilingual-embedding-large model (Conneau et al., 2019; Nils Reimers, 2019; Thakur et al., 2020) as our similarity model. We calculate the similarity between the ground truth claim (c_m) and the claim modified by the large language models(LLMs), using this as our similarity score.

Difference similarity score. To validate the capability of large-scale language models in complex

clarifications, we conducted a sliding window evaluation on the difference dataset as describe in Section 3.3. During the evaluation, we set the window size k to 5.

4.2 LLM Settings

To comprehensively evaluate the ability of LLMs in argument claim clarification, we selected three large language models for experimentation: Llama2-7b-chat (Touvron et al., 2023), Mistral-7b-instruct (Jiang et al., 2023), and Qwen2-7b-chat (Yang et al., 2024).We conducted experiments for LLMs under the conditions of zero-shot, few-shot, supervised fine-tuning (SFT), and RLHF.

Zero-shot/Few-shot We utilize the natural language prompts proposed by Reif et al. (2021) to generate clarified claim y given original claim x. For the few-shot setting experiments, we used 1, 3, and 5 examples to examine the effect of the number of examples on the performance of large language models (LLMs) in the claim clarification task. These examples were selected from our validation set.

SFT We trained on 21,237 pairs of claims. The prompts were configured in the same manner as in the zero-shot setting. During the fine-tuning process, we employed the LORA fine-tuning method (Hu et al., 2021). During the fine-tuning stage, we uniformly set the number of training epochs to 5 and learning rate to 5×10^{-6}.

PPO For PPO setting, we used method introduced in Section 3.2 to train large language models. We use the TRLX (Havrilla et al., 2023) framework for PPO training. We initialize π^{RL} with a pre-trained LLM π^{PRT}, which is prompted in natural language to generate y given x. For the clarity model, We trained Roberta-large model (Liu et al., 2019) on the training set with 16,989 pairs of claims, resulting in a 1:1 ratio of clarified to not clarified examples. It has an F1-score of 0.601. For the similarity model, we use paraphrase-MiniLM (Reimers and Gurevych, 2019). We set the hyperparameter α as 0.5 and learning rate to 5×10^{-6}.

5 Results and Discussions

In this section, we present the final experimental results and discuss the performance of the large language models in claim clarification based on the four evaluation metrics introduced in the previous section.

5.1 Overall Performance

Table 1 presents the results of comparing the performance of three different models: Llama2-7b-chat, Mistral-7b-instruct, and Qwen2-7b-chat in both overall and few-shot scenarios.

Llama2-7b-chat PPO improves performance the most across all metrics. Win rate and Clarity score increase from zero-shot to SFT, and to PPO, suggesting that optimization methods significantly enhance the performance. However, the overall similarity decreases slightly in PPO (0.711) compared to SFT (0.752), while the PPO's difference similarity (0.592) increase slightly compared to SFT (0.581). This indicates that PPO can make more precise adjustments in more complex claim clarification tasks compared to the SFT method.

For few-shot settings, Win rate improves steadily from 1-shot (0.720) to 5-shot (0.771), but Clarity score has a slight dip at 3-shot before improving in 5-shot. The overall similarity increases from 1-shot (0.674) to 5-shot (0.701), suggesting Llama2 performs better when given more context. Difference similarity slightly decreases from 1-shot (0.547) to 5-shot (0.545), indicating that adding more examples might not significantly improve its differentiation ability when handling complex cases.

Mistral-7b-instruct It has the highest zero-shot Win rate (0.797) compared to the other models. Similar to Llama2, PPO improves the model's metrics, with a high Clarity score (0.750) and Win rate (0.856). Overall, Mistral-7b shows a strong balance among all of the 4 evaluation metrics.

For few-shot settings, Win rate remains high across all shot settings, with the best performance in 1-shot (0.800). Clarity score improves as more shots are provided, from 0.571 (1-shot) to 0.588 (5-shot). The model shows a slight increase in overall similarity from 1-shot (0.688) to 5-shot (0.708), indicating better task alignment with more examples.

Qwen2-7b-instruct It shows lower Win rates compared to the other two models. However, the SFT and PPO stages improve performance considerably for the performance of this model among all of the 4 evaluation metrics. The overall similarity and difference similarity remains lower compared to Llama2 and Mistral.

Model		Win Rate	Clarity	Overall Sim.	Difference Sim.
Llama2-7b-chat	zero-shot	0.743	0.581	0.696	0.513
	1-shot	0.720	0.553	0.674	0.547
	3-shot	0.766	0.551	0.679	0.552
	5-shot	0.771	0.594	0.701	0.545
	SFT	0.847	0.732	**0.752**	0.581
	PPO	**0.854**	**0.759**	0.711	**0.592**
Mistral-7b-instruct	zero-shot	0.797	0.583	0.711	0.530
	1-shot	0.800	0.571	0.688	0.552
	3-shot	0.798	0.584	0.703	0.559
	5-shot	0.803	0.588	0.708	0.553
	SFT	0.833	0.683	**0.744**	0.575
	PPO	**0.856**	**0.750**	0.713	**0.603**
Qwen2-7b-instruct	zero-shot	0.752	0.565	0.683	0.501
	1-shot	0.733	0.562	0.669	0.533
	3-shot	0.752	0.577	0.678	0.537
	5-shot	0.756	0.579	0.693	0.542
	SFT	**0.849**	0.704	**0.700**	0.579
	PPO	0.846	**0.722**	0.696	**0.583**

Table 1: The overall results of Llama2-7b-chat, Mistral-7b-instruct and Qwen2-7b-chat.

For few-shot settings, Win rate increases slightly with more examples, but remains lower than Llama2 and Mistral. Clarity score improves slightly in 5-shot (0.579) compared to 1-shot (0.562). Overall similarity also increases from 1-shot (0.669) to 5-shot (0.693), showing better alignment as more examples are provided. Difference similarity remains relatively stable across the shots, indicating that more examples may not help Qwen2 handle with complex claim clarification scenario.

In summary, Mistral-7b-instruct demonstrates consistently strong performance, especially in zero-shot and few-shot scenarios. Llama2-7b-chat performs well in overall metrics, especially after SFT and PPO. Qwen2-7b-chat shows improvements with optimization and few-shot learning, but generally performs lower in comparison to the other two models. In few-shot settings, all models show performance improvements with more examples, though the Difference sim tends to stabilize or decrease slightly, which indicates that it is hard to improve model's ability in handling complex clarification scenario by just giving more examples.

5.2 Clarity Analysis

The clarity score in the table measures how clear the model's responses are. As shown in Table 1, the clarity score improves significantly for all models with SFT and PPO optimization, especially for Llama2-7b-chat and Mistral-7b-instruct. Qwen2-7b-chat shows improvement but remains behind the other two models in clarity throughout.

In Table 1, the models benefit from few-shot learning, but the improvement in clarity is more gradual compared to the larger gains seen with SFT and PPO. Mistral-7b-instruct maintains a better balance in clarity, showing steady improvement across the few-shot settings, while Llama2-7b-chat has a more significant increase between 3-shot and 5-shot scenarios. Qwen2-7b-chat remains slightly behind in clarity across all settings.

5.3 Similarity Analysis

The overall similarity and difference similarity reveals clear patterns across the table. As seen in Table 1, SFT significantly boosts the general performance of all three models, with Llama2-7b-chat and Mistral-7b-instruct showing the greatest improvements. However, after PPO optimization, there is a slight decline in overall similarity for both models, indicating that while PPO enhances clarity and Win rate, for the case of a large number of simple instances which only modifies only a few words, the more accurate expression leads to a decrease in overall similarity because the modifications exceed the ground truth standard. Qwen2-7b-chat starts with lower overall similarity and sees only modest improvements, maintaining a gap in performance compared to the other models.

When it comes to handling complex scenarios

	Text	BLEU	Difference Sim.
Original Claim	A religious idea should also be tested. Christians should test their ideas to find scriptural support. Even if they show support from Scripture, they should be open to go deeper to see how other scriptures support or not.	0.704	0.488
Ground Truth	A religious idea should also be tested. Christians should ask if their ideas to find scriptural support. Not just advocating a doctrine, but digging deeper. Even if they show support from Scripture, they should be open to go deeper to see how other scriptures support or not.	-	-
SFT	Christian believers ought to systematically evaluate their interpretations of religious texts by examining multiple biblical passages for consistency, coherence, and potential contradictions, aiming to establish a comprehensive understanding that aligns with established principles of hermeneutics and encourages ongoing spiritual growth.	0.005	0.497
PPO	Christians ought to conduct thorough scriptural research to validate their religious beliefs regarding specific issues, such as the environment, and remain open to reevaluating their understanding in light of additional biblical evidence.	0.010	0.603

Table 2: The clarified claim generated by the Llama2-7b-chat model using SFT and PPO, together with their BLEU scores and difference similarity scores compared with ground truth text.

(in terms of Difference Similarity), all models improve with SFT and PPO, particularly Mistral-7b-instruct, which demonstrates the strongest gains in complex task performance after PPO. Llama2-7b-chat also shows consistent improvement, but the gains are more modest compared to Mistral. Qwen2-7b-chat starts with the lowest difference similarity and continues to lag slightly behind even after optimization, indicating it struggles more with complex tasks than the other models.

In Table 1, all models show gradual improvements in overall similarity as they are exposed to more examples, reflecting that few-shot learning enhances general task performance. However, the improvements in difference similarity are relatively small across all models, indicating that few-shot learning primarily enhances general ability rather than handling complex scenarios. Mistral-7b-instruct and Llama2-7b-chat maintain higher scores in both overall and difference similarity, while Qwen2-7b-chat continues to trail behind, especially in complex scenario performance.

5.4 Sliding Window Evaluation vs. Exact Match Based Method

From the examples in Table 2, we can see that LLMs tend to expand or refine the concepts in the original text when performing the claim clarification task, resulting in more precise expressions. LLMs also rewrite the entire claim text, ensuring that while the concepts in the original text are expanded or refined, the rewritten text remains coherent. These changes result in significant differences in specific wording between the text modified by the LLMs and ground truth text since annotators tend to make changes at a finer granularity. Although the semantics are preserved as much as possible, traditional exact-match-based metrics, such as BLEU, struggle to evaluate the quality of the modifications effectively.

Using the results from the Llama2-7b-chat model after SFT and PPO in Table 2 as an example, their BLEU scores, computed against the ground truth reference, are 0.005 and 0.010, respectively. In contrast, the difference similarity scores generated by our proposed sliding window evaluation are 0.497 and 0.603. However, the original claim achieved a BLEU score of 0.704 compared to the ground truth, but the difference similarity score indicated that it was, in fact, not modified. These results demonstrate that BLEU scores fail to adequately assess the quality of rewrites in this context.

5.5 PPO vs. SFT

In Table 1, both SFT and PPO versions show improvements compared to their zero-shot counterparts. Fine-tuning leads to significant jumps in clarity score, overall similarity, and difference similarity for all models. For instance, in Llama2-7b-chat, fine-tuning increases the overall similarity from 0.696 (zero-shot) to 0.752 and the clarity score from 0.581 to 0.732. Similarly, Mistral-7b-instruct and Qwen2-7b-chat also show notable improvements in general and complex task performance

after fine-tuning.

However, when comparing PPO to SFT, we see different effects. PPO further enhances the clarity score in all models—e.g., from 0.732 (SFT) to 0.759 (PPO) in Llama2-7b-chat—but overall similarity and task alignment do not always increase as much. In Llama2-7b-chat, PPO slightly reduces the overall similarity from 0.752 to 0.711, and in Mistral-7b-instruct, it goes from 0.744 (SFT) to 0.713 (PPO). These discrepancies highlight that PPO focuses more on improving specific behaviors, like clarity and response quality, but may lose some alignment with the original training data, which affects general task performance. As shown is Table 2, the LLM with PPO tend to rewrite broad concepts into more precise descriptions, like "Christians should test their ideas to find scriptural support." to "Christians ought to conduct thorough scriptural research to validate their religious beliefs regarding specific issues, such as the environment." Meanwhile, the text generated by SFT tends to add more modifiers to the original text, making it appear more precise without significantly altering the underlying meaning. For example, the understanding of scripture is broken down into consistency, coherence, and potential contradictions.

6 Conclusion

This paper conducts a comprehensive investigation of large language models in claim clarification task, under zero-shot, few-shot, SFT settings, and also proposes to use PPO algorithm to improve the clarify ability of LLMs, and a new metric based an a sliding window strategy is also introduced to support the evaluation. Through the experiments and analysis presented in this paper, we have demonstrated that large language models can directly clarify ambiguous or unclear claims in some simple cases. When handling complex scenarios, SFT proves highly effective in improving both general and complex scenario capabilities by tailoring the model to specific tasks. PPO, on the other hand, boosts clarity and fluency but may slightly reduce overall task alignment.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62272439) and the Fundamental Research Funds for the Central Universities.

Limitations

The research primarily centers on three large language models with 7b size. This means that the results may not be generalizable to all LLM architectures or to future, more advanced models. While the PPO algorithm enhances clarity, it introduces a trade-off by slightly reducing overall task alignment, indicating that the optimization may not be ideal for balancing all aspects of claim clarification. Further exploration in both broader LLM scenarios and fine-tuning strategies is needed to address these limitations. In further work, we plan to conduct more extensive evaluations across a wider range of LLMs, and under more complex argument optimization scenarios.

References

Farhad Amouzgar. 2019. *Deep reinforcement learning as text generator in image captioning*. Ph.D. thesis, Macquarie University.

Aristotle and George A. Kennedy. 2006. *On Rhetoric: A Theory of Civic Discourse*. Oup Usa.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the essence of a claim? cross-domain claim identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2066, Copenhagen, Denmark. Association for Computational Linguistics.

Gustavo H De Rosa and Joao P Papa. 2021. A survey on text generation using generative adversarial networks. *Pattern Recognition*, 119:108098.

Roxanne El Baff, Henning Wachsmuth, Khalid Al Khatib, Manfred Stede, and Benno Stein. 2019. Computational argumentation synthesis as a language modeling task. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 54–64, Tokyo, Japan. Association for Computational Linguistics.

Roxanne El Baff, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. 2020. Analyzing the Persuasive Effect of Style in News Editorial Argumentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3154–3160, Online. Association for Computational Linguistics.

Neele Falk and Gabriella Lapesa. 2023. Bridging argument quality and deliberative quality annotations with adapters. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2469–2488, Dubrovnik, Croatia. Association for Computational Linguistics.

Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. 2016. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 549–554.

Timon Gurcke, Milad Alshomary, and Henning Wachsmuth. 2021. Assessing the sufficiency of arguments through conclusion generation. In *Proceedings of the 8th Workshop on Argument Mining*, pages 67–77, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chengcheng Han, Xiaowei Du, Che Zhang, Yixin Lian, Xiang Li, Ming Gao, and Baoyuan Wang. 2023. DialCoT meets PPO: Decomposing and exploring reasoning paths in smaller language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8055–8068, Singapore. Association for Computational Linguistics.

Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. 2023. trlX: A framework for large scale reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8578–8595, Singapore. Association for Computational Linguistics.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Fan Huang, Haewoon Kwak, Kunwoo Park, and Jisun An. 2024. ChatGPT rates natural language explanation quality like humans: But on which scales? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3111–3132, Torino, Italia. ELRA and ICCL.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Jialu Li, Esin Durmus, and Claire Cardie. 2020. Exploring the role of argument structure in online debate persuasion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8905–8912, Online. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024. ICLE++: Modeling fine-grained traits for holistic essay scoring. 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 8465–8486, Mexico City, Mexico. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Iryna Gurevych Nils Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *https://arxiv.org/abs/1908.10084*.

Jonathan Osborne. 2010. Arguing to learn in science: The role of collaborative, critical discourse. *Science*, 328(5977):463–466.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Joonsuk Park, Cheryl Blake, and Claire Cardie. 2015. Toward machine-assisted participation in erulemaking: An argumentation model of evaluability. In

Proceedings of the 15th International Conference on Artificial Intelligence and Law, pages 206–210.

Isaac Persing and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, Sofia, Bulgaria. Association for Computational Linguistics.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *ArXiv*, abs/2109.03910.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Mahdin Rohmatillah and Jen-Tzung Chien. 2021. Corrective guidance and learning for dialogue management. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1548–1557.

Keshav Singh, Paul Reisert, Naoya Inoue, and Kentaro Inui. 2021. A comparative study on collecting high-quality implicit reasonings at a large-scale. *ArXiv*, abs/2104.07924.

Gabriella Skitalinskaya, Jonas Klaff, and Henning Wachsmuth. 2021. Learning from revisions: Quality assessment of claims in argumentation at scale. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1718–1729, Online. Association for Computational Linguistics.

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.

Gabriella Skitalinskaya and Henning Wachsmuth. 2023. To revise or not to revise: Learning to detect improvable claims for argumentative writing support. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15799–15816, Toronto, Canada. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *ArXiv*, abs/2009.01325.

Megha Sundriyal, Tanmoy Chakraborty, and Preslav Nakov. 2023. From chaos to clarity: Claim normalization to empower fact-checking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6594–6609, Singapore. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv e-prints*, pages arXiv–2010.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. 2016. Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691, Osaka, Japan. The COLING 2016 Organizing Committee.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017a. 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.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017b. 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, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017c. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark. Association for Computational Linguistics.

Henning Wachsmuth and Till Werner. 2020. Intrinsic quality assessment of arguments. *arXiv preprint arXiv:2010.12473*.

Douglas N Walton. 1996. *Argument structure: A pragmatic theory*. University of Toronto Press Toronto.

Yiran Wang, Xuanang Chen, Ben He, and Le Sun. 2023. Contextual interaction for argument post quality assessment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10420–10432, Singapore. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Ke-Yang Chen, Kexin Yang, Mei Li, Min Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yunyang Wan, Yunfei Chu, Zeyu Cui, Zhenru Zhang, and Zhi-Wei Fan. 2024. Qwen2 technical report. *ArXiv*, abs/2407.10671.

Hao Zhu, Yonatan Bisk, and Graham Neubig. 2022. Language learning from communicative goals and linguistic input. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Timon Ziegenbein, Gabriella Skitalinskaya, Alireza Bayat Makou, and Henning Wachsmuth. 2024. 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.