ArgBench: Benchmarking LLMs on Computational Argumentation Tasks

Yamen Ajjour
Leibniz University Hannover
y.ajjour@ai.uni-hannover.de

Carlotta Quensel
Leibniz University Hannover
c.quensel@ai.uni-hannover.de

Nedim Lipka
Adobe Research
lipka@adobe.com

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

Abstract

Argumentation skills are an essential toolkit for large language models (LLMs). These skills are crucial in various use cases, including self-reflection, debating collaboratively for diverse answers, and countering hate speech. In this paper, we create the first benchmark for a standardized evaluation of LLM-based approaches to computational argumentation, encompassing 33 datasets from previous work in unified form. Using the benchmark, we evaluate the generalizability of five LLM families across 46 computational argumentation tasks that cover mining arguments, assessing perspectives, assessing argument quality, reasoning about arguments, and generating arguments. On the benchmark, we conduct an extensive systematic analysis of the contribution of few-shot examples, reasoning steps, model size, and training skills to the performance of LLMs on the computational argumentation tasks in the benchmark.

1 Introduction

Utilizing the capabilities of large language models (LLMs) for argumentation has drawn significant interest within the NLP community. Some promising applications where argumentation is applied include countering hate speech (Saha and Srihari, 2023a), rebutting misinformation (Zheng et al., 2025), and suggesting various arguments for the possible answers to an input task (Eskandari Miandoab and Sarathy, 2024). Going beyond step-by-step reasoning to understand and generate conflicting reasoning paths helps enhance LLMs' ability to detect reasoning gaps and effectively counter societal harms. This requires not only effective LLM-based methods but also comprehensive benchmarks for computational argumentation.

In general, research on computational argumentation is motivated by applications such as debating machines (Slonim et al., 2021) and writing support (Skitalinskaya et al., 2021). Fundamental *skills* needed for these endeavors include mining arguments (Stab and Gurevych, 2017), assessing their perspectives (Bar-Haim et al., 2017) and their quality (Wachsmuth et al., 2017) as well as classifying reasoning patterns of arguments (Lawrence and Reed, 2016), or generating counterarguments to input arguments (Hua et al., 2019).

While computational argumentation can equip LLMs with human-like reasoning skills (Stahl et al., 2025), existing benchmarks for tracking LLM progress largely overlook respective tasks. Reasoning benchmarks like ARC (Clark et al., 2018) or MMLU (Hendrycks et al., 2020) are limited to tasks with one correct answer, such as mathematics and common sense reasoning. Argumentation, however, deals with questions on complex topics, with conflicting perspectives and divergent reasoning paths, such as "Should we increase the minimum wage?". The few existing argumentation-specific LLM works are limited in scope (details in Section 2), covering only few skills, such as argument mining and generation (Chen et al., 2024).

In this paper, we present a benchmark to assess the capabilities of LLMs over 33 computational argumentation datasets. The benchmark includes 46 tasks that are grouped into five skills: argument mining, argument quality assessment, argument perspective assessment, argument reasoning, and argument generation. It offers a comprehensive evaluation framework for LLMs, covering a broad range of argumentation tasks. The benchmark provides two setups to evaluate LLMs: *prompting* and *leave-one-task-out* per skill. In the prompting setup, an LLM is primed with the task definition, without fine-tuning. In leave-one-task-out, we evaluate the model's ability to generalize to unseen tasks. Specifically, an LLM is evaluated on one task from each skill after fine-tuning it on the remaining tasks in the benchmark. The two evaluation setups assess the computational argumentation skills that a model possesses and its ability to generalize to new computational argumentation tasks.

Findings of the Association for Computational Linguistics: ACL 2026, pages 21846–21874

Based on the new benchmark, we evaluate five LLM families that capture different architectures and post-training methods. Our experiments in prompting indicate reasonable performance for some skills (an F_1-score of 0.486 in perspective assessment for few-shot prompting, on average), while still leaving notable room for improvement in general. Furthermore, LLMs' cross-task generalization appears limited, though models demonstrate better generalizability on specific argument reasoning and quality assessment tasks.

The main contributions of the paper are:

- A benchmark to develop and assess the computational argumentation skills of LLMs covering 33 datasets and 46 tasks.
- A quantitative analysis of LLMs' generalization across these tasks.
- Extensive insights into which skills LLMs are good at by default, and which require more resources or more sophisticated approaches.[1]

2 Related Work

Existing training and evaluation methodologies for LLMs show that argumentation enhances LLMs' capabilities on different tasks. Among these, Du et al. (2024) train LLMs to debate on six tasks that encompass factuality and mathematical reasoning. On all tasks, the debating models outperform models on their own, showing the merit of multi-agent debating for factuality and reasoning. Liang et al. (2024) similarly demonstrate the advantage of multi-agent debate over self-reflection for machine translation and unintuitive arithmetic reasoning . In contrast, Lin et al. (2024) stress that the ability of LLMs to self-criticize is crucial for logic-centric tasks, such as code generation and arithmetic reasoning. Eskandari Miandoab and Sarathy (2024) compare argument generation to chain-of-thought prompting by letting an LLM generate arguments for candidate answers and then picking the answer with the best argument. They demonstrate the benefit of argument generation on bias detection datasets, such as StereoSet (Nadeem et al., 2021) and DiFair (Zakizadeh et al., 2023). Khan et al. (2024) advise an approach where two LLMs debate to convince an LLM-based judge with the right answer in reading comprehension. The experiments show that even weaker judges can improve the ability of stronger models in reading comprehension. Altogether, prior works thus underline the potential of systematically evolving LLMs toward computational argumentation skills, for which we lay the ground in this paper.

Computational argumentation studies skills such as mining arguments from natural language text (Stab and Gurevych, 2017), assessing their perspectives (Bar-Haim et al., 2020), quality (Wachsmuth et al., 2017), and reasoning (Habernal et al., 2018a) as well as generating arguments (Hua et al., 2019). Existing benchmarks for computational argumentation focus on single or a few tasks. Feger et al. (2025) combine multiple argument mining datasets to systematically test the models' generalization capacities, showing their limited generalizability over argument mining datasets. A similar benchmark is proposed by Gemechu et al. (2024) for argument relation identification. Beyond their restriction to argument mining, neither of the benchmarks targets LLMs nor includes an LLM in their evaluation.

The landscape of benchmarks for LLMs is also focused on single or a few tasks (Pietroń et al., 2025; Abkenar et al., 2024). Gurjar et al. (2025) evaluate LLMs at mapping arguments in long discussions to key points. A similar benchmark in terms of focus is Dhole et al. (2025) which uses LLM judges to evaluate retrieval-augmented argument generation. The most comprehensive benchmark of LLMs in terms of genre is proposed by Gemechu et al. (2025), which is focused on the task of identifying a missing premise or conclusion in an argumentation in seven genres. Compared to these benchmarks, the breadth of tasks in our benchmark largely exceeds any existing study. Moreover, we design a setting that evaluates LLMs' ability to generalize to unseen tasks and enables the use of model-independent measures (noting reliability limits for argument generation).

Similar to benchmark studies, LLM-based approaches to computational argumentation are still few, mostly focusing on single tasks or specific genres. Lin et al. (2023) explore instruction finetuned LLMs for argument generation, finding that they largely outperform standard models on argumentation datasets. Chen et al. (2024) investigate the effectiveness of LLMs at argument mining and generation, while Deshpande et al. (2024) study to what extent LLM-generated context information benefits quality assessment, and Mouchel et al. (2025) align the argument generation skills of an LLM towards reasoning skills. In recent work,

[1] Our experiment code can be accessed here: https://github.com/webis-de/argbench

Stahl et al. (2025) develop a specialized instruction fine-tuning process to generalize LLMs' capabilities across computational argumentation tasks. While they release the seed tasks used in their self-instruct (Wang et al., 2023) pipeline, this data does not cover the full spectrum of tasks. As such, all these approaches may serve as starting points for different aspects of our benchmark, which covers 46 computational argumentation tasks in total.

3 The ArgBench Benchmark Dataset

A key advantage of training LLMs at large scale to follow instructions is the generalization to unseen tasks enabled thereby (Ouyang et al., 2022). The generalization capabilities of LLMs are also crucial in computational argumentation, given the richness of aspects studied in this research area and the wide spectrum of genres that argumentation is relevant to. Hence, assessing the performance of LLMs on computational argumentation tasks requires a representative set of tasks.

To this end, we resort to the survey of Lauscher et al. (2022), which groups tasks into four skills: argument mining, argument assessment, argument reasoning, and argument generation. However, we decided to divide assessment into argument perspective assessment and argument quality assessment, since these skills capture conceptually different aspects of argumentation. Based on this organization, we collected 46 tasks that cover many genres and a broad spectrum of how computational argumentation tasks are defined. Most of these come from the time before the LLM era. However, despite its different workings, we still expect a convincing LLM to be able to tackle them effectively.[2]

Table 1 lists the 46 benchmark tasks we derived from 33 datasets. More details about the datasets, their sizes, and their covered nine genres are listed in Table 8 (Appendix). For all datasets, we uniformly defined the tasks as prompts that an LLM gets to complete with the appropriate answer. To ensure the quality of the benchmark, we employed consistent terminology, format, and instruction layouts and used the same evaluation measure for the same type of tasks. Whenever possible, we introduced unknown terminology as defined in the respective papers of the datasets. Table 1 outlines the input and output for each task. The task definitions as well as the input and output extraction and formatting can be found in Appendix A.2, with exemplary task definitions in Tables 10 to 13.

According to how they are defined, the tasks fall into three groups: *classification*, *generation*, and *segmentation*. In the segmentation tasks, the model splits a document into multiple spans with their labels. In argument unit segmentation, for example, a document is split into multiple spans which are prepended with `argumentative` or `non-argumentative`, indicating whether the span is part of an argument or not. In the classification tasks, the model labels an argument or an argument unit with one of multiple labels (e.g., `low` or `high` quality). In the generation tasks, the model generates a text segment (e.g., an argument) for a given input (e.g., a topic).

Exemplarily, the prompt for the classification task *suboptimal claim detection* (Skitalinskaya and Wachsmuth, 2023) looks as follows:

> `Judge if the following claim can be improved by revising it. Possible outputs:` **`Improvable`** `if revision should be made,` **`Non-Improvable`** `if no revision is necessary. Only output Improvable or Non-Improvable.`

3.1 Evaluation Setup

As an evaluation metric, we use macro F_1-score for all segmentation and classification tasks. For segmentation tasks, we measure the F_1-score for each positive class while considering a true positive in case of an exact match between an output span and a ground-truth span. We return the macro-average of all positive labels in case a task maps spans of text to multiple positive labels, for example argumentative in case of unit segmentation (Ajjour et al., 2017). For generation tasks, we also need an automated metric despite its limitations. For meaningful insights, we utilize BertScore (Zhang et al., 2020) to capture the semantic similarity between the ground-truth and generated text.

We adopt two evaluation setups that are tailored to existing learning methods for LLMs: *prompting* and *leave-one-task-out*. For both, we utilize the original test sets, when available. Otherwise, we split the dataset into training, validation, and test sets with a 60/20/20 ratio, guaranteeing no topic overlap between the sets to foster generalization across topics. As the size of the datasets is heavily

[2]The full benchmark can be found here: `https://github.com/webis-de/argbench-data`

Task	Input	Output	Source
Argument Mining			
Conclusion Extraction	Sentence*	`conclusion` or `no-conc.`	Poudyal et al. (2020)
Premise Extraction	Sentence*	`premise` or `no-premise`	Poudyal et al. (2020)
Premise Extraction	Motion, Evidence	`accept` or `reject`	Ein-Dor et al. (2020)
Relation Detection	Sentence pair*	`related` or `not-related`	Poudyal et al. (2020)
Relation Identification	Unit pair*	`attack` or `support`	Peldszus (2014)
Relation Identification	Unit pair*	`attack` or `support`	Stab and Gurevych (2017)
Relation Identification	Unit pair*	`attack` or `support`	Skeppstedt et al. (2018)
Relation Identification	Argument pair	3 Labels (e.g., `attack`)	Menini et al. (2018)
Relation Identification	Unit pair*	`evidence` or `reason`	Park and Cardie (2018)
Unit Classification	Argument unit*	3 Labels (e.g., `claim`)	Stab and Gurevych (2017)
Unit Classification	Argument unit*	6 Labels (e.g., `anecdote`)	Al-Khatib et al. (2017)
Unit Classification	Argument unit*	5 Labels (e.g., `fact`)	Park and Cardie (2018)
Unit Segmentation	Document	List of (`arg.` or `non-arg.`, Span)	Habernal and Gurevych (2015)
Unit Segmentation	Document	List of (`arg.` or `non-arg.`, Span)	Al-Khatib et al. (2016)
Unit Segmentation	Document	List of (`arg.` or `non-arg.`, Span)	Stab and Gurevych (2017)
Unit Segmentation	Document	List of (`arg.` or `non-arg.`, Span)	Poudyal et al. (2020)
Argument Perspective Assessment			
Argument Similarity	Argument pair	4 Labels (e.g., `dissimilar`)	Reimers et al. (2019)
Aspect Detection	Argument unit	List of (`aspect` or `no-asp.`, Span)	Schiller et al. (2021)
Frame Identification	Argument, Frame	`match` or `no-match`	Ajjour et al. (2019)
Key Point Matching	Argument, Key Point	`match` or `no-match`	Bar-Haim et al. (2020)
Stance classification	Topic, Conclusion	`pro` or `con`	Bar-Haim et al. (2017)
Stance classification	Topic, Conclusion	`pro` or `con`	Stab et al. (2018)
Argument Quality Assessment			
Argument Ranking	Premise pair	`better` or `worse`	Gretz et al. (2020)
Argument Ranking	Argument pair	`better` or `worse`	Gleize et al. (2019)
Argument Ranking	Conclusion pair	`better` or `worse`	Skitalinskaya et al. (2021)
Argument Rating	Argument	`Low`, `medium`, or `high`	Wachsmuth et al. (2017)
Argument Rating	Argument	6 Labels (e.g., `very high`)	Ng et al. (2020)
Controversy Scoring	Argument	3 Labels (e.g., `very contr.`)	Habernal et al. (2018b)
Reasonableness Scoring	Argument	3 Labels (e.g., `neutral`)	Habernal et al. (2018b)
Suboptimal Claim Det.	Claim	`improvable` or `not-imp.`	Skitalinskaya and Wachsmuth (2023)
Claim Improve. Sugg.	Claim, Quality Issue	`match` or `no-match`	Skitalinskaya and Wachsmuth (2023)
Argument Reasoning			
Ad-hominem Detection	Argument	`ad-hominem` or `not-ad-hom.`	Habernal et al. (2018b)
Fallacy Detection	Conclusion	13 Labels (e.g., `ad-hominem`)	Jin et al. (2022)
Fallacy Detection	Argument unit	List of (7 Fallacies , Spans)	Goffredo et al. (2023)
Scheme Classification	Argument	7 Labels (e.g., `false cause`)	Saha and Srihari (2023b)
Warrant Identification	Claim, Two Warrants	`warrant 1` or `warrant 2`	Habernal et al. (2018a)
Argument Generation			
Argument Generation	Topic, Stance, Facts	Argument	Saha and Srihari (2023b)
Argument Summariz.	Argument	Topic	Roush et al. (2024)
Claim Generation	Topic Stance pairs	Argument	Alshomary et al. (2021)
Claim Optimization	Conclusion	Conclusion	Skitalinskaya et al. (2023)
Counterargument Gen.	Argument	Argument	Hua and Wang (2018)
Counterargument Gen.	Argument	Argument	Hua et al. (2019)
Warrant Generation	Conclusion, Premise	Warrant	Bhagavatula et al. (2020)
Warrant Generation	Conclusion, Premise	Warrant	Habernal et al. (2018a)
Warrant Generation	Conclusion, Premise	Warrant	Boltužić and Šnajder (2016)
Warrant Generation	Conclusion, Premise	Warrant	Becker et al. (2020)

Table 1: The 46 computational argumentation tasks in our benchmark, with modeled inputs and output. The output of classification tasks are labels with 1–2 words and encode a class (e.g., `pro` or `con`). For segmentation tasks, the output is a list of labels and the labeled text spans. For generation tasks, the output is a text span that represents an argumentative concept (e.g., Conclusion). *We prepend the input with a context window that contains the input.

skewed, we sample 1000 instances from validation and test sets whose size exceeds this threshold. Similarly, we sample 3000 instances from training sets whose size is larger than 3000.

Prompting In the prompting setup, an LLM is evaluated on the test sets of the tasks without fine-tuning them. This setup evaluates an LLM's out-of-the-box computational argumentation abilities acquired during its training. The score of a skill in ArgBench is the macro-average of the scores of all the tasks under the skill. The size of the test sets for all tasks is 33,795 instances. Table 9 (Appendix) lists the number of instances for each skill.

Leave-One-Task-out Here, we fine-tune the LLM on the training sets of all tasks except one, on which we evaluate the model. This setting assesses the ability of the LLM to learn computational argumentation skills and to generalize to unseen tasks. As unseen target tasks, we randomly sampled one task from each skill. The target tasks are: *unit segmentation* on Web Discourse (Habernal and Gurevych, 2015), *argument rating* (Wachsmuth et al., 2017), *argument similarity* (Reimers et al., 2019), *ad-hominem detection* (Habernal et al., 2018b), and *counterargument generation* (Hua and Wang, 2018). The remaining tasks are used for training. As a validation task, we select stance classification (Stab et al., 2018).

4 Experiments

In this section, we evaluate the performance of state-of-the-art LLMs in both setups of the benchmark, *prompting* and *leave-one-task-out*. We start by introducing the LLMs used in the experiments and then report on the prompting techniques used for each setup, before we come to the results and a subsequent manual evaluation.

LLMs We cover a total of ten state-of-the-art open-weight instruction fine-tuned LLMs from five model families of different sizes as well as one closed-weight LLM for comparison:

- `Mistral` (Jiang et al., 2023) with 7 and 22 billion parameters, and `Mixtral-8x7b` (Jiang et al., 2024) with 56 billion parameters
- `Qwen3` (Yang et al., 2025) with 4 and 32 billion parameters
- `Llama-3.1/3.3` (Dubey et al., 2024) with 8 and 70 billion parameters
- `DeepSeek-R1` (DeepSeek-AI et al., 2025) with 7 and 32 billion parameters
- `Phi-3.5-MoE-7.6b` (Abdin et al., 2024) with 7.6 billion parameters
- `GPT-4.1` (OpenAI et al., 2024), estimated to have 1.8 trillion parameters

We used `huggingface.co` to load all open-weight LLMs. For the prompting experiments, we load the models with `bfloat16` precision. The exact model versions can be found in Table 16 in the Appendix. To foster reproducibility, we run all experiments with a fixed seed: 1516.

4.1 Prompting Setup

LLMs enable dedicated prompting techniques for a given task, such as *few-shot* and *chain-of-thought* prompting. Their impact on computational argumentation tasks is largely unexplored. To close this gap, we employ respective prompting approaches tailored to the benchmark tasks:

Zero-Shot For each task, we prompt the model with its definition followed by the instance. For exemplary prompts, see Tables 10–13 (appendix).

Few-Shot We randomly select examples from the training set and append them to the task definition. We sample four examples in our experiments since the average label count in the classification tasks is 3.13. This allows one example per label on average.

Chain-of-Thought We expect chain-of-thought prompting (Wei et al., 2022) to be a promising technique for computational argumentation where tasks often compose multiple steps by concept. For example, to generate a counterargument to an argument, concrete steps can include extracting the main conclusion and premise of the argument, attacking each of them separately, and finally synthesizing the rebuttal. To trigger chain-of-thought across tasks while clearly delineating the final output from the reasoning steps, we add the following instruction after the task description: "Think step by step and prepend your output with Output:"

4.2 Leave-One-Task-Out Setup

For this setup, we fine-tuned LLMs with 4-bit precision using LoRa (Hu et al., 2022) in either of two training settings:

Training on All Other Tasks We fine-tuned each open-weight LLM on all tasks except for the target task using the respective instructions.[3] Then, we evaluated it on the target task. On the validation task, we optimized the learning rate and early stopping threshold using the Tree-structured Parzen Estimator (Watanabe, 2023). Details on hyperparameter optimization are in Appendix A.3, including tested ranges (Table 14) and the best values (Table 18).

Training on the Target Task To assess the generalizability of LLMs to unseen tasks, we evaluate LLMs on each target task, after fine-tuning them

[3] Due to limited resources, we excluded `Llama-3.3-70b` and `Mixtral-8x7b` in the leave-one-task-out-experiments.

LLM	Prompt	Mining	Perspective	Quality	Reasoning	Generation	Macro
`DeepSeek-R1-7b`	Zero-shot	0.306	0.329	0.383	0.226	0.789	0.407
	Few-shot	0.308	0.334	0.395	0.248	0.791	0.415
	Chain-of-thought	0.373	0.422	0.389	0.270	0.814	**0.454**
`DeepSeek-R1-32b`	Zero-shot	0.352	0.362	0.373	0.292	0.820	0.440
	Few-shot	0.333	0.363	0.402	0.286	0.807	0.438
	Chain-of-thought	0.406	0.478	0.397	0.274	0.832	**0.477**
`Llama-3.1-8b`	Zero-shot	0.388	0.486	0.377	0.387	0.836	**0.495**
	Few-shot	0.367	0.465	0.358	0.332	0.834	0.471
	Chain-of-thought	0.396	0.492	0.367	0.389	0.827	0.494
`Llama-3.3-70b`	Zero-shot	0.490	0.552	0.419	0.524	0.836	0.564
	Few-shot	0.527	0.620	0.418	0.526	0.844	**0.587**
	Chain-of-thought	0.460	0.531	0.439	0.396	0.819	0.529
`Mistral-7b`	Zero-shot	0.385	0.331	0.360	0.356	0.832	0.453
	Few-shot	0.434	0.316	0.344	0.353	0.844	0.458
	Chain-of-thought	0.441	0.388	0.404	0.351	0.831	**0.483**
`Mistral-22b`	Zero-shot	0.431	0.409	0.335	0.345	0.834	0.471
	Few-shot	0.400	0.463	0.311	0.463	0.847	**0.497**
	Chain-of-thought	0.400	0.474	0.353	0.430	0.829	0.497
`Mixtral-8x7b`	Zero-shot	0.427	0.556	0.332	0.395	0.833	0.508
	Few-shot	0.443	0.605	0.390	0.398	0.837	**0.535**
	Chain-of-thought	0.409	0.523	0.356	0.389	0.829	0.501
`Qwen3-4b`	Zero-shot	0.378	0.476	0.332	0.336	0.839	0.472
	Few-shot	0.406	0.494	0.367	0.380	0.847	**0.499**
	Chain-of-thought	0.375	0.484	0.407	0.278	0.809	0.471
`Qwen3-32b`	Zero-shot	0.415	0.533	0.394	0.484	0.837	0.533
	Few-shot	0.501	0.588	0.422	0.495	0.843	**0.570**
	Chain-of-thought	0.450	0.552	0.430	0.358	0.820	0.522
`Phi-3.5-MoE-7.6b`	Zero-shot	0.385	0.410	0.321	0.366	0.825	0.461
	Few-shot	0.325	0.429	0.329	0.332	0.840	0.451
	Chain-of-thought	0.399	0.398	0.358	0.388	0.818	**0.472**
`GPT-4.1`	Zero-shot	0.536	0.607	0.440	0.556	0.849	0.597
	Few-shot	0.597	0.664	0.477	0.560	0.865	**0.633**
	Chain-of-thought	0.542	0.603	0.436	0.465	0.840	0.577

Table 2: **Results for prompting setup.** Performance of all LLMs in zero-shot, few-shot, and chain-of-thought prompting, on all tasks of each of the five skills: argument *mining*, *perspective* and argument *quality* assessment, argument *reasoning*, and *generation*. *Macro* is the macro average over all skills (best macro per LLM bold).

on the training set of that task. This gives us an upper bound on tackling the task with LLMs.

4.3 Results

Prompting Table 2 shows the results of all eleven LLMs for each of the five skills as well as their macro average. Generally, we see that GPT-4.1 performs best, but the largest open-weight models come rather close. In zero-shot prompting, the larger models of `Qwen3`, `Llama`, `Mistral`, and `DeepSeek-R1` outperform their smaller counterparts on the macro score. This suggests that model size benefits computational argumentation tasks by default. The gain of larger over smaller models is lowest for argument quality assessment with a drop of –0.025 for `Mistral` and a maximum of 0.062 for `Qwen3`. In contrast, increasing the model size achieves the highest gain on argument mining (0.037–0.102) and perspective assessment (0.033–0.078). When we increase the model parameters by a factor of 4 or more, LLMs achieve even higher performance on argument reasoning (0.066–0.148).

Few-shot prompting achieves the best macro-average score for large models (32 billion parameters or more), except `DeepSeek-R1-32b` which is optimized specifically for reasoning. Across all these large models, few-shot prompting seems to be the most effective prompting technique for argument reasoning and generation. In contrast, it performs worse than other prompting techniques on argument quality assessment for the `Llama` and `Mistral` families. Our error analysis of the output of `Mistral-22b` in Appendix A.1 shows a tendency to prefer positive labels (e.g., `better` vs `worse`), especially on argument ranking tasks.

For argument quality assessment, chain-of-thought outperforms zero-shot prompting for all models except `Llama-3.1-8b` and `GPT-4.1`. Surprisingly, however, its performance remains similar or subpar to zero-shot prompting on argument reasoning tasks for the `Llama`, `Mistral`,

Model	Training	Mining	Perspective	Quality	Reasoning	Generation	Macro
		Unit Segmentation	Argument Similarity	Argument Rating	Ad-hominem Detection	Counterargument Generation	
DeepSeek-R1-7b	On the target task	0.141	0.490	0.512	0.844	0.797	0.557
	On all other tasks	0.138	0.462	0.519	0.819	0.784	0.544
DeepSeek-R1-32b	On the target task	0.242	0.491	0.513	0.903	0.803	0.590
	On all other tasks	0.207	0.381	**0.562**	0.883	0.789	0.564
Llama-3.1-8b	On the target task	0.210	0.469	0.377	0.893	0.800	0.550
	On all other tasks	0.159	0.273	0.419	0.872	0.777	0.500
Mistral-7b	On the target task	0.226	0.483	0.542	0.897	0.821	0.590
	On all other tasks	0.191	0.207	0.518	0.840	0.784	0.508
Mistral-22b	On the target task	**0.270**	**0.636**	0.554	**0.912**	**0.825**	**0.639**
	On all other tasks	0.167	0.468	0.529	0.890	0.791	0.569
Qwen3-4b	On the target task	0.235	0.480	0.381	0.876	0.801	0.555
	On all other tasks	0.134	0.447	0.451	0.879	0.789	0.540
Qwen3-32b	On the target task	0.261	0.474	0.388	0.895	0.821	0.568
	On all other tasks	0.138	0.248	0.280	0.803	0.781	0.450
Phi-3.5-MoE-7.6b	On the target task	0.198	0.445	**0.562**	0.872	0.791	0.574
	On all other tasks	0.173	0.250	0.477	0.818	0.781	0.500
GPT-4.1	No Training	0.212	0.441	0.504	0.867	0.821	0.569

Table 3: **Results for leave-one-task-out setup.** Performance of all evaluated LLMs on five target tasks (Habernal and Gurevych, 2015; Reimers et al., 2019; Wachsmuth et al., 2017; Habernal et al., 2018b; Hua and Wang, 2018), when fine-tuning a LoRa adapter *on the target task*'s training set or *on all other tasks*' training sets. *Macro* is the macro average over all tasks. The best result per column is marked bold.

and Qwen3 families. Our error analysis of chain-of-thought in argument reasoning in Appendix A.1 reveals that the error source often lies in difficulties in processing emotional language and instruction following failures. Another challenge is the tendency of Qwen3-32b to enter unfinished reasoning chains (78% instances of argument reasoning tasks). On argument mining, chain-of-thought is the most effective prompting technique for all small models (8 billion parameters or less) except Qwen3-4b. In contrast, few-shot prompting dominates other prompting techniques for large models on argument mining except DeepSeek-R1-32b.

We conclude that argument quality assessment seems the most challenging argumentation skill for LLMs. This is demonstrated by the low performance of LLMs on this skill in general as well as the small gain from increasing the model size or applying prompting techniques. The low performance of applying chain-of-thought on argument reasoning for three model families signals the inadequacy of step-by-step reasoning techniques to characterize argument reasoning patterns and gaps.

Leave-One-Task-Out Table 3 lists the results of eight open-weight LLMs on each target task when training on its training set (in-task) and when training on the training sets of all remaining 45 tasks (leave-one-task-out). For comparison, we also show GPT-4.1 without training.

Mistral-22b achieves the highest macro-average score over the five tasks (0.639), but F_1-scores of only 0.270 and 0.554 on unit segmentation (Habernal and Gurevych, 2015) and argument rating (Wachsmuth et al., 2017) respectively. These results indicate that, even with fine-tuning on the target, certain basic computational argumentation tasks remain a challenge for LLMs. When trained on other tasks, Mistral-22b achieves the best performance on argument similarity, ad-hominem detection, and counterargument generation, while DeepSeek-R1-32b performs the best with F_1-scores of 0.207 on unit segmentation and 0.562 on argument rating. On ad-hominem detection, both DeepSeek-R1-32b and Mistral-22b show only a drop of 0.02 from the in-task performance.

On argument rating, the DeepSeek-R1 family achieves F_1-score gains of 0.005 to 0.049 points when utilizing training on other tasks compared to target-task training. This suggests the potential of some LLMs to transfer the skill of evaluating argument quality to novel tasks. In contrast, for unit segmentation, the results of all models are consistently lower when trained on other tasks, dropping by 0.003 to 0.123 points. A similar task transfer challenge can be observed in argument similarity (a drop of 0.028 to 0.276 points) and counterargument generation (decreases of 0.01 to 0.04 points).

Taken together, the experiments highlight the limited generalizability of LLMs across computational argumentation tasks. While models such as `DeepSeek-R1-32b` and `Mistral-22b` generalize well to ad-hominem detection and argument rating, their performance remains lower than that of fine-tuning models on other target tasks, suggesting a limited ability to generalize to unseen computational argumentation tasks. Challenging skills for generalization, according to our experiments, are argument mining and generation, as well as perspective assessment. This hints at the ability of LLMs to generalize argumentation concepts to tasks related to evaluating and judging argumentation rather than generating arguments or extracting arguments and their perspectives.

4.4 Manual Evaluation

Automatic evaluation is of limited reliability for generation tasks. To gain further insights, we manually evaluated the output of the LLMs on the counterargument generation task of (Hua and Wang, 2018). This task aims at synthesizing counterarguments that properly challenge the arguments posted by authors on ChangeMyView by leveraging supporting information retrieved from Wikipedia to craft the arguments. We sampled 50 arguments from the test set of the task and generated counterarguments with four selected models (`Mistral-22b`, `Mistral-7b`, `Qwen3-32b`, and `Phi-3.5-MoE-7.6b`) in the leave-one-task-out setup. These were the top four models on the counterargument generation task, and they cover different model sizes, architectures, and regions of origin. The setup yielded $50 \cdot 2 \cdot 4 = 400$ input triples (comprising the argument, supporting information, and generated counterarguments). Three annotators from Upwork evaluated these triples on three dimensions using a 5-point Likert scale (where a higher value indicates higher quality): *argumentativeness*, *countering*, and *relevance*. More information about the annotation process is in the Appendix A.4.1, including a screenshot of the annotation interface in Figure 1.

Table 4 presents the mean ratings, showing that `Mistral-22b` is judged best, both when trained on the target task and on all other tasks. This corroborates our findings in Section 4.2. Notably, the performance generally decreases across all dimensions without in-task training: for argumentativeness by 0.126 to 0.807, for countering by 0.040 to 0.686, and for relevance by 0.450 to 0.560. These results underscore the importance of task-specific training for optimal performance on counterargument generation. Furthermore, the higher cross-task performance on the countering dimension compared to argumentativeness suggests that LLMs without task-specific training are slightly better at generating opposing stances than at grounding these stances as well-reasoned arguments.

Model	**Training**	**Arg**	**Cou**	**Rel**
`Mistral-7b`	On the target task	3.120	3.120	2.533
	On all other tasks	2.460	2.753	2.020
`Mistral-22b`	On the target task	**3.487**	**3.493**	**2.670**
	On all other tasks	2.680	2.807	2.220
`Qwen3-32b`	On the target task	3.227	3.187	2.647
	On all other tasks	2.513	2.740	2.087
`Phi-3.5-MoE -7.6b`	On the target task	2.253	2.473	1.980
	On all other tasks	2.127	2.433	2.027

Table 4: **Manual evaluation results.** Mean ratings of the *arg*umentativeness, *cou*ntering, and *rel*evance of the counterarguments generated by four LLMs for 50 test instances from Hua and Wang (2018). The LLMs were evaluated after training either on the training set of the target task, or on the training sets of all other tasks. The best result per criterion is marked bold.

5 Follow-up Experiment: Skill Transfer

The results of the leave-one-task-out experiments indicate that task-specific training is required for optimal performance on some tasks, such as counterargument generation (Hua and Wang, 2018). In this section, we therefore study which training skill most effectively transfers to each of the target tasks. This provides an intuition for how to tailor an optimal training set for a given target task.

5.1 Skill Transfer Setup

To explore skill transfer, we fine-tune a LoRa adapter on the training data of each skill, fine-tune it further on the target task, and evaluate it on the test set of the target task. In this way, we can assess which skill prepares LLMs best for unseen tasks. We conduct this experiment using `Mistral-22b`, since it showed the best generalization capabilities across the evaluated models in the leave-one-task-out experiment. We employ a continual learning setup, where we fine-tune the LoRa adapter for five epochs on the training skill and further fine-tune it on the target task.[4] For comparison, we compare the results of `Mistral-22b` in the skill transfer experiment to the results of `Mistral-22b` in the in-task and leave-one-task-out experiment.

[4]We utilize the hyperparameters for the leave-one-task-out experiment for fine-tuning on a training skill and the in-task experiments for fine-tuning on the target tasks.

Training Skill	Mining	Perspective	Quality	Reasoning	Generation	Macro
	Unit Segmentation	**Argument Similarity**	**Argument Rating**	**Ad-hominem Detection**	**Counterargument Generation**	**All**
On the target task	0.270	0.636	0.554	0.912	0.825	0.639
On all other tasks	0.167	0.468	0.529	0.890	0.791	0.569
Argument Mining	**0.194**	0.479	**0.552**	0.908	**0.803**	0.587
Perspective Assessment	0.183	**0.598**	0.540	0.893	0.799	**0.603**
Argument Quality Assessment	0.118	0.541	0.540	0.904	0.801	0.580
Argument Reasoning	0.097	0.486	0.534	**0.909**	0.800	0.565
Argument Generation	0.058	0.524	0.544	0.906	0.799	0.566

Table 5: **Skill transfer results.** Performance of `Mistral-22b` on the five target tasks, when fine-tuning a LoRa adapter on the training sets all tasks of each skill and then on the training set of the target task. The five target tasks are the same as in Table 3.

5.2 Results

Table 5 compares the results of `Mistral-22b` of the skill transfer experiment to the in-task and leave-one-task-out results. We observe that training on the same skill of the target task achieves the best performance for unit segmentation, argument similarity, and ad-hominem detection, with F_1-scores of 0.194, 0.598, and 0.909, respectively. However, on counterargument generation and argument rating, argument mining is a better transfer skill than the target task's skill. This shows the importance of training on argument extraction tasks for counterargument generation and quality assessment tasks.

Since training on the same training skill of a target task is expected to bring good results on the target task, we turn our focus to the best training tasks while excluding the same training skill. For unit segmentation (Habernal and Gurevych, 2015), perspective assessment is the best transfer skill when excluding argument mining, outperforming training on all other tasks (0.183 vs. 0.167). This shows that tasks related to understanding the perspective of an argument are effective training tasks for extracting argument units. This is likely because argument units typically feature perspective-taking language, such as stance or frames.

For argument similarity, quality assessment is the most effective transfer skill, exceeding the performance obtained by training on all training tasks (0.468 vs. 0.541). This likely stems from the pairwise framing of some quality tasks, which resembles the input format of argument similarity. By examining the best training skill for ad-hominem detection, we notice that argument mining is the most effective transfer skill, with a drop of 0.001 in comparison to training on the same target tasks' skill (argument reasoning). The competitive performance achieved by fine-tuning on argument mining for ad-hominem detection positions argument mining as the best training skill on four of the five target tasks (all but argument similarity).

6 Conclusion

The ability of LLMs to understand, analyze, and generate arguments is essential to counter societal harms, to identify fallacious reasoning, and much more. To enable systematic training and evaluation of computational argumentation skills, we introduce ArgBench, a benchmark tailored to LLMs that provides a unified evaluation protocol for 46 existing tasks from five skills: argument mining, perspective assessment, argument quality assessment, argument reasoning, and argument generation. Thereby, it serves as the most comprehensive computational argumentation benchmark available to date. While most tasks have been proposed before the LLM era, we still expect a convincing LLM to tackle them effectively.

In two experimental setups, we have presented empirical insights into the few-shot and chain-of-thought prompting performance of five LLM families on the benchmark and into their generalization ability of computational argumentation skills obtained from fine-tuning. Our prompting results suggest that performance increases with model size, especially for mining, assessing perspectives, and reasoning. In contrast, assessing argument quality is the most challenging skill for LLMs, with chain-of-thought being best for it. Cross-task generalization remains challenging, but seems promising for some quality assessment and reasoning tasks. Manual evaluation of LLMs' output in one generation task corroborates our finding that task-specific fine-tuning is needed for optimal performance in some tasks. Future research on ArgBench may study how to select the best training tasks for a target task.

7 Limitations

The main goal of this paper is to provide a comprehensive and unified benchmark for LLM-based approaches to computational argumentation. To reach this goal, we had to make a few compromises that we cover in the following:

Unified Modeling of Tasks Computational argumentation is highly diverse in input and output requirements, varying notably across different datasets for the same task. To provide a common LLM-adjusted interface to all tasks, we needed to slightly reframe certain tasks, as indicated above. This included simplifying argument units to the sentence level, reducing the predefined labels for open tasks such as frame identification (Ajjour et al., 2019) to six labels, or approaching an argument ranking task (Gleize et al., 2019) in a comparative way. We refrained from using specific output formats, such as JSON, to avoid limiting approaches in what they create and not to favor model families optimized toward these formats.

Coverage of Tasks While we present the largest and most comprehensive collection of computational argumentation tasks to our knowledge, some existing datasets have not been included for different reasons. Some have application-specific argument models, such as context-dependent claim detection. (Levy et al., 2014). Others seem rather specific for a general benchmark, such as appropriateness rewriting (Ziegenbein et al., 2024), or their data is not allowed to be redistributed, such as for argument specificity classification (Durmus et al., 2019).

Coverage of LLMs The results of the baseline prompting and leave-one-task-out experiments presented above required extensive computation. For space and computational complexity reasons, we could not include model sizes that exceed 32 billion parameters and focused on studying models with different architectures and post-training techniques. To explore their general impact, we included five open-weight model families. While we contrast small and large models in this work, we cannot exclude that other model families or sizes behave differently.

Unified Evaluation of Outputs By nature, LLMs may provide any string as output and cannot be strictly forced to match the output format aimed for. This particularly holds for recent model families, such as `DeepSeek-R1`, pretrained to carry out chain-of-thought reasoning as part of their output generation. In our evaluation setup, we foster a clear output specification by ending with "Output:", as indicated above. Moreover, we used regular expressions to match the output of the different LLMs to the desired format for segmentation tasks (i.e., a list of label and span pairs separated by line breaks).

Coverage of Measures To further support a unified evaluation setting, we opted for as common and as few as possible evaluation measures in our benchmark. Four of the five argumentation skills we cover could be mapped to a setup that allows for macro F_1-score, which we see as appealing to obtain comparability even across the respective tasks. As for most families of text generation tasks, a reliable evaluation of argument generation would actually require human judgments, which is unfeasible in a benchmark. We thus decided to use one of the most common measures of semantic similarity (BertScore). Additionally, we conducted a targeted manual evaluation of LLM outputs for counterargument generation (Hua and Wang, 2018) to capture human perception of the generated text. We explicitly refrained from LLM judges in the evaluation, as they may be outdated soon and conceptually conflict with the idea of comparing different LLMs to each other.

Acknowledgments We thank Dmytro Bozhkov for his role in benchmark preparation and Roxanne El Baff for her thorough and insightful review of the paper.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, and 68 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

Mohammad Yeghaneh Abkenar, Weixing Wang, Hendrik Graupner, and Manfred Stede. 2024. Assessing open-source large language models on argumentation mining subtasks. *arXiv preprint arXiv:2411.05639*.

Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. Modeling frames in argumentation. 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 2922–2932, Hong Kong, China. Association for Computational Linguistics.

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128. Association for Computational Linguistics.

Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, and Benno Stein. 2017. Patterns of argumentation strategies across topics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1351–1357, Copenhagen, Denmark. Association for Computational Linguistics.

Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. A News Editorial Corpus for Mining Argumentation Strategies. In *26th International Conference on Computational Linguistics (COLING 2016)*, pages 3433–3443. Association for Computational Linguistics.

Milad Alshomary, Shahbaz Syed, Arkajit Dhar, Martin Potthast, and Henning Wachsmuth. 2021. Counter-argument generation by attacking weak premises. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1816–1827, Online. Association for Computational Linguistics.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance Classification of Context-Dependent Claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 251–261. Association for Computational Linguistics.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. From Arguments to Key Points: Towards Automatic Argument Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039. Association for Computational Linguistics.

Maria Becker, Katharina Korfhage, and Anette Frank. 2020. Implicit knowledge in argumentative texts: An annotated corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2316–2324.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Filip Boltužić and Jan Šnajder. 2016. Fill the gap! analyzing implicit premises between claims from online debates. In *Proceedings of the Third Workshop on Argument Mining*, pages 124–133.

Tuhin Chakrabarty, Aadit Trivedi, and Smaranda Muresan. 2021. Implicit premise generation with discourse-aware commonsense knowledge models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6247–6252. Association for Computational Linguistics.

Guizhen Chen, Liying Cheng, Anh Tuan Luu, and Lidong Bing. 2024. Exploring the potential of large language models in computational argumentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2309–2330, Bangkok, Thailand. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 2 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Darshan Deshpande, Zhivar Sourati, Filip Ilievski, and Fred Morstatter. 2024. Contextualizing argument quality assessment with relevant knowledge. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 316–326, Mexico City, Mexico. Association for Computational Linguistics.

Kaustubh Dhole, Kai Shu, and Eugene Agichtein. 2025. ConQRet: A new benchmark for fine-grained automatic evaluation of retrieval augmented computational argumentation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5687–5713, Albuquerque, New Mexico. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019. Determining relative argument specificity and stance for complex argumentative structures. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4630–4641, Florence, Italy. Association for Computational Linguistics.

Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2020. Corpus wide argument mining - A working solution. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7683–7691.

Kaveh Eskandari Miandoab and Vasanth Sarathy. 2024. "let's argue both sides": Argument generation can force small models to utilize previously inaccessible reasoning capabilities. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 269–283, Miami, Florida, USA. Association for Computational Linguistics.

Marc Feger, Katarina Boland, and Stefan Dietze. 2025. Limited generalizability in argument mining: State-of-the-art models learn datasets, not arguments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23900–23915, Vienna, Austria. Association for Computational Linguistics.

Debela Gemechu, , Ramon Ruiz-Dolz, , Henrike Beyer, , and Chris Reed. 2025. Natural language reasoning in large language models: Analysis and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3717–3741, Vienna, Austria. Association for Computational Linguistics.

Debela Gemechu, Ramon Ruiz-Dolz, and Chris Reed. 2024. ARIES: A general benchmark for argument relation identification. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. 2019. Are You Convinced? Choosing the More Convincing Evidence with a Siamese Network. In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 967–976.

Pierpaolo Goffredo, Mariana Espinoza, Serena Villata, and Elena Cabrio. 2023. Argument-based detection and classification of fallacies in political debates. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 11101–11112. Association for Computational Linguistics.

Shai Gretz, Yonatan Bilu, Edo Cohen-Karlik, and Noam Slonim. 2020. The workweek is the best time to start a family – a study of GPT-2 based claim generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 528–544, Online. Association for Computational Linguistics.

Omkar Gurjar, Agam Goyal, and Eshwar Chandrasekharan. 2025. Argcmv: An argument summarization benchmark for the llm-era. *arXiv preprint arXiv:2508.19580*.

Ivan Habernal and Iryna Gurevych. 2015. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2127–2137, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018a. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1930–1940. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018b. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 386–396. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Xinyu Hua, Zhe Hu, and Lu Wang. 2019. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2661–2672. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2018. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 219–230. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothé Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

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, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 7180–7198. Association for Computational Linguistics.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Anne Lauscher, Henning Wachsmuth, Iryna Gurevych, and Goran Glavaš. 2022. Scientia potentia est—on the role of knowledge in computational argumentation. *Transactions of the Association for Computational Linguistics*, 10:1392–1422.

John Lawrence and Chris Reed. 2016. Argument mining using argumentation scheme structures. In *Computational Models of Argument*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 379–390.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Jiayu Lin, Rong Ye, Meng Han, Qi Zhang, Ruofei Lai, Xinyu Zhang, Zhao Cao, Xuanjing Huang, and Zhongyu Wei. 2023. Argue with me tersely: Towards sentence-level counter-argument generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16705–16720, Singapore. Association for Computational Linguistics.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. Criticbench: Benchmarking llms for critique-correct reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1552–1587, Bangkok, Thailand. Association for Computational Linguistics.

Stefano Menini, Elena Cabrio, Sara Tonelli, and SerenaVillata. 2018. Never Retreat, Never Retract: Argumentation Analysis for Political Speeches. In *Proceedings of the Thirty-second Association for the Advancement of Artifical Intelligene (AAAI) Conference of Artifical Intelligence*, pages 4889–4896. AAAI Press.

Luca Mouchel, Debjit Paul, Shaobo Cui, Robert West, Antoine Bosselut, and Boi Faltings. 2025. A logical fallacy-informed framework for argument generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7296–7314, Albuquerque, New Mexico. Association for Computational Linguistics.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. Stereoset: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5356–5371. 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.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 2 others. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Joonsuk Park and Claire Cardie. 2018. A Corpus of e-Rulemaking User Comments for Measuring Evaluability of Arguments. In *Proceedings of the 2018 International Conference on Language Resources and Evaluation (LREC 2018)*.

Andreas Peldszus. 2014. Towards Segment-based Recognition of Argumentation Structure in Short Texts. In *Proceedings of the First Workshop on Argumentation Mining*, Baltimore, Maryland. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Proceedings of the 2015 European Conference on Argumentation: Argumentation and Reasoned Action (ECA 2015)*.

Marcin Pietroń, Rafał Olszowski, Jakub Gomułka, Filip Gampel, and Andrzej Tomski. 2025. A comprehensive study of llm-based argument classification: from llama through gpt-4o to deepseek-r1. *arXiv preprint arXiv:2507.08621*.

Prakash Poudyal, Jaromir Savelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. ECHR: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75, Online. Association for Computational Linguistics.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 567–578. Association for Computational Linguistics.

Allen Roush, Yusuf Shabazz, Arvind Balaji, Peter Zhang, Stefano Mezza, Markus Zhang, Sanjay Basu, Sriram Vishwanath, Mehdi Fatemi, and Ravid Shwartz-Ziv. 2024. Opendebateevidence: A massive-scale argument mining and summarization dataset. *CoRR*, abs/2406.14657.

Sougata Saha and Rohini Srihari. 2023a. Consolidating strategies for countering hate speech using persuasive dialogues. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 378–392, Goa University, Goa,.

Sougata Saha and Rohini K. Srihari. 2023b. Argu: A controllable factual argument generator. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8373–8388. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Aspect-controlled neural argument generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–396, Online. Association for Computational Linguistics.

Maria Skeppstedt, Andreas Peldszus, and Manfred Stede. 2018. More or less controlled elicitation of argumentative text: Enlarging a microtext corpus via crowdsourcing. In *Proceedings of the 5th Workshop on Argument Mining 2017 (ArgMining 2017)*, pages 155–163. Association for Computational Linguistics.

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. 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.

Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen, Lena Dankin, Lilach Edelstein, Liat Ein Dor, Roni Friedman-Melamed, Assaf Gavron, Ariel Gera, Martin Gleize, Shai Gretz, Dan Gutfreund, Alon Halfon, Daniel Hershcovich, and Ranit Aharonov. 2021. An autonomous debating system. *Nature*, 591:379–384.

Christian Stab and Iryna Gurevych. 2017. Parsing Argumentation Structure in Persuasive Essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic Argument Mining from Heterogeneous Sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 3664–3674, Brussels, Belgium. 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.

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. Association for Computational Linguistics.

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.

Shuhei Watanabe. 2023. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *CoRR*, abs/2304.11127.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Mahdi Zakizadeh, Kaveh Eskandari Miandoab, and Mohammad Taher Pilehvar. 2023. Difair: A benchmark for disentangled assessment of gender knowledge and bias. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 1897–1914. Association for Computational Linguistics.

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*.

Xiaofan Zheng, Minnan Luo, and Xinghao Wang. 2025. Unveiling fake news with adversarial arguments generated by multimodal large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7862–7869, Abu Dhabi, UAE. Association for Computational Linguistics.

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.

A Appendix

The appendix details the error analysis of the *quality assessment* and *argument reasoning* tasks (A.1), describes the task selection and normalization for ArgBench (A.2), and provides implementation details of our experiments (A.3). Finally, it lists experimental results on more models and provides more information about the manual evaluation of counterargument generation (A.4).

A.1 Error Analysis

The prompting experiments reveal substantial room for improvement across all evaluated models and skills. This is particularly evident in quality assessment and reasoning tasks, even for the highest-performing models. Table 6 lists the performance of `Mistral-22b`, `Llama-3.3-70b`, and `Qwen3-32b` across all tasks within these two skills. The results show that LLMs are good at warrant identification (Habernal et al., 2018a) and ad-hominem detection (Habernal et al., 2018b). On the other hand, LLMs struggle with reasonableness scoring (Habernal et al., 2018b) and reasoning tasks such as scheme classification (Saha and Srihari, 2023b) and fallacy detection (Goffredo et al., 2023; Jin et al., 2022). In the following, we describe typical errors exhibited by the three models.

Analysis of model outputs reveals that `Mistral-22b` exhibits a label bias toward the positive or high-magnitude labels in almost all tasks. Examples of these positive labels are `very controversial` for controversy scoring or `quite reasonable` for reasonableness scoring. The problem is especially pronounced in the three pairwise argument ranking tasks where `Mistral-22b` with zero-shot prompting selects `better` over `worse` in 2,669 out of 3,000 (89%) cases. This bias is further amplified in few-shot prompting where `Mistral-22b` predicts `better` in 2,801 out of 3,000 (93%) cases. `Llama-3.3-70b` predicts `better` in 1,471 (49%) cases in zero-shot-prompting and in 2,294 (76%) cases in few-shot-prompting, indicating a

Task (Dataset)	Mistral-22b			Qwen3-32b			Llama-3.3-70b		
	Zero-shot	Few-shot	CoT	Zero-shot	Few-shot	CoT	Zero-shot	Few-shot	CoT
Argument Quality Assessment									
Argument Ranking (Claim Revisions)	.388	.346	.421	.348	.442	.560	.570	.550	**.606**
Argument Ranking (IBM Evidence Quality)	.351	.350	.360	.619	.606	.602	.621	.365	**.664**
Argument Ranking (IBM Rank 30k)	.517	.477	.417	.507	.454	.515	.535	.513	**.546**
Argument Rating (Dagstuhl 15512)	.343	.255	.360	.467	**.502**	.381	.414	.433	.360
Argument Rating (GAQ)	.340	.209	.384	.388	.338	.204	.374	.262	**.404**
Improvement Suggestion (Claim Revisions)	.454	.502	.440	.448	.503	**.531**	.466	.508	.499
Controversy Scoring (CMV Ad-hominem)	.202	.256	.313	.327	.357	.307	.414	**.512**	.502
Reasonableness Scoring (CMV Ad-hominem)	.129	.134	.215	.175	**.282**	**.282**	.085	.171	.077
Suboptimal Claim Detection (Claim Revisions)	.296	.268	.268	.270	.316	**.484**	.293	.451	.292
Macro-average	.335	.311	.353	.394	.422	.430	.419	.418	**.439**
Argument Reasoning									
Ad-hominem Detection (CMV Ad-hominem)	.522	.815	.757	.825	.842	.643	.853	**.873**	.858
Fallacy Detection (ElecDeb60to20)	.027	.121	.044	.060	.075	**.126**	.086	.079	.068
Fallacy Detection (Logic)	.283	.368	.328	.402	.427	.214	.483	**.514**	.300
Scheme Classification (ArgU)	.242	.231	.268	.283	.253	.134	**.332**	.257	.203
Warrant Identification (SemEval'18)	.652	.779	.753	.850	.876	.673	.865	**.908**	.549
Macro-average	.345	.463	.430	.484	.495	.358	.524	**.526**	.396

Table 6: **Prompting error analysis.** Macro F_1-scores performance of `Mistral-22b`, `Llama-3.3-70b`, and `Qwen3-32b` across all argument quality assessment and argument reasoning tasks. The macro-average over the tasks of each skill, which represents the score for the skill, is also reported. The best model and prompting technique per task (row) is highlighted.

slightly lower bias than `Mistral-22b` towards positive labels as well.

In argument rating tasks (Ng et al., 2020; Wachsmuth et al., 2017), `Mistral-22b` tends toward the middle labels: `average` or `medium`. In argument rating (Wachsmuth et al., 2017), the models have to classify an argument on a quality dimension (e.g., Rhetoric) into either `low`, `average`, or `high`, indicating how good the argument is on a given dimension. `Mistral-22b` responds with the middle label in 1,666 out of 2,000 (83%) cases in zero-shot prompting in the two argument rating tasks. In few-shot prompting, `Mistral-22b` predicts `medium` in 1,135 out of 2,000 (57%) cases. The tendency of the model to select the middle label suggests the inclination of the model to pick the "safe" option. This bias may stem from the inherent subjectivity of argument rating, where argument quality often depends on the target audience.

We identify three primary error sources in chain-of-thought prompting: incomplete reasoning chains, difficulties in processing emotional language, and failures in instruction following.

Incomplete reasoning occurs when the model exhausts all the token budget without generating the final output. Specifically, `Qwen3-32b` entered unfinished reasoning loops in 5,845 out of 7,720 (76%) argument quality assessment instances and in 2,299 out of 2,953 (78%) argument reasoning instances.[5] Table 7 provides exemplary predictions for `Qwen3-32b`, `Mistral-22b`, and `Llama-3.3-70b` in fallacy detection (Jin et al., 2022). Instance 2 illustrates a typical error where `Qwen3-32b` iterates over multiple candidate fallacies, hesitates to assign the correct `appeal to emotion` label, and eventually stops while reflecting on an irrelevant candidate (`fallacy of extension`). This behavior may stem from excessive self-reflection, where `Qwen3-32b` exhausts the token budget by weighing multiple labels despite having initially identified the correct one. Additionally, the lengthy instructions required for fallacy detection and scheme classification (Jin et al., 2022; Saha and Srihari, 2023b; Goffredo et al., 2023) may exacerbate this token exhaustion.

Handling emotional language is another source of errors that LLMs struggle with in argument reasoning tasks. Instances 3 and 4 show arguments that exemplify an `appeal to emotion`, but `Llama-3.3-70b` and `Mistral-22b` incorrectly categorize them as `false dilemma` and `fallacy of relevance`, respectively. In Instance 3, `Llama-3.3-70b` focuses on the comparative structure of the argument that presents two extreme options, thereby overlooking the appeal

[5]Notice that in these cases we assigned a random label as the output of the model. We did not optimize the max output tokens for each model separately to have a fair comparison between the models.

Instructions	A fallacy a failure in reasoning which renders an argument invalid. Classify the following sentence into one of the following fallacies: [Definitions for fallacies are omitted for sake of space. Look up the repo for details] Think step by step and prepend your output with Output:
	Instance 1 (Correct Prediction)
Input	Sentence: The meteorologist predicted the wrong amount of rain for May. Obviously the meteorologist is unreliable.
Ground truth	`faulty generalization`
`Qwen3-32b`:	<think> Okay, let's see. The sentence is: "The meteorologist predicted the wrong amount of rain for May. Obviously the meteorologist is unreliable." I need to figure out which fallacy this is. First, the person is saying that because the meteorologist was wrong once (in May), they're unreliable. That sounds like a faulty generalization. Like, one instance isn't enough to conclude that the meteorologist is always wrong. But wait, maybe there's another angle. Is there an ad-hominem here? Because they're attacking the meteorologist's reliability instead of the argument? Hmm, but the argument is about the prediction. Wait, the person is using the incorrect prediction to say the meteorologist is unreliable. But if they're using a single instance to generalize, that's a faulty generalization. So maybe that's the main issue here. The person is taking one wrong prediction and concluding the whole person is unreliable. That's a classic case of hasty generalization, which is a type of faulty generalization. So I think the answer is Faulty Generalization. </think> Output: Faulty Generalization
	Instance 2 (Incorrect Prediction)
Input	Sentence: I really deserve an "A" on this paper, professor. Not only did I study during my grandmother's funeral, but I also passed up the heart transplant surgery, even though that was the first matching donor in 3 years.
Ground truth	`appeal to emotion`
`Qwen3-32b`:	<think> Okay, let's see. The user wants me to classify the given sentence into one of the listed fallacies. The sentence is: "I really deserve an "A" on this paper, professor. Not only did I study during my grandmother's funeral, but I also passed up the heart transplant surgery, even though that was the first matching donor in 3 years." First, I need to understand what the speaker is doing here. They're trying to argue that they deserve an A by citing extreme examples of their dedication. Studying during a funeral and refusing a life-saving surgery. But these examples are clearly exaggerated and possibly false. The key here is whether the argument is using irrelevant or extreme examples to support the claim. Looking at the fallacy options. Ad-hominem is attacking the person, which doesn't fit here. Appeal to emotion might be a possibility because they're trying to evoke sympathy, but the main issue is the use of extreme, possibly false, examples. Faulty Generalization is about drawing conclusions from insufficient samples, but that's not the case here. False Causality is implying a cause-effect without evidence, but the argument isn't about causation. The Fallacy of Extension is attacking an exaggerated version of the opponent's position, but here the speaker is exaggerating their own actions. Wait, the Fallacy of Extension is when you misrepresent someone's argument by exaggerating it. But in this case, the speaker is exaggerating their
	Instance 3 (Incorrect Prediction)
Input	Sentence: There must be objective rights and wrongs in the universe. If not, how can you possibly say that torturing babies for fun could ever be right?
Ground truth	`appeal to emotion`
`Llama-3.3-70b`:	The given sentence can be classified as an example of a false dilemma. The argument presents only two options: either there are objective rights and wrongs in the universe, or torturing babies for fun could be considered right. this is a false dichotomy because it ignores the possibility of other moral frameworks, such as moral relativism or subjective morality, where the morality of an action is determined by cultural, personal, or situational factors, rather than by objective universal standards. the sentence implies that if one does not accept the existence of objective rights and wrongs, they must condone or consider acceptable the torture of babies for fun, which is not a necessary or logical consequence. this oversimplification of complex moral issues into only two extremes is characteristic of a false dilemma.
	Instance 4 (Incorrect Prediction)
Input	Sentence: Thousands of pets are left homeless and hungry because lousy people turn them out on the streets instead of taking care of them.
Ground truth	`appeal to emotion`
`Mistral-22b`:	Output: Fallacy of Relevance: The sentence introduces a conclusion about pets being homeless and hungry, but it does not address the actual reasons behind this issue. It simply blames "lousy people" without providing evidence or a logical connection between the two. Additionally, it does not consider other possible factors that could contribute to the problem.

Table 7: Examples of correct and incorrect predictions for `Qwen3-32b`, `Mistral-22b`, and `Llama-3.3-70b` with Chain-of-thought in the task Fallacy Detection (Jin et al., 2022).

to audience sympathy. Instance 4 shows a similar example where the model focuses on the logical connection between the conclusion and the reason and ignores the emotional content of the whole argument.

Finally, we observed frequent instruction following failures in `Llama-3.3-70b` and `Mistral-22b`. `Llama-3.3-70b` ignores the formatting instructions in some cases, exclusively in chain-of-thought prompting. An example of such instances is Instance 3. In this example, `Llama-3.3-70b` does not use the prefix "Output:" and does not generate the fallacy at the end of the output. Other failures include using labels that are not listed in the task instructions or inventing a new label format (e.g., "**Output: ...**"). Appendix A.3 provides more details on output extraction.

A.2 Task Selection and Formulation

In the following, we outline all tasks in the ArgBench benchmark for each of the five skills that we derived from the datasets listed in Table 8. While not critical to the paper itself, we detail the selection and formulation of the tasks to provide insights into the benchmark creation process. Table 9 contains the size of the test sets in ArgBench for each skill. Further below, Tables 10 to 13 contain the task definitions as well as the input template for an exemplary set of tasks. The full list of task prompts can be found in the git repository.

Argument Mining Argument mining is the process of extracting arguments from natural language text. Traditional mining pipelines start with *unit segmentation*, where the task is to extract argument units (also called argument *components*) from an input document, that is, text spans that have an argumentative function. In our benchmark, we include four widely-used datasets for the task (Habernal and Gurevych, 2015; Al-Khatib et al., 2016; Stab and Gurevych, 2017; Poudyal et al., 2020) and define the instruction for them as follows:

```
Split the documents into spans
that are argumentative and those
that are not-argumentative,
separated with a line break.
```

After segmenting a document into units, *unit classification* is the task to assign one of multiple labels to each unit that describes the role of a unit in an argument (e.g., `major claim`, `claim`, or `premise`) or the type of evidence (e.g., `anecdote`, `expert opinion`, or similar).[6] To capture task diversity, we include three known datasets with texts of different genres (Stab and Gurevych, 2017; Al-Khatib et al., 2017; Park and Cardie, 2018) and different label schemes (see also Table 1 in the main paper). We prepend the argument unit with a context window to classify the argument unit. The context window spans five or more sentences that contain the input argument unit. We chose the maximum number of sentences as a context window size, while maintaining that the input's length for an instance does not exceed 1024 tokens. Simpler variants of argument unit classification split the task into several binary classification tasks for the target unit types. ArgBench therefore also contains three of these tasks: *conclusion extraction* in case law (Poudyal et al., 2020), *premise extraction* in case law (Poudyal et al., 2020), and premise extraction in Wikipedia (Ein-Dor et al., 2020). We add context of at least five sentences that contain the input sentence to these as well.

The third mining step is *relation detection*, which means to decide whether there is an argumentative relation between two argument units that makes them part of the same argument. In isolation, we include in case law as defined by Poudyal et al. (2020), where the input is a sentence pair and the output is either `related` or `not-related`. Negative sentence pairs (`not-related`) are all possible sentence pairs within a window of five sentences. We append the input sentence pair with a context of five instances that contain the candidate sentence pairs as input for this task.

After detecting the relation between argument units, the next step is *relation identification* (Peldszus, 2014), where the task is to classify the relation between two argument units into types that are defined in the argument model. Aiming for diversity again, ArgBench includes four classical relation identification datasets (Peldszus, 2014; Stab and Gurevych, 2017; Skeppstedt et al., 2018; Park and Cardie, 2018). The input is a pair of argument units accompanied by a context window, and the output is (`reason` or `evidence`) in E-rulemaking (Park and Cardie, 2018) and (`support` or `attack`) in the other three datasets. As a context window, we take all sentences between the candidate sentence pair. A similar task is to identify the relation between two arguments instead of argument units. In

[6] In our benchmark, we assume each argument unit to span one sentence to allow an easy and consistent evaluation.

Skill	Task	Dataset	Size	Genre	Source
Argument Mining	Conclusion Extraction	ECHR	42	Case Law	Poudyal et al. (2020)
	Premise Extraction	ECHR	42	Case Law	Poudyal et al. (2020)
	Premise Extraction	IBM-Evidence-Sentence	29,429	Encyclopedia	Ein-Dor et al. (2020)
	Relation Detection	ECHR	42	Case Law	Poudyal et al. (2020)
	Relation Identification	Microtexts-1	112	Student Essays	Peldszus (2014)
	Relation Identification	Essays	402	Student Essays	Stab and Gurevych (2017)
	Relation Identification	Microtexts-2	171	Student Essays	Skeppstedt et al. (2018)
	Relation Identification	Political-debates	1,462	Spoken Debates	Menini et al. (2018)
	Relation Identification	E-rulemaking	731	Web Forum	Park and Cardie (2018)
	Unit Classification	Essays	402	Student Essays	Stab and Gurevych (2017)
	Unit Classification	Editorials	731	News	Al-Khatib et al. (2017)
	Unit Classification	E-rulemaking	731	Web Forum	Park and Cardie (2018)
	Unit Segmentation	WebDiscourse	340	Web Forum	Habernal and Gurevych (2015)
	Unit Segmentation	Editorials	300	News	Al-Khatib et al. (2016)
	Unit Segmentation	Essays	402	Student Essays	Stab and Gurevych (2017)
	Unit Segmentation	ECHR	42	Case Law	Poudyal et al. (2020)
Argument Perspective Assessment	Argument Similarity	UKP-Aspect	3,595	Mixed	Reimers et al. (2019)
	Aspect Detection	UKP-Corpus	5,032	Mixed	Schiller et al. (2021)
	Frame Identification	Webis-argument-framing	12,326	Online Debates	Ajjour et al. (2019)
	Key Point Matching	ArgKP	24,093	Encyclopedia	Bar-Haim et al. (2020)
	Stance Classification	IBMSC	2,394	Encyclopedia	Bar-Haim et al. (2017)
	Stance Classification	Ukp-sentential	25,492	Mixed	Stab et al. (2018)
Argument Quality Assessment	Argument Ranking	IBM-Evicence-Quality	5,697	Encyclopedia	Gleize et al. (2019)
	Argument Ranking	IBM-RANK-30k	30,497	Encyclopedia	Gretz et al. (2020)
	Argument Ranking	Claim Revisions	587,881	Online Debates	Skitalinskaya et al. (2021)
	Argument Rating	Dagstuhl-15512	320	Online Debates	Wachsmuth et al. (2017)
	Argument Rating	GAQ	5,285	Web Forum	Ng et al. (2020)
	Controversy Scoring	ChangeMyView	7,242	Web Forum	Habernal et al. (2018b)
	Reasonableness Scoring	ChangeMyView	7,242	Web Forum	Habernal et al. (2018b)
	Suboptimal Claim Detection	Claim Revisions	198,089	Online Debates	Skitalinskaya and Wachsmuth
	Claim Improve. Suggestion	Claim Revisions	198,089	Online Debates	Skitalinskaya and Wachsmuth
Argument Reasoning	Ad-hominem Detection	Ad-hominem-CMV	1800	Web Forum	Habernal et al. (2018b)
	Fallacy Detection	Logic	3,800	Mixed	Jin et al. (2022)
	Fallacy Detection	ElecDeb60to20	1,640	Spoken Debates	Goffredo et al. (2023)
	Scheme Classification	ArgU	69,427	Online Debates	Saha and Srihari (2023b)
	Warrant Identification	SemEval-2018-task12	1,970	News	Habernal et al. (2018a)
Argument Generation	Argument Generation	ArgU	2,990	Online Debates	Saha and Srihari (2023b)
	Argument Summarization	OpenDebateEvidence	4,957,726	Online Debates	Roush et al. (2024)
	Claim Generation	Belief Arguments	51,470	Online Debates	Alshomary et al. (2021)
	Claim Optimization	Claim Revisions	198,089	Online Debates	Skitalinskaya et al. (2023)
	Counterarg. Generation	ChangeMyView	268,881	Web Forum	Hua and Wang (2018)
	Counterarg. Generation	Candela	287,152	Web Forum	Hua et al. (2019)
	Warrant Generation	ART	72,846	Narratives	Bhagavatula et al. (2020)
	Warrant Generation	ARC	1,654	News	Habernal et al. (2018a)
	Warrant Generation	Ideological Debates	494	Online Debates	Boltužić and Šnajder (2016)
	Warrant Generation	Micortext-1	112	Student Essays	Becker et al. (2020)

Table 8: The list of all computational argumentation datasets in ArgBench together with the numbers of input units.

ArgBench, we include the political debates dataset for this task (Menini et al., 2018), which takes as input an argument pair and classifies their relation into `support`, `attack`, or `not-related`.

Argument Perspective Assessment This skill covers the identification of all concepts describing the perspective of an argument on a given topic. As this comes in many task shades, we aim to reflect this variety in Benchmark, rather than having many datasets for each task.

One main perspective assessment task is *stance classification*, which means to determine the position of an argumentative text toward a topic (e.g., `pro`, `con`, or `neutral`). As two known but very different examples, we include essays (Stab and Gurevych, 2017) and the Wikipedia sentences from IBMSC (Bar-Haim et al., 2017). The input for stance classification tasks in ArgBench is an argument and a topic, and the output is the stance.

In *frame identification*, the goal is to extract the aspect of a topic that a given argument discusses, which we represent with one widely used dataset (Ajjour et al., 2019). For a unified frame identification setup, we provide an argument and a frame and ask the model to output `match` or `no-match`. We sample five wrong frames on the same topic for each argument in addition to the right frame associated with each argument. In case

Skill	Instances
Argument Mining	10,039
Argument Perspective Assessment	5,758
Argument Quality Assessment	7,720
Argument Reasoning	2,953
Argument Generation	7,325

Table 9: Count of instances per skill in the prompting experiment.

the count of frames for a topic is less than five, we sample frames from other topics. The rationale behind choosing five negative frames is that on average a topic is framed using six frames in the Webis-argument-framing dataset. The related task of *aspect detection*, represented by the data of Schiller et al. (2021), is defined by asking the model to segment an input document into spans that correspond to a specific aspect or not. Similar to unit segmentation, this task is evaluated by measuring precision, recall, and F_1-score of exact matches of the spans labeled with aspect with the ground truth spans labeled with aspect.

A more specific task is to assess *argument similarity*. In case of the data of Reimers et al. (2019), the input is a pair of arguments and the output one of four labels: `high similarity`, `some similarity`, `no similarity`, and `different topics`. Finally, *key point matching* means to decide whether an argument aligns with a key point or not and has been introduced in dataset from a successful shared task (Bar-Haim et al., 2020). It takes as input an argument and a key point and returns `match` or `no-match`.

Argument Quality Assessment The quality of an argument is crucial to persuade an audience of the perspective that the argument holds. This task can be modeled as an *argument rating* task, that is, to assign a value from a predefined scale (e.g., `low`, `medium`, or `high`) for a given quality dimension (e.g., reasonableness). For two of the most known respective datasets (Wachsmuth et al., 2017; Ng et al., 2020), we provide as input the argument, the quality dimension (e.g., effectiveness), and the definition of the quality dimension as provided in the paper. In addition, we analogously define two dimension-specific tasks in ArgBench, which are studied on ChangeMyView (Habernal et al., 2018b) (*reasonableness scoring* and *controversy scoring*).

In contrast to absolute ratings, argument quality can be framed in a relative manner, where an LLM judges which argument in an argument pair is better. We also represent these *argument ranking* tasks with two datasets (Gleize et al., 2019; Skitalinskaya et al., 2021), providing the two pieces of evidence or arguments as input. We ask the model to output whether the first argument is `better` or `worse` than the second argument. Assessing the quality of arguments is also modeled as a scoring task, where the input is a list of arguments and the output is a list of scores that reflects their relative quality (Gretz et al., 2020). Defining the task as scoring a long list of arguments violates the design constraint of limiting the input instance to 1024 tokens. To ensure a computationally feasible setup within ArgBench, we integrate this task in ArgBench by modeling it as a pair-wise argument ranking task. For each argument in the dataset, we sample an argument on the same topic and consider it `better` if it has a higher score, and `worse` otherwise.

With their data, Skitalinskaya and Wachsmuth (2023) have proposed two specific tasks that we further include. One is *suboptimal claim detection* (i.e., detecting weak claims in need of revision), which is modeled as a binary classification task. The input for this task is the claim, and the output is either `improvable` or `not-improvable`. The other is *claim improvement suggestion*, which is to make one of multiple revision suggestions for a claim. We define the task as a binary classification task, where the input is the claim and one of the four revision suggestions, and the output is `match` or `no-match`. The four revision suggestions that are included in this work are: clarification, typo/grammar, links, and other.

Argument Reasoning For understanding an argument's reasoning, reasoning patterns are often modeled via argumentation *scheme classification*; we include one dataset here (Saha and Srihari, 2023a) with six scheme classes (e.g., `argument from consequences`). Habernal et al. (2018a) introduced the *warrant identification* task for an argument (i.e., why the conclusion follows from the premise), given a valid and an invalid warrant as input. Another line of research is concerned with various types of *fallacy detection*. Aiming to well-represent the spectrum, we include both the data of Habernal et al. (2018b), where the goal is to detect ad-hominem arguments in the ChangeMyView forum (i.e., attacks on authors of arguments), the one of Jin et al. (2022) who classify fallacies into 13 types (e.g., `false cause`). We

Task	Task Definition	Authors
Conclusion Extraction	Given the following document, Judge if the following sentence is a conclusion or not. A conclusion is a controversial statement and the central component of an argument. Answer only with conclusion and No-conclusion. `Instance Format` Sentence: Document:	(Poudyal et al., 2020)
Premise Extraction	Given the following document, Judge if the following sentence is a premise or not. A Premise is a reason for justifying or refuting a claim. Answer only with Premise or No-premise. `Instance Format` Sentence: Document:	(Poudyal et al., 2020)
Premise Extraction	Judge if evidence can be used to support or attack the motion. Possible outputs: Accept if evidence can be an argument to support or attack the motion or Reject if the evidence can not be used to attack or support the motion. Only output Accept or Reject. `Instance Format` Motion: Evidence:	(Ein-Dor et al., 2020)
Relation Detection	Given the following document and two sentences, your task is to judge whether they are part of the same argument. An argument consists of a conclusion and multiple premises. Your task is to judge whether the two sentences are part of the same argument, where one sentence supports or attacks the other. Output related if there is an argumentative relation between the two sentences or not-related if not. Only output related or not-related. `Instance Format` Sentence 1: Sentence 2: Document:	(Poudyal et al., 2020)
Relation Identification	Given the following essay and the appended source and target argument units that appear in the essay. Output Support if the source argument unit supports the target argument unit, or output Attack if the source attacks the target. Only output Support or Attack. `Instance Format` Source: Target: Document:	(Peldszus and Stede, 2015)
Unit Classification	Given the following document and span, classify the span that appears in the document into the following argument unit types: Common Ground, Assumption, Testimony, Statistics, Anecdote, or Other. Common Ground: is common knowledge, a self-evident fact, an accepted truth, or similar. [Definitions for unit types are omitted for sake of space. Look up the github repo for more details.] `Instance Format` Span: Document:	(Al-Khatib et al., 2016)

Table 10: Prompt definition and instance format for each task in our benchmark. Very similar tasks are dropped and only the task with the oldest paper is kept. Remaining tasks are in the next three tables.

Task	Task Definition	Authors
Unit Segmentation	Given the following document, split all of the document into argumentative units and non-argumentative units. An argumentative unit is a statement that has an argumentative function for example a claim or anecdote. An argumentative unit may span a clause, a complete sentence, multiple sentences, or something in between. Prepend each argumentative unit with argumentative: and spans that are not Argumentative with Non-argumentative:. Output the extracted spans as they are ordered in the given document and separate them by a new line. Do not add a new formatting or enumeration also do not rephrase the argument units. Order the output spans as they appear in the document. `Instance Format` Document:	(Habernal and Gurevych, 2015)
Argument Similarity	Judge if the argument pairs are similar (on the same topic and cover similar aspects) or dissimilar (e.g., cover different topics or different aspects). Possible outputs: High similarity if two arguments are very similar, Some Similarity if two arguments are somewhat similar, No Similarity if two arguments are not similar, or Different Topics if two arguments belong to different topics. Only output High Similarity, Some Similarity, No Similarity, Different Topics. `Instance Format` Argument 1: Argument 2:	(Reimers et al., 2019)
Aspect Detection	Given the following argument, split the argument into spans of text that cover an aspect or not. An aspect is a span of the argument that characterizes the argument. Multiple aspects can be found in an argument. Prepend the aspect span with Aspect and the not-aspect span with Not-aspect. Do not rephrase the spans or modify it. Always process the whole argument. Multiple aspects can be found in an argument. In case there is no aspect, simply output the argument with Not-aspect before it. `Instance Format` Argument:	(Schiller et al., 2021)
Frame Identification	Judge if the given frame captures the most salient aspect of the given argument on the given topic. The frame is the main highlighted aspect of the topic which resonate with as specific audience. Possible responses: Match if the argument emphasizes the given frame and No-match if the argument is not emphasized by the frame. Only output Match or No-match. `Instance Format` Topic: Argument: Frame:	(Ajjour et al., 2019)
Key Point Matching	Judge if the following key point summarizes the given argument. A key point is a short talking point. Key points may be viewed as high-level arguments. They should be general enough to match a significant portion of the arguments, yet informative enough to make a useful summary. Possible responses: Match if argument is summarized by key point and No-match if argument is not summarized by key point. Only output Match or No-match. `Instance Format` Argument: Key Point:	(Bar-Haim et al., 2020)
Stance Classification	Classify the stance of the following claim into Pro or Con. Answer with Pro if the following claim supports the following topic. Answer with Con if the claim attacks the topic. Only answer with Pro or Con. `Instance Format` Topic: Claim:	(Bar-Haim et al., 2017)

Table 11: Continuation of Table 10. Details are described there.

Task	Task Definition	Authors
Argument ranking	Given the following argument pairs, is the first argument better or worse than the second argument in terms of quality. Only respond with Better or Worse. `Instance Format` Argument 1: Argument 2:	(Gretz et al., 2020)
Argument rating	Judge the quality of the argument according to quality aspect. Possible outputs: Low if arguments aspect quality is low, Average if argument's aspect quality is average, High if arguments aspect quality is high. Only output Low, Average, or High. `Instance Format` Argument: Quality Aspect: Quality Aspect Definition:	(Wachsmuth et al., 2017)
Controversy Scoring	Classify the following post according to its controversy into either Not Really Controversial, Somehow Controversial, or Very Controversial. `Instance Format` Post:	(Habernal et al., 2018b)
Reasonableness Scoring	Classify the following post according to its reasonableness into : Quite Stupid, Neutral, or Quite Reasonable. `Instance Format` Post:	(Habernal et al., 2018b)
Suboptimal Claim Detection	Judge if the following claim can be improved by revising it. Possible outputs: Improvable if revision should be made, Non-Improvable if no revision is necessary. Only output Improvable or Non-Improvable. `Instance Format` Claim:	(Skitalinskaya and Wachsmuth, 2023)
Claim Improvement Suggestion	Given an argumentative claim, does the following quality issue match the following claim. Available quality issues are Clarification, Typo/Grammar, Links, or Other. If the quality issue matches the claim, output Match. If the quality issue does not apply to the claim, output No-match. Only output Match or No-match. `Instance Format` Claim: Quality Issue:	(Skitalinskaya and Wachsmuth, 2023)
Fallacy Detection	Given the following argument, split the argument into spans that contains one of the following fallacies. A fallacy is a failure in reasoning which renders an argument invalid. In case a span does not contain a fallacy, simply prepend it with No-fallacy. The split spans should be separated by newlines and be output in the exact order they appear in the argument. Add before each span that covers a fallacy the name of the fallacy and a colon. Do not rephrase anything in the argument. Here are the candidate fallacies: [Definitions for fallacies are omitted for sake of space. Look up the repo for details] `Instance Format` Argument:	(Goffredo et al., 2023)
Fallacy Detection	A fallacy is a failure in reasoning which renders an argument invalid. Classify the following sentence into one of the following fallacies. [Definitions for fallacies are omitted for sake of space. Look up the repo for details] `Instance Format` Sentence:	(Jin et al., 2022)

Table 12: Continuation of Table 10. Details are described there.

Task	Task Definition	Authors
Scheme Classification	Classify the following argument according to the following Walton's argument schemes. [Definitions for schemes are omitted for sake of space. Look up the repo for details] Only output one of the following argument schemes: means for goal, goal from means, from consequence, source knowledge, source authority, rule or principle, and other. `Instance Format` Argument:	(Saha and Srihari, 2023b)
Warrant Identification	Given the following reason and claim along with the debate title and a short description of the debate they occur in, identify the correct warrant from two candidates. Warrant 1 and Warrant 2. The warrant explains why the claim follows from the reason. Only output Warrant 1 or Warrant 2. `Instance Format` Reason: Claim:	(Habernal et al., 2018b)
Argument Generation	Given the following argument type, topic, stance, and facts, generate an argument that holds that stance on the topic and is based on the facts. Facts are real-world concepts, propositions, and knowledge and do not refer to only knowledge-based facts. `Instance Format` Argument Type: Topic: Stance:	(Saha and Srihari, 2023b)
Argument Summarization	Provide an abstractive summary/card-tag of the argument made in the document below. `Instance Format` Document:	(Roush et al., 2024)
Claim Optimization	Given the following input argumentative claim with context information on the debate, rewrite the claim such that the output claim improves upon input claim in terms of text quality and argument quality, and preserves the meaning of the claim as far as possible. `Instance Format` Thesis: Claim:	(Skitalinskaya et al., 2023)
Conclusion Generation	Given a discussion topic and a collection of topic stances that describe users stance on various issues, generate a claim that is based on the user stance. A claim is a controversial statement and the central component of an argument. `Instance Format` Topic: User Stances:	(Alshomary et al., 2021)
Counter Argument Generation	Write a counterargument to the following original post and take into account retrieved passages related to the post. `Instance Format` Post: Retrieved Passages:	(Hua et al., 2019)
Counter Argument Generation	Given a statement and relevant evidence, generate a counterargument that attacks to the original argument and highlights the given key phrases. `Instance Format` Statement: Key Phrases: Evidence:	(Hua and Wang, 2018)
Warrant Generation	Given a premise and a claim, generate an enthymem. An enthymem is a reason with which the claim follows logically from the premise. `Instance Format` Premise: Claim:	(Chakrabarty et al., 2021)

Table 13: Continuation of Table 10. Details are described there.

define all these tasks in the benchmark as classification tasks: Given an argument, output its reasoning pattern or the valid warrant as a label.

In contrast, Goffredo et al. (2023) approaches the task by asking the model to segment the input text into six fallacy types. Accordingly, we ask the model to split the text into spans of text that correspond to one of the fallacies and to append the span with the fallacy label. In case a span does not cover a fallacy, the model is required to output `no fallacy`. For evaluation, we measure exact matches with the ground-truth spans for the six fallacy types and return their macro average.

Argument Generation The last skill is to craft an argument or argument unit for some input (e.g., for a statement). Again, we seek to capture the variety of tasks in our ArgBench benchmark.

For *argument generation*, Saha and Srihari (2023b) first generate a template that is then filled with facts. Hua and Wang (2018) and Hua et al. (2019) tackle the *counterargument argumentation* for a given topic. Researchers also approach generating arguments by generating one of its components (either a conclusion or its warrants). Among these, Alshomary et al. (2021) proposed belief-based *claim generation* to obtain claims that have a pro or con position on a big issue. With four different datasets, Chakrabarty et al. (2021) studied the *warrant generation task*, producing an argument's warrant, given its premise and conclusion. Skitalinskaya et al. (2023) introduced *claim optimization* as the task of revising claims to improve their quality, whereas Roush et al. (2024) aimed for *argument summarization*. Since generation matches the general idea of LLMs, we simply model all tasks as completion tasks, given the respective input.

A.3 Implementation Details

Output Extraction To extract the output of the LLMs, we used regular expressions to capture the output of the model and to remove the chain-of-thought reasoning steps. As a first cleaning step, we removed all reasoning clues that are surrounded by "<think>" and "<think/>". For classification tasks, we first inspect the first or the last tokens of the completed text. If none of the labels are found at the beginning of the generated text, we inspect the text after one of the following indicators: Output:" or "<think/>". In case we could not extract the label, we sample a label from the candidate labels randomly and consider it to be the predicted label.

Hyperparameter	Low	High	Step
Learning rate	10^{-6}	10^{-4}	–
Early stopping threshold	10^{-3}	10^{-4}	10^{-4}

Table 14: The value range for the hyperparameters in the in-task and leave-one-task-out experiments. For learning rate, we did not choose any pre-defined step and sampled from all real numbers between the low and high values.

Hyperparameter	Value
Epochs	30
Early stopping patience	3
Mini Batch size	4
Accumulation steps	4
Early stopping metric	Loss
Cutoff len	1024
Lora rank	16
Lora dropout	0.05
Lora alpha	16
Seed	1517

Table 15: Hyperparameters that are shared for the large language models in the in-task and leave-one-task-out experiments.

For segmentation tasks, we extract all spans of texts that are prepended with one of the span labels (e.g., `argumentative` and `Non-argumentative` for unit segmentation). For generation tasks, we take all the generated text as the output of the model after removing the reasoning steps. In case the generated text includes "Output:", we consider the output to be the text after this token.

Hardware We ran our experiments on an instance that is equipped with an NVIDIA A100 with 80 GB GPU Memory. The instance has eight 8-core CPU, each of which has access to 32 GB RAM.

Models We used HuggingFace to load the large language models in *Bfloat16 precision*. To optimize the LoRa adapters for these models, we used *adamw_bnb_8bit,* which uses 8-bit float precision to represent the states of the optimizer (e.g., Momentum).

Hyperparameter Optimization To optimize the models, we used the *Tree-structured Parzen Estimator* (Watanabe, 2023) implementation in Optuna[7] with 10 trials to fine-tune the models in all experiments. We consistently used the same range of values, which are shown in Table 14. Table 18 shows the parameters with the best performance

[7]https://optuna.org/

on the validation sets of the in-task and leave-one-task-out experiments. The values of other parameters that are used in the experiments are shown in Table 15. For fine-tuning on the target tasks, we always used 20 epochs, except for unit segmentation (Ajjour et al., 2017) where we used 30 epochs. For the leave-one-task-out experiment, we chose one of the validation tasks, stance classification (Stab et al., 2018) to optimize the hyperparameters of the models.

Benchmark Distribution The following tasks from the released benchmark are available for download upon agreeing to the license. To do so, please contact the authors.

- Suboptimal Claim Detection (Skitalinskaya and Wachsmuth, 2023)
- Claim Optimization (Skitalinskaya et al., 2023)
- Argument Ranking (Skitalinskaya et al., 2021)
- Claim Improvement Suggestions (Skitalinskaya and Wachsmuth, 2023)
- Argument Summarization (Roush et al., 2024)

A.4 Experiments

A.4.1 Manual evaluation

The manual evaluation started with a pilot study where two authors labeled 50 counterarguments for five randomly sampled arguments from counterargument generation (Hua and Wang, 2018). The goal of the pilot study was to develop the annotation guideline and validate the counterargument collection process. For the main study, which we report on in the main body, we recruited three annotators from three different countries on Upwork: the USA, Canada, and India. We have paid each annotator 15 dollars per hour and requested from them a confirmation that they are fine with labeling sensitive content about social issues like racism. All in all, the crowdworkers spend 14 hours on the annotation task. A screenshot of the interface and the instructions are provided in Figure 1. Detailed definitions of the quality criteria on which the annotators assessed the counterarguments are listed here:

- **Argumentativeness**: whether the counterargument constitutes an argument.
- **Countering**: whether the counterargument takes an opposite stance to the argument.
- **Relevance**: whether the counterargument's content covers the content of the evidence and key phrases.

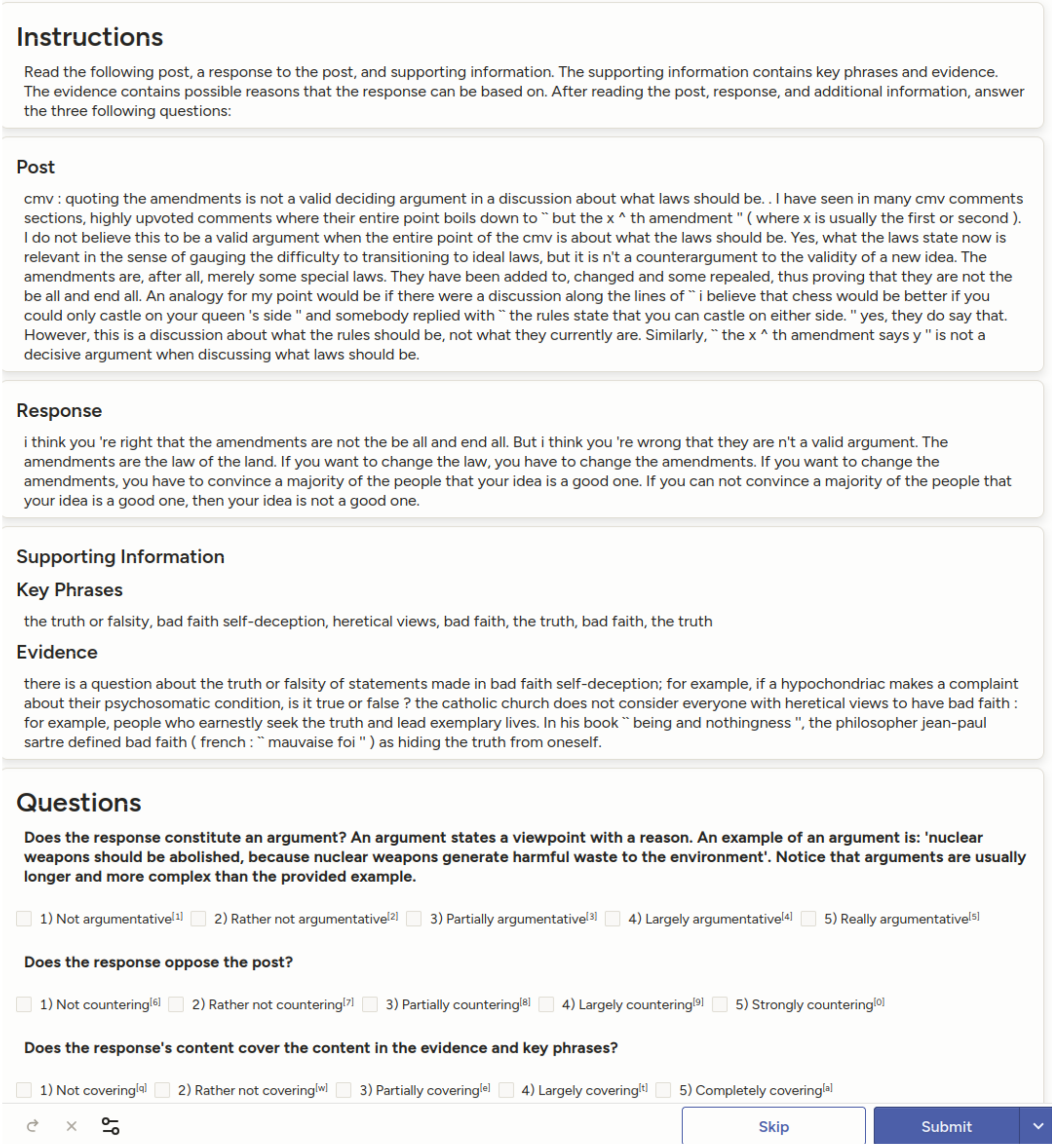

Figure 1: The interface for the manual evaluation of counterarguments that are generated by `Mistral-22b`, `Mistral-7b`, `Qwen3-32b`, and `Phi-3.5-MoE-7.6b` on the counterargument generation task (Hua and Wang, 2018).

Model	Link
DeepSeek-R1-7b	http://hf.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
DeepSeek-R1-32b	https://hf.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B
Qwen3-4b	https://hf.co/Qwen/Qwen3-4B
Qwen3-32b	https://hf.co/Qwen/Qwen3-32B
Mistral-7b	https://hf.co/mistralai/Mistral-7B-Instruct-v0.3
Mistral-22b	https://hf.co/mistralai/Mistral-Small-Instruct-2409
Mixtral-8x7b	https://hf.co/mistralai/Mixtral-8x7B-Instruct-v0.1
Llama-3.1-8b	https://hf.co/meta-llama/Llama-3.1-8B-Instruct
Llama-3.3-70b	https://hf.co/meta-llama/Llama-3.3-70B-Instruct
Phi-3.5-MoE-7.6b	https://hf.co/microsoft/Phi-mini-MoE-instruct

Table 16: The urls for the hugging face models used in the paper

Task	Prompting Technique	Top-k	Min Tokens	Max Tokens
Aspect Detection	Few-shot-prompting	50	32	1024
	Chain-of-thought	50	32	2048
Argument Unit Segmentation	Few-shot-prompting	50	32	2048
	Chain-of-thought	50	32	2048
Fallacy Detection	Few-shot-prompting	50	32	1024
	Chain-of-thought	50	32	2048
Generation Tasks	Few-shot-prompting	50	32	512
	Chain-of-thought	50	32	1024
Classification Tasks	Few-shot-prompting	1	16	32
	Chain-of-thought	1	16	300

Table 17: The generation parameters used in all experiments for the segmentation tasks (argument unit segmentation (Habernal and Gurevych, 2015; Al-Khatib et al., 2016; Stab and Gurevych, 2017), aspect detection (Schiller et al., 2021), and fallacy detection (Goffredo et al., 2023), generation tasks, and classification tasks. Min tokens and max tokens refer to the count of tokens to be generated for each task. Classification tasks include all tasks in the benchmark whose output is a class without completion. Generation tasks cover argument generation tasks. The parameters for few-shot prompting is also used for zero-shot prompting.

	DeepSeek-R1-7b		DeepSeek-R1-32b		Llama-3.1-8b		Mistral-7b	
Task	**LR**	**EST**	**LR**	**EST**	**LR**	**EST**	**LR**	**EST**
Training on the Target Task								
Unit Segmentation	$4.98 \cdot 10^{-5}$	10^{-3}	$8.25 \cdot 10^{-5}$	10^{-4}	$8.06 \cdot 10^{-5}$	10^{-4}	$3.49 \cdot 10^{-5}$	10^{-4}
Argument Similarity	$4.19 \cdot 10^{-5}$	10^{-3}	$7.33 \cdot 10^{-5}$	$7 \cdot 10^{-3}$	$3.74 \cdot 10^{-5}$	10^{-4}	$9.83 \cdot 10^{-5}$	$7 \cdot 10^{-4}$
Argument Rating	$9.93 \cdot 10^{-5}$	10^{-4}	$2.53 \cdot 10^{-5}$	10^{-4}	$3.28 \cdot 10^{-5}$	$4 \cdot 10^{-4}$	$3.49 \cdot 10^{-5}$	$4 \cdot 10^{-4}$
Ad-hominem Detection	$3.22 \cdot 10^{-5}$	10^{-3}	$7.71 \cdot 10^{-5}$	10^{-4}	$3.13 \cdot 10^{-5}$	$8 \cdot 10^{-4}$	$1.31 \cdot 10^{-5}$	$7 \cdot 10^{-4}$
Counterargument Generation	$8.96 \cdot 10^{-5}$	10^{-4}	$7.49 \cdot 10^{-5}$	10^{-4}	$9.98 \cdot 10^{-5}$	10^{-4}	$9.88 \cdot 10^{-5}$	10^{-4}
Training on all other Tasks								
Five Target tasks	$9.51 \cdot 10^{-5}$	10^{-4}	$8.65 \cdot 10^{-5}$	10^{-4}	$3.99 \cdot 10^{-5}$	10^{-4}	$2.87 \cdot 10^{-5}$	10^{-4}

	Mistral-22b		Qwen3-4b		Qwen3-32b		Phi-3.5-MoE-7.6b	
Task	**LR**	**EST**	**LR**	**EST**	**LR**	**EST**	**LR**	**EST**
Training on the Target Task								
Unit Segmentation	$8.06 \cdot 10^{-5}$	$5 \cdot 10^{-3}$	$8.6 \cdot 10^{-5}$	10^{-4}	$6.05 \cdot 10^{-5}$	$9 \cdot 10^{-3}$	$9.79 \cdot 10^{-5}$	10^{-3}
Argument Similarity	$5.36 \cdot 10^{-5}$	$5 \cdot 10^{-4}$	$6.08 \cdot 10^{-5}$	10^{-3}	$2.86 \cdot 10^{-5}$	$5 \cdot 10^{-4}$	$6.72 \cdot 10^{-5}$	10^{-3}
Argument Rating	$1.76 \cdot 10^{-5}$	$8 \cdot 10^{-4}$	$1.94 \cdot 10^{-5}$	10^{-3}	$7.09 \cdot 10^{-5}$	10^{-4}	$6.99 \cdot 10^{-5}$	$6 \cdot 10^{-4}$
Ad-hominem Detection	$6.48 \cdot 10^{-5}$	10^{-4}	$2.98 \cdot 10^{-5}$	10^{-3}	$6.40 \cdot 10^{-5}$	$7 \cdot 10^{-4}$	$7.57 \cdot 10^{-5}$	10^{-4}
Counterargument Generation	$6.53 \cdot 10^{-5}$	10^{-4}	$9.86 \cdot 10^{-5}$	$4 \cdot 10^{-4}$	$8.70 \cdot 10^{-5}$	$6 \cdot 10^{-4}$	$8.55 \cdot 10^{-5}$	$\cdot 10^{-4}$
Training on all other Tasks								
Five Target tasks	$7.06 \cdot 10^{-5}$	10^{-4}	$9.79 \cdot 10^{-5}$	10^{-4}	$3.19 \cdot 10^{-5}$	10^{-4}	$5.88 \cdot 10^{-5}$	10^{-4}

Table 18: The hyperparameters learning rate (LR) and early stopping threshold (EST) when *training on the target task* and *training on all other tasks*. The target tasks are: unit segmentation (Habernal and Gurevych, 2015), argument rating (Wachsmuth et al., 2017), argument similarity (Reimers et al., 2019), ad-hominem detection (Habernal et al., 2018b), and counterargument generation (Hua and Wang, 2018).