LLMs for Argument Mining: Detection, Extraction, and Relationship Classification of pre-defined Arguments in Online Comments

Matteo Guida[1] **Yulia Otmakhova**[1] **Eduard Hovy**[1] **Lea Frermann**[1]
[1]School of Computing and Information Systems,
The University of Melbourne
guida@student.unimelb.edu.au,
{y.otmakhova,eduard.hovy,lea.frermann}@unimelb.edu.au

Abstract

Content Warning: *This paper discusses examples of harmful language. The authors do not support such content. Reader caution is advised.*

Automated large-scale analysis of public discussions around contested issues like abortion requires detecting and understanding the use of arguments. While Large Language Models (LLMs) have shown promise in language processing tasks, their performance in mining topic-specific, pre-defined arguments in online comments remains underexplored. We evaluate four state-of-the-art LLMs on three argument mining tasks using datasets comprising over 2,000 opinion comments across six polarizing topics. Quantitative evaluation suggests an overall strong performance across the three tasks, especially for large and fine-tuned LLMs, albeit at a significant environmental cost. However, a detailed error analysis revealed systematic shortcomings on long and nuanced comments and emotionally charged language, raising concerns for downstream applications like content moderation or opinion analysis. Our results highlight both the promise and current limitations of LLMs for automated argument analysis in online comments.[1]

1 Introduction

Online discourse on social media or in discussion fora on complex controversial topics brings both challenges and opportunities for understanding the formation and spread of opinions, and their expression through arguments, at scale. Automatic analysis of public debate is crucial for tracking how opinions form and spread, identifying the evidence supporting different viewpoints, and evaluating the quality of public discourse (Stede and Schneider, 2018).

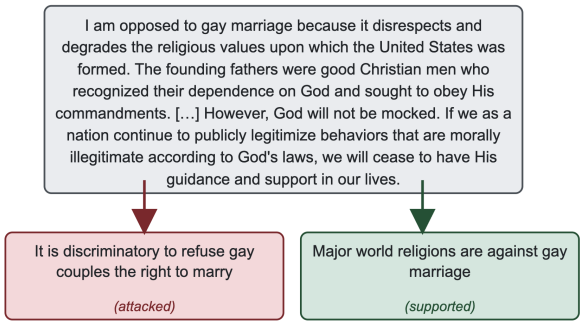

Figure 1: An online comment (top) which makes use of two pre-defined arguments (red and green boxes). The comment attacks A1 (left) and supports A2 (right).

Accordingly, a rich body of work on computational argument mining and understanding has emerged which includes the detection of argumentative discourse units in texts (Habernal and Gurevych, 2017; Hidey et al., 2017), the relationships of these units (in terms of attack and support, (Carstens and Toni, 2015; Ruiz-Dolz et al., 2021) and the identification of use cases of pre-defined arguments in heterogeneous texts (Boltužić and Šnajder, 2014; Hasan and Ng, 2014; Levy et al., 2014). This latter approach enables researchers to abstract away from individual expressions by aggregating them into pre-defined argument types, thereby facilitating the analysis of broader and recurring argumentation patterns that would be difficult to capture through 'bottom-up' argument mining.

Here, we build on this approach. We start with a controversial *topic* ("Legalisation of Abortion"), paired with pre-defined *arguments*[2] which can be either in favour of ("Abortion is a woman's right"), or against the topic ("Abortion kills a life"). Our goal is to identify usages of these arguments in online comments. A comment can make use of an argument by either *supporting* or *attacking* it

[1]Our code, data and prompts can be found at: https://github.com/mattguida/llm-for-arg-min

[2]In this paper, we use the term *argument* to refer to "a general, concise statement that directly supports or contests the given topic", following Levy et al., 2014, pg.1489.

Proceedings of The 23rd Annual Workshop of the Australasian Language Technology Association, pages 176–191
November 26-28, 2025

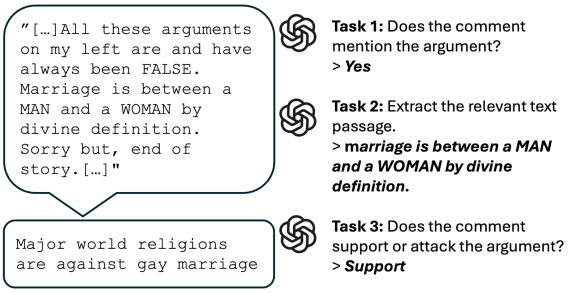

Figure 2: A comment (top, left) and pre-defined argument (bottom, left). We predict whether the comment makes use of the argument (Task 1), where it mentions the argument (Task 2) and whether it supports or attacks the argument (Task 3).

(Figure 1).

Correspondingly, we formulate three tasks to disentangle models' performance: identify whether an argument is used in a comment (Task 1); extract the span of text in which the argument is being used (Task 2); and assess whether the argument is supported or attacked in the comment (Task 3). This is illustrated in Figure 2. While these tasks are not new (cf., Section 2), taken together they provide a comprehensive picture of model performance.

On this basis, we make two empirical contributions. First, we inspect the ability of large-language models (LLMs) to identify pre-defined arguments in noisy online comments. With the increased performance and adoption of LLMs, and large-scale opinion analysis in social media being a conceivable use-case, to the best of our knowledge LLMs have not yet been systematically tested on these tasks using a set of topic-specific, pre-defined arguments. Second, given the sensitive nature of the task and consequential importance to avoid systematic bias in model performance, we conduct a detailed qualitative and quantitative error analysis of model outputs.

To assess LLMs on the proposed tasks, we utilize datasets of over 2,000 opinion comments spanning six polarizing topics (Boltužić and Šnajder, 2014; Hasan and Ng, 2014). For each topic, a set of pre-defined arguments has been identified, and comments were annotated for the presence and usage (support vs attack) of arguments. We experiment with four state-of-the-art LLMs comprising open and closed-source models of varying sizes. Our findings are two-fold: first, fine-tuned LLMs outperform both prompted LLMs and traditional fine-tuned models (RoBERTa) on argument detection and extraction tasks, although at a significant environmental cost. Second, our error analysis exposes systematic weaknesses: LLMs frequently over-predict arguments in comments using strong and emotional language, struggle to distinguish the implicit and explicit use of arguments, and perform worse on longer, more nuanced comments. These patterns suggest that while LLMs show promise for argument mining using pre-defined arguments, their current limitations could lead to biased analyses in applications like public opinion analysis or content moderation.

2 Related Work

Argument Mining A vast body of work has studied argumentation from theoretical and empirical perspectives, adopting an open-domain, bottom-up approach to identify argumentative units directly from unstructured text. Early research focused on automatically identifying arguments (or premises) and conclusions (or claims) in opinionated texts such as essays or online discussions (Habernal and Gurevych, 2017; Hidey et al., 2017; Feng and Hirst, 2011; Stab and Gurevych, 2017). Other work examined the interaction of these components, as premises supporting or attacking a claim (Cocarascu and Toni, 2017; Carstens and Toni, 2015; Ruiz-Dolz et al., 2021). These tasks are often addressed jointly through structured prediction models (Egawa et al., 2020; Stab and Gurevych, 2017).

Argument Mining with pre-defined Arguments Another line of research focuses on identifying pre-defined arguments – typically sourced from debate platforms – in unstructured text. In works on argument search, for instance, such arguments are retrieved and ranked from such platforms in response to user queries (Al et al., 2017; Stab et al., 2018). Beyond argument search and ranking, Levy et al. (2014) automatically detected claims from Wikipedia articles that were relevant to a set of pre-defined arguments.

Similarly, key point analysis (KPA) identifies lists of "key points" that summarize arguments about a variety of topics (Bar-Haim et al., 2020b,a) and is thus similar in flavour to our Task 1. These KPA datasets are based on crowd-sourced arguments with a strict length limitation (210 characters max as opposed to a *median* 480 characters in the data we use – see Table 7 and Table 8 for complete statistics and examples), and with crowd-sourced associated key points. While evaluation on KPA data sets is a worthwhile avenue for future work,

in this paper we focus (a) on datasets that support all three evaluation tasks, which allows for a comprehensive evaluation of LLMs, and (b) real-world online commentary, which is more representative of natural, varied, and "heated" discussions, thus potentially harder for the model to understand.

To do so, we build on Boltužić and Šnajder (2014) and Hasan and Ng (2014), who developed datasets which labelled opinion comments on divisive issues (like abortion) with the presence and usage of carefully crafted pre-defined issue-related arguments from online debate platforms (details in Section 3.1). The original works trained SVMs and Maximum Entropy models, respectively, on selected subsets of our proposed tasks.

Argument Mining with Large Language Models LLMs have caused substantial performance gains across argument mining tasks ranging from argument extraction (de Wynter and Yuan, 2024), understanding (Gorur et al., 2024; Otiefy and Alhamzeh, 2024), and quality assessment (van der Meer et al., 2022). However, for tasks like argument generation and persuasiveness (Hinton and Wagemans, 2023) and argumentative fallacy identification (Ruiz-Dolz and Lawrence, 2023) results were mixed. Similarly, cross-task review papers on argument mining have reported mixed results (Chen et al., 2024; Alsubhi et al., 2023; Ruiz-Dolz et al., 2024).

We complement this line of evaluation with the first comprehensive assessment of LLMs to detect and understand pre-defined arguments in opinion comments (but see Gorur et al. (2024) for a study specific to relation classification). We systematically assess fine-tuned, and few-shot LLMs on all three defined tasks and conduct detailed qualitative and quantitative error analyses.

3 Methodology

3.1 Data

Our study builds on prior research in natural language processing, particularly works that intersected curated arguments from online debate platforms with large-scale online discussions.

The **COMARG dataset**: Boltužić and Šnajder (2014) manually annotated 373 comments from the discussion platform *Procon.org* with a pre-defined list of arguments retrieved from *Idebate.org*. It encompasses two topics: the legalisation of gay marriage (GM) and the inclusion of the phrase "Under God" in the U.S. Pledge of Allegiance (UGIP). GM comments were labeled for the presence of three arguments in favor (Pro) and four arguments against the topic (Con), while the UGIP topic featured three Pro and three Con arguments. Each attested comment-argument pair was further classified based on whether the comment explicitly supported, implicitly supported, explicitly attacked or implicitly attacked the argument. Inter-annotator agreement was moderate, and the final labels were decided by majority vote, excluding all cases where no majority was reached.

The **YRU dataset**: Hasan and Ng (2014) sourced 1900 comments from *createdebate.com*, covering four topics: abortion (AB), gay rights (GR), legalization of marijuana (MA), and the Obama presidency (OB). For each topic, annotators identified a set of 6-9 arguments each supporting and opposing the topic. The data set was originally developed for the task of argument extraction, i.e., manually labeled with spans of text that employed a specific argument. Annotator agreement on this labelling task was reported as moderate to high, and disagreements were resolved through discussion. Table 6 in the Appendix lists all arguments for the six topics across both datasets.

3.2 Task Definitions

We assess our models on three argument mining tasks designed to test their abilities to *detect*, *extract*, and *understand the use of* arguments in online comments.

Task 1: Binary Argument Detection Given an argument A and a comment C, the task is to classify, in binary fashion, whether C makes use of A. We run this task on both YRU and COMARG, across all six topics.

Task 2: Argument Span Extraction Given an argument A and a comment C, the goal is to extract the span within C that expresses A. Only the YRU dataset comes with manually annotated argument spans, so we evaluate this task over the four YRU topics.

Task 3: Argument Relationship Classification Given an argument A and a comment C, we determine the relationship between A and C as C either attacking or supporting A. We consider two formulations of this task: either a binary classification as support or attack; or a 4-way classification distinguishing between explicit/implicit support for or an explicit/implicit attack of an argument. Only the COMARG dataset labels the type of usage of

an argument, so we evaluate relation classification over the two topics in this dataset.

3.3 Data Pre-Processing

For binary argument detection (Task 1) we pre-processed the original datasets to conform to support a binary classification task. For the COMARG dataset we consider all comment-argument pairs labeled as exhibiting any form of argumentative relationship as present (1). The data contained an explicit label of 'makes no use of an argument', which we reuse as our negative (not present) label (0). The YRU dataset is annotated for arguments on the sentence level. We project these labels to the comment-level, and consider them as present (1). All arguments not identified in any sentence were labeled as not present (0).

For the span extraction (Task 2), we only considered the labels present in the original YRU dataset and the manually annotated spans in the comment. Finally, for the argument relationship classification (Task 3), we treated the data in the COMARG dataset differently for the two subtasks. In subtask 3a we conflated the original labels in a binary fashion, only aiming at identifying whether the comment supports or attacks the argument. For subtask 3b we considered the original scale of implicit/explicit support and attack, we thus left the original 4-way labeling unaltered.

3.4 Models

We selected four Large Language Models (LLMs) from different model families, spanning one open-source – Llama3-8b-Instruct (Dubey et al., 2024) – and three proprietary models: GPT4o-mini and GPT-4o (Achiam et al., 2023), and Gemini1.5-Flash (Reid et al., 2024). We followed established practices to minimize non-deterministic behavior and output variability (Zhang et al., 2023; Meng et al., 2023), i.e. setting the temperature to 0 and the top_p parameter to 1 (Liu et al., 2023; Brown et al., 2023). [3]

Prompts In preliminary experiments, we varied our prompts along three key dimensions: structure (unstructured vs. structured step-by-step instructions), specificity (varying level of detail on task requirements and constraints), and role assignment (including/excluding the specific assignment of a role such as "you are an expert in argument analysis"). For argument detection (Task 1), a structured prompt with detailed instructions but without role assignment performed best. For both span extraction (Task 2) and argument relationship classification (Task 3), prompts that combined structured step-by-step instructions with explicit role assignment achieved superior performance. These optimized prompts were used for all subsequent experiments.[4]

Each task was attempted as zero-shot, 1-shot and 5-shot. To assess the impact of chosen examples, each few-shot experiment was run five times with randomly sampled, non-overlapping instruction examples. We manually verified that examples were instructive, and that the five-shot example set covered all classes.

RoBERTa Baselines We fine-tuned one RoBERTa model (Liu, 2019) for each task, by combining all the available data across topics. The relatively small number of samples for individual topics renders topic-wise fine-tuning infeasible.

For the classification tasks, we concatenated each comment-argument pair using the [SEP] token as a delimiter. We randomly split the data into five stratified folds for cross-validation, ensuring a balanced label distribution in each split. Each model was trained for 3 epochs with a batch size of 16. For the span extraction task, we formatted the data equivalent to extractive question-answer tasks, where arguments serve as "question", and relevant spans as the "answer" to be extracted. We fine-tuned a RoBERTa model on this data using the QuestionAnsweringModel from SimpleTransformers[5] again with five fold stratified cross validation, training for a total of 10 epochs and with a batch size of 16.[6]

LLM Fine-tuning To disentangle the effect of fine-tuning from model size, we also fine-tune one of our LLMs. For Llama3-8b-Instruct we performed parameter-efficient fine-tuning using low-rank adaptation (LoRA) (Hu et al., 2021), with cross-validation on five stratified folds. The details of hyperparameters and training protocol are provided in Appendix I. We include fine-tuned Llama only for the argument detection task and the argument extraction task, because the fine-

[3] For Llama3-8b-Instruct, we also set the top_k parameter to 1. GPT4o-mini and Gemini1.5Flash do not feature manual configuration of this parameter.

[4] The full prompts are released in our repository.

[5] `https://simpletransformers.ai/docs/qa-model/`

[6] Information about the parameters are reported inc Appendix H.

Model	GM	UG	AB	GR	MA	OB	Comb
Majority	0.40	0.41	0.47	0.47	0.46	0.48	0.44
RoBERTa							0.61
Zero shot							
Gemini1.5-f	0.79	0.73	0.73	0.67	0.66	0.67	0.72
GPT4o	0.76	**0.75**	**0.81**	0.72	**0.68**	0.66	0.68
GPT4o-m	0.75	0.74	0.76	0.67	0.66	0.67	0.69
Llama3	0.69	0.65	0.65	0.65	0.63	0.63	0.65
One shot							
Gemini1.5-f	**0.80**	**0.75**	0.74	0.68	0.67	0.67	0.72
GPT4o	0.75	0.73	0.79	**0.73**	0.65	**0.68**	0.73
GPT4o-m	0.78	0.63	0.75	0.67	0.67	0.67	0.70
Llama3	0.63	0.63	0.62	0.63	0.59	0.60	0.61
Five shot							
Gemini1.5-f	**0.80**	0.74	0.73	0.67	0.67	0.67	0.73
GPT4o	0.76	0.72	0.76	0.71	0.66	**0.68**	0.71
GPT4o-m	0.75	0.63	0.75	0.68	**0.68**	0.67	0.70
Llama3	0.60	0.62	0.61	0.63	0.59	0.59	0.60
Llama3 FT							**0.76**

Table 1: Results for binary argument detection (Task 1) for six topics and the combined data set (final column) as macro-averaged F1. We report a majority baseline (predicting the most frequent class), and fine-tuned RoBERTa and fine-tuned Llama3 (Llama3 FT) on the combined data only. The best F1 scores per data set are bolded. 1-shot and 5-shot results are averaged over five runs. The majority baseline is defined as predicting the most frequent class in the training data.

Model	AB	GR	MA	OB	Comb
RoBERTa					0.44
	Zero shot				
Gemini1.5-flash	0.42	0.41	0.37	0.38	0.40
GPT4o	0.31	0.32	0.30	0.32	0.31
GPT4o-m	0.28	0.29	0.27	0.25	0.27
Llama3	0.29	0.33	0.27	0.28	0.29
	One shot				
Gemini1.5-flash	0.46	0.46	0.43	0.47	0.46
GPT4o	0.36	0.41	0.37	0.41	0.39
GPT4o-m	0.35	0.38	0.37	0.36	0.37
Llama3	0.36	0.42	0.37	0.41	0.39
	Five shot				
Gemini1.5-flash	**0.50**	**0.51**	**0.48**	**0.55**	0.51
GPT4o	0.44	0.48	0.42	0.47	0.45
GPT4o-m	0.43	0.46	0.42	0.43	0.44
Llama3	0.48	0.50	0.43	0.50	0.48
Llama3 FT					**0.54**

Table 2: Results for Argument Extraction (Task 2) for the four topics in the YRU data set and the combined data set (final column) as Rouge-L. Models as in Table 1. The best Rouge-L scores per data set are bolded. 1-shot and 5-shot results are averaged over five runs with different examples.

tuned RoBERTa for the relationship classification task was widely outperformed by all LLMs in the prompting setup.

4 Results

We present quantitative results of our four LLMs and baselines across tasks, then detail error analysis. We find that (1) fine-tuned Llama outperformed all other models in detecting and extracting arguments; (2) larger LLMs generally outperformed smaller models and are more robust to different few-shot examples (exhibiting smaller variance); (3) that instruction examples (one- or five-shot) do not necessarily lead to enhanced performance; and (4) that the *detection* of arguments in comments (Task 1) is more challenging for LLMs than binary relationship classification (Task 3), which calls for caution with and future research on automated argument extraction in online discussion.

4.1 Task 1: Binary Argument Detection

We test four models (Llama, GPT4o, GPT4o-mini, Gemini) in 0-, 1-, and 5-shot settings across six different topics on predicting whether a given argument is stated in a comment or not. Results in Table 1 show that all LLMs outperform the baselines, and that the fine-tuned Llama3 performs best overall.[7] Among the prompt-based models, the largest variants (GPT4o and Gemini) outperform their smaller counterparts. We observe a strong variance across topics, with abortion (AB) and gay marriage (GM) performing best. Finally, and perhaps counterintuitively, we do not observe consistent improvement with more examples. The standard deviation (std) across five model runs for few-shot experiments was ±0.01 to ±0.02 for larger models, indicating high robustness to varying inputs, while smaller models showed slightly higher std, ±0.02 to ±0.03, especially in 1-shot settings.

4.2 Task 2: Argument Extraction

Here, we tasked models with identifying the exact span of text in which an argument is being used. We report the ROUGE-L scores (Lin, 2004) between predicted and golden spans.

Results in Table 2 reveal that, similar as in Task 1, the fine-tuned Llama3 outperformed all

[7] For task 1, the F1 SDs of the fine-tuned LLM range from ±0 to ±0.01, indicating robustness.

other models.[8] In prompting experiments, 5-shot Gemini consistently performs best. We observe a consistent improvement with exposure to more examples in the task instruction. We posit that this is due to the extractive nature of the task, which is more challenging for LLMs out-of-the-box compared to classification (Task 1). Most interestingly, we observe that most LLMs outperform the RoBERTa baseline only in the 5-shot setting on the combined data set, and the gap between non-fine tuned LLMs and RoBERTa is small (with the exception of 5-shot Gemini). Larger models (Gemini, GPT4o) show low std (±0.01 to ±0.03), while smaller models (GPT4o-mini, Llama) exhibit slightly higher std (±0.02 to ±0.05), especially in 5-shot settings.

While ROUGE-L evaluates strict lexical overlap, it disproportionately penalizes extracted spans that use different wordings to express the same point as in the golden spans. For example, for the argument "*Gay people should have the same rights as straight people*", a gold span "*Its not our job to tell people what they should do*" and a predicted span "*Personally, I think love is equal, whether is in the form of a man and a woman, a man with a man, or a woman with a woman*" are both expressions of the given argument, but achieve a ROUGE-L score of only 0.08 due to low lexical overlap, ignoring their semantic affinity. To assess this, we additionally computed BERTScore (Zhang et al., 2020), which computes token-level semantic similarity using BERT contextual embeddings, for the best-performing model (Gemini) averaged over all data sets. Across splits, BERTScores are consistently high (mean F1=0.87–0.91). While BERTScore is known to over-estimate extractive performance of models, and should not be used as the sole metric in a task like argument understanding where subtle differences in wording have large effects, a comparison of both metrics and manual inspection suggests that the ROUGE-L scores are a lower-bound of true model performance.

4.3 Task 3: Argument Relationship Classification

Given a comment and an argument featured in the comment, we ask models whether the argument is *supported* or *attacked* in the comment, either in a **binary** fashion, or on a 4-way **scale** (explicitly/implicitly supports; explicitly/implicitly attacks). Focusing on the binary task (Table 3, left) we observe that the two largest models (Gemini and GPT4o) consistently perform best, achieving almost perfect results. Exposure to examples does not improve performance and, in fact, substantially decreases results for GPT4-mini and Llama3. We observe a substantial performance decrease when moving to the 4-way classification task (Table 3, right), with the larger LLMs again performing best. The F1 std for the models show that Gemini1.5-f indicates low variability (std ±0.02), while GPT-4o-m and GPT-4o have substancial variability (std ±0.03 to ±0.16), and Llama3 shows even higher variability (std ±0.07 to ±0.10).

RoBERTa fails on this task, barely outperforming the Majority baseline, due to the small number of instance per label. This is supported by the fact that RoBERTa achieves better results on the binary classification than on the 4-way classification task, where class merging increases the number of examples per category.

Interestingly, performance across models was higher in the binary version of Task 3 than Task 1. In other words, models do better at identifying whether a comment *supports or attacks* a given argument than at detecting whether a comment *uses*

Model	Binary			Scale		
	GM	UG	Comb	GM	UG	Comb
Majority	0.39	0.37	0.38	0.14	0.37	0.25
RoBERTa			0.39			0.15
Zero shot						
Gemini1.5-f	0.92	**0.96**	0.94	0.55	0.59	0.57
GPT4o	0.94	**0.96**	**0.95**	0.56	**0.61**	0.58
GPT4o-m	0.77	0.91	0.84	0.40	0.40	0.40
Llama3	0.83	0.78	0.80	0.34	0.45	0.39
One shot						
Gemini1.5-f	**0.93**	0.90	0.91	**0.57**	**0.61**	**0.59**
GPT4o	0.71	0.86	0.78	0.40	0.40	0.40
GPT4o-m	0.65	0.81	0.73	0.37	0.38	0.37
Llama3	0.55	0.73	0.64	0.30	0.30	0.30
Five shot						
Gemini1.5-f	**0.93**	**0.96**	0.94	**0.57**	**0.61**	**0.59**
GPT4o	0.68	0.92	0.80	0.40	0.40	0.40
GPT4o-m	0.64	0.86	0.75	0.37	0.37	0.37
Llama3	0.54	0.74	0.64	0.29	0.29	0.29

Table 3: Results for Argument Relationship Classification (Task 3) showing F1 scores. Left: binary classification (support vs attack); Right: 4-way classification (explicit/implicit support/attack). The best F1 scores per data set are bolded. 1-shot and 5-shot results are averaged over five runs. The majority baseline is defined as predicting the most frequent class in the training data.

[8] With F1 standard deviations ranging from 0.01 to 0.015 across the folds, indicating stability

Comment	Argument	Topic
I think every woman and anyone that's for abortions, that has a voluntary abortion should have every reproduction organ removed from their body [...]	Unwanted babies are ill-treated by parents and/or not always adopted	AB
Obama is another Hitler. There is not an ounce of capitalism or freedom in him. Why won't anybody in the media talk against him? Its because of the fairness doctrine. You're not allowed to speak against him. Stop listening to the liberal media.	Not eligible as a leader	OB

Table 4: Representative examples of false positive (FPs) predictions in Task 1, where the model falsely detected an argument in a comment. FPs often occur for comments that use strong/emotional language.

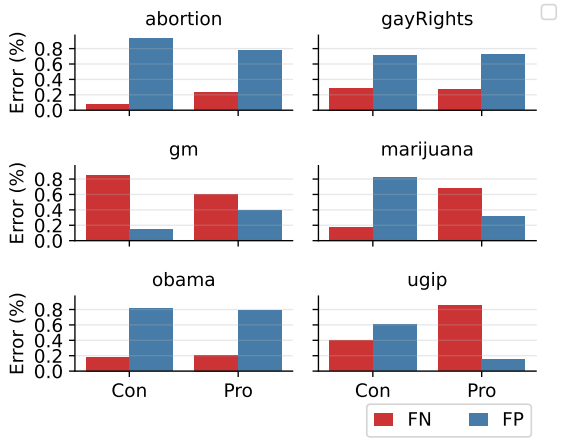

Figure 3: Proportion of false positive and false negative errors for Pro and Con arguments in each dataset.

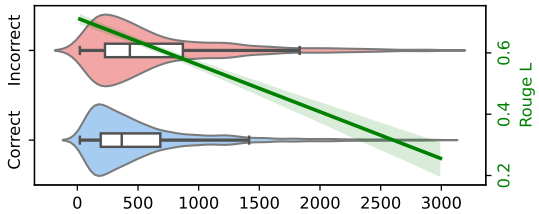

Figure 4: The effect of comment length on comment identification accuracy (Task 1; Violin/box plots) and argument extraction (Task 2; Rouge-L).

the argument. The models benefited from examples uniformly only for argument extraction (Task 2), but not in the classification tasks. Consistently, a fine-tuned RoBERTa model performed competitively with the LLMs on Task 2.

4.4 Error Analysis

Where exactly did LLMs fail on fine-grained argument detection, extraction and relation classification? To better understand this, we quantitatively and qualitatively inspected the predictions of the overall best k-shot model (Gemini, 5-shot). We systematically compared model predictions against gold labels, analyzing false positives (incorrectly identifying arguments) or false negatives (missing actual arguments) in Task 1, inspecting golden spans and predicted spans in the extraction task (Task 2), and the misclassification patterns in the relationship classification in Task 3.

False positives dominate in argument detection. As detailed in Figure 3, across the full dataset, false positive predictions (FP) of argument presence significantly outnumber false negatives (FN), accounting for approximately 66% of all errors. This pattern is particularly strong for Con arguments (which are against a topic), where 76% of all errors are FPs (62% for Pro arguments, in support of a topic). In other words: argumentative content is systematically over-predicted in comments that critique a given topic.

This tendency is particularly strong for the topics *Abortion Rights* and *Obama Presidency*, where FPs for Con arguments account for 92% and 81% of errors. The only exception is the topic *Gay Marriage*, where FNs heavily dominate Con arguments. These findings raise concerns for applications like content moderation and debate analysis systems, where stance-specific systematic misclassifications can lead to a skewed picture of opinions as well as erroneous classification of non-argumentative text as supporting particular positions.

Arguments are harder to identify in long comments. We observed a significant negative relationship between comment length (up to 3,000 characters to reduce the impact of outliers[9]) and model accuracy across both Task 1 (binary argument detection) and Task 2 (argument extraction). This is illustrated in Figure 4. For Task 1 we find a significant difference in mean length

[9] All significance results hold for stricter length thresholds (i.e., even fewer outliers), too, e.g., considering only comments of up to 750 characters.

Comment	Argument	Gold	Pred
Immorality should never has A SAY, should never be accepted as something normal. Marriage is between a man and a woman not between 2 men or 2 women. It is against our nature, against our God	It is discriminatory to refuse gay couples the right to marry	Implicit Attack	Explicit Attack
Homosexuality is considered risky behavior and cannot produce offspring and should not be considered with the same respect	Gay couples should be able to take advantage of the fiscal and legal benefits of marriage	Implicit Attack	Explicit Attack

Table 5: Task 3: Extracts of comments where Gemini incorrectly classified implicit attacks as explicit with strong/emotional language present in the comments.

for correctly vs incorrectly classified comments ($t{=}-12.103, p{<<}0.001$). For extraction (Task 2) performance we find a significant negative correlation between comment length and Rouge-L (Pearson's r $=-0.27, p{<<}0.001$). For downstream applications, this length effect could systematically bias system performance against more elaborate reasoning in comments, therefore potentially distorting the representation viewpoints expressed in texts. It also points to an opportunity for future work to address this gap.

Strong and emotional language. Manual inspection of 50 random mis-classified data points for each task, stratified across topics, revealed systematic language-related patterns in model failures. For Task 1, we observed frequent false positive predictions of arguments in emotionally charged or sarcastic comments (see examples in Table 4). Similar effects were observed for Task 3, where the model most often confused implicit attacks of Pro arguments with explicit attacks in cases where aggressive and offensive rhetoric overshadowed the actual argumentative content (see Table 5 for examples). Our findings suggest that strong and emotional language – which is common in online discussion – compromises model performance on the identification of argumentative content. Inappropriate reliance on surface-level cues can result in systematic bias in downstream applications.

5 Conclusion

We investigated how well LLMs can detect and understand the use of pre-defined arguments in online comments on contested topics. To do so, we separated the objective into three tasks: 1) assessing whether an argument is used in a comment, 2) extracting the exact span in which is it present, 3) and assessing whether the comment supports or attacks the argument.

We found that overall LLMs perform well on classification tasks (1, 3). While argument span extraction results in terms of Rouge-L appeared weak, manual analysis and additional validation through BERTScore indicates that models often extract argument-relevant spans which, however, may differ from the gold annotations. Task-specific fine-tuning yielded the best results, albeit with considerable computational and environmental costs. Interestingly, increased model size or examples did not consistently boost performance, though LLMs remained robust to example selection.

Our error analysis of one of the strongest LLMs revealed systematic limitations: Gemini systematically over-predicted arguments in emotional content, and performance degrades significantly with comment length. Both calls for follow-up work and raises concerns about reliability for a variety of downstream applications, such as content moderation tools or public opinion analysis where current models could systematically miss long or more nuanced arguments that require extended reasoning. Conversely, Gemini tended to overpredict argumentative content in strongly worded text, indicating overreliance on superficial linguistic cues. Such amplification strongly worded claims by LLMs may pose challenges for balanced, large-scale opinion analysis.

While we split argument analysis into atomic tasks to uncover specific weaknesses, end-to-end models remain appealing. Our results can guide their evaluation by identifying challenge cases for benchmarks and inform design decisions, such as prompt tuning or few-shot selection to address underrepresented arguments

In conclusion, our systematic evaluation provides a thorough overview of current performance, and systematic error analysis. It constitutes a basis for future work to explore how the identified shortcomings can be addressed for instance through improved prompting and fine-tuning, and to broaden our analysis to further topics and genres.

6 Limitations

The data used in this study is limited in scope, both in terms of size and the range of topics and arguments it covers. While this controlled data set enabled a detailed analysis of Large Language Models (LLMs) in argumentation tasks, it may not fully represent the complexity and diversity of real-world argumentation. Notably, the datasets employed were released in 2014, and may not capture more recent arguments or shifts in public opinion. For instance, the arguments related to the subtopic of gay marriage might no longer be relevant, especially given the legalization of gay marriage in the US in 2015, shortly after the data was released. On account of the limited data set size, we needed to conflate all datapoints for Task 1 to fine-tune our RoBERTa baseline. Due to time and cost constraints, as well as environmental considerations, we were only able to fine-tune one LLM (Llama3) on two of the proposed tasks.

7 Ethical Considerations

This study investigates the performance of LLMs in AM-related tasks on polarizing topics, which may involve sensitive or controversial discussions. We emphasize that the views in the data do not represent our own views, and that the findings and conclusions of this research are not intended to amplify or legitimize harmful, discriminatory, or unethical viewpoints. Instead, the goal is to evaluate and enhance the understanding of LLMs' capabilities in argument detection, classification and extraction, also analyzing their shortcomings and implications. Our research does not seek to endorse divisive or harmful opinions.

Acknowledgments

This paper was written with the support from the Melbourne Research Scholarship by the University of Melbourne provided to MG. This work was supported by the Australian Research Council Discovery Early Career Research Award (Grant No. DE230100761). We appreciate the computational resources provided for this research by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Henning Wachsmuth Martin Potthast Khalid Al, Khatib Yamen Ajjour, Jana Puschmann Jiani Qu Jonas Dorsch, and Viorel Morari Janek Bevendorff Benno Stein. 2017. Building an argument search engine for the web. *EMNLP 2017*, page 49.

Sarah M. Alsubhi, Areej M. Alhothali, and Amal A. Al-Mansour. 2023. AraBig5: The Big Five Personality Traits Prediction Using Machine Learning Algorithm on Arabic Tweets. *IEEE Access*, 11:112526–112534.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020a. From arguments to key points: Towards automatic argument summarization. *Preprint*, arXiv:2005.01619.

Roy Bar-Haim, Yoav Kantor, Lilach Eden, Roni Friedman, Dan Lahav, and Noam Slonim. 2020b. Quantitative argument summarization and beyond: Cross-domain key point analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 39–49, Online. Association for Computational Linguistics.

Filip Boltužić and Jan Šnajder. 2014. Back up your Stance: Recognizing Arguments in Online Discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Sarah Brown, Peter Anderson, and David Miller. 2023. Understanding the role of sampling parameters in language model generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 3456–3470.

Lucas Carstens and Francesca Toni. 2015. Towards relation based Argumentation Mining. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 29–34, Denver, CO. Association for Computational Linguistics.

Guizhen Chen, Liying Cheng, Luu Anh Tuan, and Lidong Bing. 2024. Exploring the Potential of Large Language Models in Computational Argumentation. *arXiv preprint*. ArXiv:2311.09022 [cs].

Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, Copenhagen, Denmark. Association for Computational Linguistics.

Adrian de Wynter and Tangming Yuan. 2024. "I'd Like to Have an Argument, Please": Argumentative Reasoning in Large Language Models. In *Computational Models of Argument*, pages 73–84. IOS Press.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ryo Egawa, Gaku Morio, and Katsuhide Fujita. 2020. Corpus for Modeling User Interactions in Online Persuasive Discussions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1135–1141, Marseille, France. European Language Resources Association.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 987–996, Portland, Oregon, USA. Association for Computational Linguistics.

Deniz Gorur, Antonio Rago, and Francesca Toni. 2024. Can Large Language Models perform Relation-based Argument Mining? *arXiv preprint*. ArXiv:2402.11243 [cs].

Ivan Habernal and Iryna Gurevych. 2017. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics*, 43(1):125–179.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762, Doha, Qatar. Association for Computational Linguistics.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. 2017. Analyzing the Semantic Types of Claims and Premises in an Online Persuasive Forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Hinton and Jean HM Wagemans. 2023. How persuasive is ai-generated argumentation? an analysis of the quality of an argumentative text produced by the gpt-3 ai text generator. *Argument & Computation*, 14(1):59–74.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yanting Liu, Xue Zhang, and Brian Thompson. 2023. An empirical study of temperature parameter impact on large language model outputs. *Transactions of the Association for Computational Linguistics*, 11:845–862.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xiaolong Meng, Jianxin Wu, and Kai Chen. 2023. Enhancing reproducibility in large language models: A study of temperature and top-p parameters. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1123–1135.

Yasser Otiefy and Alaa Alhamzeh. 2024. Exploring Large Language Models in Financial Argument Relation Identification. In *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing*, pages 119–129, Torino, Italia. Association for Computational Linguistics.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Ramon Ruiz-Dolz, Jose Alemany, Stella M. Heras Barbera, and Ana Garcia-Fornes. 2021. Transformer-based models for automatic identification of argument relations: A cross-domain evaluation. *IEEE Intelligent Systems*, 36(6):62–70.

Ramon Ruiz-Dolz and John Lawrence. 2023. Detecting argumentative fallacies in the wild: Problems and limitations of large language models. In *Proceedings of the 10th Workshop on Argument Mining*. Association for Computational Linguistics.

Ramon Ruiz-Dolz, John Lawrence, Ella Schad, and Chris Reed. 2024. Overview of DialAM-2024: Argument Mining in Natural Language Dialogues. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 83–92, Bangkok, Thailand. Association for Computational Linguistics.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: demonstrations*, pages 21–25.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Manfred Stede and Jodi Schneider. 2018. *Argumentation mining*. Springer.

Michiel van der Meer, Enrico Liscio, Catholijn M. Jonker, Aske Plaat, Piek Vossen, and Pradeep K. Murukannaiah. 2022. HyEnA: A Hybrid Method for Extracting Arguments from Opinions. In *HHAI2022: Augmenting Human Intellect*, pages 17–31. IOS Press.

Mei Zhang, Wei Chen, Yixuan Wang, and Hongzhi Li. 2023. Investigating the impact of decoding strategies on large language model performance: A systematic analysis. *arXiv preprint arXiv:2306.09265*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

A Lists of Arguments

Here, we present the complete list of pro and con arguments from the original datasets in Table 6.

B Text Length and Examples

This section includes extensive length statistics of the argumentative texts (comments from online discussions) in our data (Table 7), as well as two examples of such comments (1 for the abortion topic, 1 for the marijuana topic – Table 8).

C Prompts

We display the prompts used for our three tasks in Table 12 to Table 10.

D RoBERTa Fine-Tuning

We fine-tuned RoBERTa-base using the following configurations for each task:

- **Task 1: Argument Detection**
  - Training batch size: 16
  - Evaluation batch size: 64
  - Number of epochs: 3
  - Warmup steps: 500
  - Weight decay: 0.01
  - Evaluation strategy: per epoch
  - Save strategy: per epoch
  - Load best model at end: Yes
- **Task 2: Argument Extraction**
  - Training batch size: 16
  - Evaluation batch size: 16
  - Number of epochs: 10
  - Maximum sequence length: 512
  - N-best size: 16
  - Evaluate during training: No
  - Save checkpoints: No
  - Overwrite output directory: Yes
  - Save model every epoch: No
- **Task 3: Relationship Classification**
  - Training batch size: 16
  - Evaluation batch size: 64
  - Number of epochs: 3
  - Warmup steps: 500
  - Weight decay: 0.01
  - Evaluation strategy: per epoch
  - Save strategy: per epoch
  - Load best model at end: Yes
  - Optimization metric: F1
  - Optimization goal: maximize

All models were trained on a single NVIDIA V100 GPU using the RoBERTa-base checkpoint as the initial model.

E Parameter-efficient finetuning (PEFT) of LlaMA

For PEFT, we used an implementation of low-rank adaptation (LoRA) from Unsloth AI[10] with the following hyperparameters:

- load in 4 bit = False
- r = 16
- target modules = q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
- lora alpha = 16
- lora dropout = 0
- bias = none
- use gradient checkpointing = unsloth
- use rslora (rank stabilized LoRA) = False

The finetuning was performed with 5-fold cross-validation (data split of 60-20-20 for train-dev-test sets, with test splits covering the whole dataset). For the classification task, the splits were stratified. The training used 8-bit Adam as optimizer and the

[10] https://github.com/unslothai/unsloth

standard learning rate of 2e-4. The number of training steps was proportional to the data size, with loss falling to near-zero values as a stop signal, and roughly amounted to 3 full epochs for the classification task and 5 full epochs for the span extraction task.

F Text Length and Examples

This section includes extensive length statistics of the argumentative texts (comments from online discussions) in our data (Table 7), as well as two examples of such comments (1 for the abortion topic, 1 for the marijuana topic – Table 8).

G Prompts

We display the prompts used for our three tasks in Table 12 to Table 10.

H RoBERTa Fine-Tuning

We fine-tuned RoBERTa-base using the following configurations for each task:

- **Task 1: Argument Detection**
  - Training batch size: 16
  - Evaluation batch size: 64
  - Number of epochs: 3
  - Warmup steps: 500
  - Weight decay: 0.01
  - Evaluation strategy: per epoch
  - Save strategy: per epoch
  - Load best model at end: Yes
- **Task 2: Argument Extraction**
  - Training batch size: 16
  - Evaluation batch size: 16
  - Number of epochs: 10
  - Maximum sequence length: 512
  - N-best size: 16
  - Evaluate during training: No
  - Save checkpoints: No
  - Overwrite output directory: Yes
  - Save model every epoch: No
- **Task 3: Relationship Classification**
  - Training batch size: 16
  - Evaluation batch size: 64
  - Number of epochs: 3
  - Warmup steps: 500
  - Weight decay: 0.01
  - Evaluation strategy: per epoch
  - Save strategy: per epoch
  - Load best model at end: Yes
  - Optimization metric: F1
  - Optimization goal: maximize

All models were trained on a single NVIDIA V100 GPU using the RoBERTa-base checkpoint as the initial model.

I Parameter-efficient finetuning (PEFT) of LlaMA

For PEFT, we used an implementation of low-rank adaptation (LoRA) from Unsloth AI[11] with the following hyperparameters:

- load in 4 bit = False
- r = 16
- target modules = q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
- lora alpha = 16
- lora dropout = 0
- bias = none
- use gradient checkpointing = unsloth
- use rslora (rank stabilized LoRA) = False

The finetuning was performed with 5-fold cross-validation (data split of 60-20-20 for train-dev-test sets, with test splits covering the whole dataset). For the classification task, the splits were stratified. The training used 8-bit Adam as optimizer and the standard learning rate of 2e-4. The number of training steps was proportional to the data size, with loss falling to near-zero values as a stop signal, and roughly amounted to 3 full epochs for the classification task and 5 full epochs for the span extraction task.

The same prompts and example/label formats were used for finetuning as for the zero-shot and few-shot experiments (see Appendix G).

[11] https://github.com/unslothai/unsloth

Data set	Pro Arguments	Con Arguments
GM	It is discriminatory to refuse gay couples the right to marry. Gay couples should be able to take advantage of the fiscal and legal benefits of marriage. Marriage is about more than procreation, therefore gay couples should not be denied the right to marry due to their biology. Others	Gay couples can declare their union without resort to marriage. Gay marriage undermines the institution of marriage, leading to an increase in out-of-wedlock births and divorce rates. Major world religions are against gay marriages. Marriage should be between a man and a woman. Others
UG	Likely to be seen as a state-sanctioned condemnation of religion. The principles of democracy regulate that the wishes of American Christians, who are a majority, are honored. "Under God" is part of the American tradition and history. America is based on democracy and the pledge should reflect the belief of the American majority Others	Implies ultimate power on the part of the state. Removing "under God" would promote religious tolerance. Separation of state and religion. Others
AB	Abortion is a woman's right. Rape victims need it to be legal. A fetus is not a human yet, so it's okay to abort. Abortion should be allowed when a mother's life is in danger. Unwanted babies are ill-treated by parents and/or not always adopted. Birth control fails at times, and abortion is one way to deal with it. Abortion is not murder. Mother is not healthy/financially solvent. Others	Put the baby up for adoption. Abortion kills a life. An unborn baby is a human and has the right to live. Be willing to have the baby if you have sex. Abortion is harmful to women. Others
GR	Gay marriage is like any other marriage. Gay people should have the same rights as straight people. Gay parents can adopt and ensure a happy life for a baby. People are born gay. Religion should not be used against gay rights. Others	Religion does not permit gay marriages. Gay marriages are not normal/against nature. Gay parents cannot raise kids properly. Gay people have problems and create social issues. Others
MA	Not addictive. Used as a medicine for its positive effects. Legalized marijuana can be controlled and regulated by the government. Prohibition violates human rights. Does not cause any damage to our bodies. Others	Damages our bodies. Responsible for brain damage. If legalized, people will use marijuana and other drugs more. Causes crime. Highly addictive. Others
OB	Fixed the economy. Ending the wars. Better than the Republican candidates. Makes good decisions/policies. Has qualities of a good leader. Ensured better healthcare. Executed effective foreign policies. Created more jobs. Others	Destroyed our economy. Wars are still ongoing. Unemployment rate is high. Healthcare bill is a failure. Poor decision-maker. We have better Republicans than Obama. Not eligible as a leader. Others

Table 6: Pro and Con Arguments for All Subtopics and Datasets

Topic	Min Characters	Max Characters	Mean Characters	Median Characters
Gay Marriage	33	2,454	683.06	672.0
UGIP	31	1,317	486.21	405.0
Gay Rights	44	6,441	772.25	473.0
Abortion	33	23,055	981.52	536.0
Marijuana	21	3,658	731.44	495.0
Obama	53	14,904	846.31	434.0

Table 7: Text Length Statistics of comments across topics

Topic	Comment
Abortion	Why should you kill a innocent baby? That is exactly what abortion is. Even though the mother does not want the baby, she should still have it. Most of the people who want an abortion and never go through with it, actually say they would regret killing the baby. Should America become ‘"I get to do whatever I want to just because I can"?
Marijuana	I believe marijuana should be legal for many reasons. First of all it is proven that it helps with different things medically such as when going through chemo it gives you appetite, it helps with pain control etc. Also i feel personally that alcohol is more dangerous then marijuana. I have seen many people killed from drunk drivers and it is a shame that so many people drive drunk. But, i have never heard of anyone dying from smoking too much weed, killing someone from an accident because they smoked weed, or anything like that.. Marijuana is a natural herb and it is legal in many other places and could possible make some money for the country if legalized!

Table 8: Example Comments for Abortion and Marijuana Topics

Analyze whether the following comment about {topic} contains a specific argument.
Argument to check for: {argument}
Instructions:
1. Determine if the comment explicitly or implicitly uses the given argument
2. Assign a binary label:
- 1 if the argument is present
- 0 if the argument is not present
Requirements:
- Only use 1 or 0 as labels
- Provide output in valid JSON format
- Do not repeat or include the input text in the response
- Focus solely on the presence/absence of the specific argument
Return your analysis in this exact JSON format:
`"id": "id", "label": label_value`
Analyze the following comment in relation to the given argument:

Table 9: Prompt for Task 1

Task: Text Span Identification for Arguments about {topic}
Target Argument: {argument_text}
Role: You are an expert in argument analysis and logical reasoning,
specializing in identifying rhetorical patterns in social discourse.
Step-by-Step Instructions:
1. Read the input text carefully
2. Locate exact text spans that:
- Directly reference the target argument
- Express the same idea as the argument
3. Extract the precise text span
4. Format the output according to specifications
Critical Requirements:
- Extract EXACT text only (no paraphrasing)
- Include COMPLETE relevant phrases
- Use MINIMUM necessary context
- Maintain ORIGINAL formatting
- Return VALID JSON only
Output Schema:
{ "id": "{id}",
"span": "exact_text_from_comment" # must be verbatim quote
}
Input Text:

Table 10: Prompt for Task 2

Task: Binary Classification of Arguments about {topic}
Input Text: {comment_text}
Target Argument: {argument_text}
Role: You are an expert in argument analysis and logical reasoning,
specializing in identifying rhetorical patterns in social discourse.
Step-by-Step Instructions:
1. Read the input text thoroughly
2. Evaluate the text's relationship to the target argument, examining:
- Direct support or opposition
- Implicit agreement or disagreement
3. Make a binary classification decision
4. Format the output according to specifications
Classification Rules:
- Label = 5: Comment supports/agrees with argument
- Label = 1: Comment attacks/disagrees with argument
Critical Requirements:
- Use ONLY specified labels (1 or 5)
- Do NOT quote or repeat input texts
- Return VALID JSON only
Output Schema: { "id": "{id}", "label": label_value # must be 1 or 5 without quotes }
Input Text:

Table 11: Prompt for Task 3 - Binary

Task: Classification of Arguments about {topic}
Input Text: {comment_text}
Target Argument: {argument_text}
Role: You are an expert in argument analysis and logical reasoning,
specializing in identifying rhetorical patterns in social discourse.
Step-by-Step Instructions:
1. Read the input text thoroughly
2. Evaluate the text's relationship to the target argument, examining:
- Direct support or opposition
- Implicit agreement or disagreement
3. Make a binary classification decision
4. Format the output according to specifications
Classification Rules:
- Label = 5: Comment supports/agrees with argument
- Label = 4: Comment supports/agrees with argument implicitly/indirectly
- Label = 2: Comment attacks/disagrees with argument implicitly/indirectly
- Label = 1: Comment attacks/disagrees with argument
Critical Requirements:
- Use ONLY specified labels (1 or 5)
- Do NOT quote or repeat input texts
- Return VALID JSON only
Output Schema: { "id": "{id}", "label": label_value # must be 1, 2, 4 or 5 without quotes }
Input Text:

Table 12: Prompt for Task 3 - Full Scale