ArgCMV: An Argument Summarization Benchmark for the LLM-era

Anonymous ACL submission

Abstract

Key point extraction is an important task in argument summarization which involves extracting high-level short summaries from arguments. Existing approaches for KP extraction have been mostly evaluated on the popular ArgKP21 dataset. In this paper, we highlight some of the major limitations of the ArgKP21 dataset and demonstrate the need for new benchmarks that are more representative of actual human conversations. Using SoTA large language models (LLMs), we curate a new argument key point extraction dataset called **ArgCMV** comprising of $\sim 12K$ arguments from actual online human debates spread across $\sim 3K$ topics. Our dataset exhibits higher complexity such as longer, co-referencing arguments, higher presence of subjective discourse units, and a larger range of topics over ArgKP21. We show that existing methods do not adapt well to ArgCMV and provide extensive benchmark results by experimenting with existing baselines and latest open source models. This work introduces a novel KP extraction dataset for long-context online discussions, setting the stage for the next generation of LLM-driven summarization research.[1]

1 Introduction

Online platforms such as Twitter and Reddit have transformed public debate into a stream of loosely structured and rapidly evolving discussions. From deliberations on policies, to debates about sports and movies, millions of users post arguments that policy-makers, content moderators, and recommendation systems need to summarize and assimilate, in order to perform downstream actions. Automatically distilling these conversation threads into focused argument summaries is therefore crucial for tasks such as analytics, proactive moderation, and personalized content recommendation (Bhatia et al., 2014; Egan et al., 2016; Lee et al., 2020; Schluger et al., 2022).

A popular formalization of this goal of argument summarization (ArgSum) is through Key Point Analysis (KPA) where the task is to extract concise, and salient "key points" (KPs) which are defined as *high level summaries of arguments* (Bar-Haim et al., 2020a). Although the KPA task was introduced over five years ago, most research still relies on the ArgKP21 (Bar-Haim et al., 2020a) dataset as the sole evaluation metric. ArgKP21 consists of debate arguments related to various controversial topics along with its stance ('pro' or 'con'), human extracted 'gold standard' key points for each argument, and a label indicating whether an argument is associated with a particular key point.

Despite the rigorous curation process, we find that ArgKP21 has certain limitations. First, the arguments in ArgKP21 are short sentences which lack the complexity of actual human debates. Next, debates involve back-and-forth between the two parties and counter-arguments often have added context related to the arguments presented by the opponent. ArgKP21's independent arguments fail to account for this dynamic nature of conversations. Relatedly, we find that ArgKP21 does not fully test the long-context understanding of models. Finally, ArgKP21 is also not representative of conversations occurring on online discussion forums like Reddit. Prior research has highlighted the need for summarization tools to help users and moderators effectively consume (Zhang and Cranshaw, 2018), curate (Choi et al., 2023), and engage (Zhang et al., 2017; Im et al., 2020) in online discussions. This need is exacerbated by the ever-increasing volume and topical diversity of user-generated content within online communities. These challenges highlight a clear need for a new ArgSum benchmark which addresses the limitations of ArgKP21 and better tests the long-context understanding of SoTA language models.

In this paper, we present **ArgCMV**, a key point-based ArgSum benchmark consisting of

[1] Datasets and code will be released upon acceptance.

long-context, multi-turn arguments from actual online human debates sourced from Reddit's `r/ChangeMyView`. `r/ChangeMyView` is a popular forum for user debates on controversial topics, and has been widely used by the NLP community as a reliable data source for task such as persuasion modeling (Tan et al., 2016; Mirzakhmedova et al., 2023), counter-argument generation (Yeginbergen et al., 2025). Overall, ArgCMV features a higher topic diversity and argument complexity compared to ArgKP21. We use source which reframing (Peguero and Watanabe, 2024).

We obtain the ground truth KPs using a combination of SoTA language models (GPT-4o-mini/GPT-4o) followed by human validation. We show that ArgCMV is a much harder benchmark through empirical analysis and comparing the performance of existing KP extraction models. We find that existing models fail to adapt to ArgCMV due to its complexity and long-context nature. We also report the performance of smaller open-source models on ArgCMV, observing the same trends.

In this work, **we make four key contributions**:
- We introduce **ArgCMV**, an ArgSum dataset for key point extraction. ArgCMV contains actual multi-turn, long-context human conversations.
- We provide statistically and theory-driven evidence for the limitations of ArgKP21 in comparison to the improved complexity of ArgCMV.
- We perform rigorous benchmarking of existing SoTA KP extraction models and open-sourced LLMs on ArgCMV, showing that they fail to adapt.
- Finally, we make the standard train, dev, test splits of our dataset along with the extracted KPs and their mappings publicly available[2] with appropriate licensing to enable future research.

We believe the introduction of ArgCMV establishes a foundation for significant LLM-based advances in argument summarization, by serving as a reliable and competitive benchmark.

2 Related Work

2.1 Existing ArgSum Datasets

Since the release of the seminal ArgKP21 corpus by (Bar-Haim et al., 2020a), several researchers have released other argument mining datasets compiled from various sources. DebateSum (Roush and Balaji, 2020) contains 180K formal debates from university debate camps and their associated evidence (used as the reference summary), OpenDebateEvidence (Roush et al., 2024) further expanded this to more than 3.5 million documents and evidence. IAM (Cheng et al., 2022) released a dataset consisting of over 1K Wikipedia articles, each labeled for evidence, stance, and claim. Guo et al. (2023) enhanced the IAM dataset with an evidence type between evidence and claim to formulate the QAM dataset. Though large and carefully curated, none of these datasets are representative of online user discussion, and while QAM has been used for KP extraction, the datasets are not targeted specifically for the KPA task.

2.2 KP Extraction and Matching Models

Friedman et al. (2021) formally introduced KP matching and KP generation task based on the ArgKP21 dataset. On the matching front, SMatchToPR (Alshomary et al., 2021) used a contrastive loss to train a Siamese network model for this task. Enigma (Kapadnis et al., 2021) used a combination of transformer embeddings and TF-IDF, Part of Speech (POS) features as inputs to a neural network. On the generation/extraction side, (Bar-Haim et al., 2020b) proposed an extractive summarization technique which first selects high-quality KP candidates and then matches them to arguments. Li et al. (2023) performed abstractive summarization by using a combination of UMAP-dimensionality reduction and `BERTopic` to cluster arguments, and then trained a Flan-T5 (Cheng et al., 2022) model to generate key points for each cluster. (Li et al., 2024b) formulated a pair-wise task to generate shared KPs between arguments, and followed by a graph partitioning algorithm. More recently, Altemeyer et al. (2025) proposed using LLMs like GPT4 as possible alternatives for KP generation and evaluation. We use Li et al. (2024b) as our baseline, and while our generation is based on Altemeyer et al. (2025), they only consider GPT4 and do not evaluate smaller models for these tasks.

2.3 LLM-based argument summarization

Ziegenbein et al. (2024) used LLMs to generate snippets from search results and neutralize them into objective sentences. Li et al. (2024a) compared different LLMs on four argument mining and summarization tasks. Beyond argument summarization LLMs have been extensively used for summarizing news articles (Zhang et al., 2024a,b), scientific articles (Tang et al., 2023; Van Veen et al., 2023), and books (Chang et al., 2023). We use LLMs for the task for KP extraction on our dataset.

[2]Will be released upon acceptance.

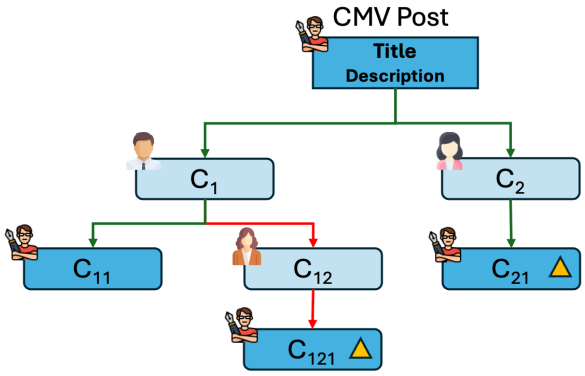

Figure 1: An example post on `r/ChangeMyView`, with **green** paths highlighting valid dialogues and **red** paths highlighting invalid branches for our data collection.

3 The `r/ChangeMyView` forum

`r/ChangeMyView` (CMV hereafter) is a Reddit community intended for users who are open to changing their opinion on a topic. Each post consists of the original poster (or OP) sharing their stance on a topic, following which other users try to present arguments aiming to persuade the OP into an opinion change. If a comment is able to change their view, the OP can report it by replying to that comment with a Δ symbol or by using `!delta`. Each CMV discussion or *thread* starts with the OP posting their opinion. The post contains and a title, which is generally a single sentence starting with "CMV:" followed by a paragraph elaborating on the subject and any supporting arguments. Other users (and the OP) can reply either directly to the original post or to any previous reply contributing to the discussion. We call a chain of replies starting from the post to a comment with no replies as a *branch*. We only collect branches where the OP makes at least one comment. Following Mirzakhmedova et al. (2023)'s terminology, we only collect *dialogues*—branches with only two unique users. Figure 1 illustrates an example CMV thread.

4 Limitations of ArgKP21

ArgKP21 (Bar-Haim et al., 2020a) is one of the most popular benchmarks for the task of key point extraction. The dataset consists of around 7000 crowd-sourced arguments belonging to 28 controversial topics, each labeled for its stance. For each of the topics, expert debaters were asked to create a set of key points for each topic. Finally, crowd workers were asked to map each key point with all its associated arguments.

In this section, we discuss some of the limitations of ArgKP21 dataset, and discuss how `r/ChangeMyView` arguments can serve as an effective alternative source capable of testing the full-capabilities of LLMs.

One of the frameworks to study arguments is to break them down into elementary units (EU) often referred to as **Argumentative Discourse Units (ADUs)** (Morio et al., 2019). (Morio et al., 2019) identifies five ADUs for online discussions namely: Fact, Policy, Rhetorical, Testimony, and Value. Arrangements of these units result in different persuasion strategies which have been linked to the overall effectiveness of arguments (Mirzakhmedova et al., 2023). From the perspective of key point extraction, Facts are important due to their objective nature. Similarly, Value and Policy statements provide context of value judgments and action suggestions on the topic. However, Testimony and Rhetorical Statements are often more related to the speaker's personal opinions and might be considered less important for key point analysis. Thus, the variety of ADU types in an argument contributes to the overall complexity of key point extraction task.

To compare this aspect between ArgKP21 and CMV data, we first label the arguments from each dataset with ADU types. For this, we use the model from (Mirzakhmedova et al., 2023) to extract ADU units from a given argument. ADU mining is a token labeling task where each token is assigned to the ADU classes using BIO labeling. In Figure 2 we show the relative distribution of each ADU type between the two datasets.

We find that our dataset shows a higher diversity of ADU types with the presence of Rhetorical and Testimony types which are almost absent in ArgKP21. This is due to the fact that CMV contains arguments from online users who often refer to personal experiences in their comments. On the other hand, ArgKP21 features a higher proportion of Policy based arguments. We also perform a χ^2 (Pearson, 1900) test and find the difference in proportion to be statistically significant ($p < 0.05$).

Next, in Table 1, we present some additional statistics to compare the two datasets. First, we note that CMV arguments are over 10 times longer than ArgKP21, as the later generally consists of single sentence arguments, for example: *There are issues more important to fund than space exploration*. CMV arguments on the other hand, contain a more comprehensive opinion of the user. We also find that on an average CMV arguments contain a higher diversity of ADU units individually as

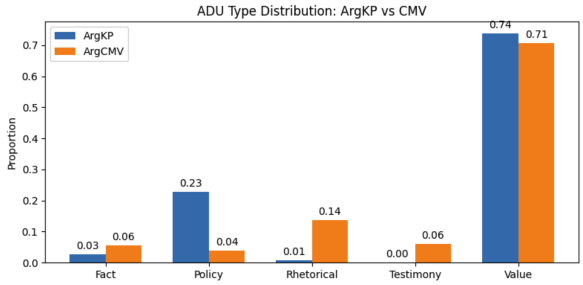

Figure 2: Distribution of ADU units for the two datasets. Subjective units such as Rhetorical are almost non-existent in ArgKP21.

see through the mean number of ADUs and Mean entropy statistics.

In addition to this, *IBM-Rank-30k* (Gretz et al., 2020), the source of ArgKP21 arguments was created as a part of a curated annotation task where crowd-workers were incentivized to generate high quality arguments on a topic. The uncontrolled nature of CMV arguments make them a better representative of actual human conversations, while the clear community rules and moderation prevent excessive noise in the data. Further, the conversational nature of Reddit produces multi-turn conversations, where users build upon their previous arguments in a to-and-fro debate meaning the future arguments can often contain references to the previous ones. This property is completely missing in ArgKP21 arguments.

Based on the aforementioned reasons, we argue that CMV provides arguments with longer length, complexity, and a more realistic representation of online user conversations making it a challenging source for long-context key point extraction.

Metric	ArgCMV Mean	ArgKP21 Mean	χ^2 pval
Mean number of tokens	196.75	19.61	*
Mean number of ADUs	4.27	1.22	*
Mean number of unique ADUs	2.09	1.17	*
Mean ADU entropy	0.87	0.24	*

Table 1: Comparison of argument complexity metrics between ArgCMV and ArgKP21 datasets. All differences between datasets were significant. ($^*p < 0.05$)

5 The ArgCMV Dataset

We now present our methodology for preparing the ArgCMV dataset in Figure 3.

Data collection:. We begin by crawling dialogue-only threads from `r/ChangeMyView`. All the data was collected between January 2020 to December 2020. Each thread consists of an original post (OP) followed by a sequence of back-and-forth comments. All messages authored by the OP—including the root post and any subsequent replies—are treated as *pro* arguments with stance +1, and every comment written by any other user is treated as a *con* argument with stance −1. This simple per-author split provides two coherent argument pools whose stances are explicit and mutually opposed.

Step 1. Key-point extraction: For every thread we send each stance-specific argument pool to an ExtractionAgent. The agent receives the full text of the pool and returns a concise list of key points (KPs) that summarize the reasoning of that side. Running the agent separately on the +1 and −1 pools yields two disjoint KP lists: *pro KPs* and *con KPs*, forming the candidate distillation of arguments that will be linked back to comments.

Step 2. Key-point mapping: Multiple users may have a shared set of ideas and key points within their arguments. In order to capture these, we use a MappingAgent. Given a single comment and the KP list that matches its stance, the MappingAgent decides which KPs—if any—are expressed in that comment. In order to minimize hallucination during annotation, we process one argument at a time. Because each comment is mapped independently, the same KP can be linked to arguments from multiple users, allowing us to capture cross-user convergence on shared ideas. After mapping all comments in a thread, we create a structured record that contains the original post, every comment, and the set of KPs it realizes. Repeating this procedure for every thread yields the final ArgCMV dataset.

Metric	Mean$_{a_1}$	Mean$_{a_2}$	Pearson r
KP Precision	87.30	98.66	0.445
KP Recall	92.10	96.33	0.530

Table 2: Manual validation results for key points extracted by `gpt-4o-mini` and matched by `gpt-4o`. We report mean percentages for the two annotators along with the Pearson's correlation for inter-annotator agreement.

Step 3. Manual Validation: The first and second authors of the paper further validated the correctness of the LLM output by performing a manual human validation. Two human annotators were asked to label a random sample of 50 arguments and the extracted KPs on the following three aspects: 1)

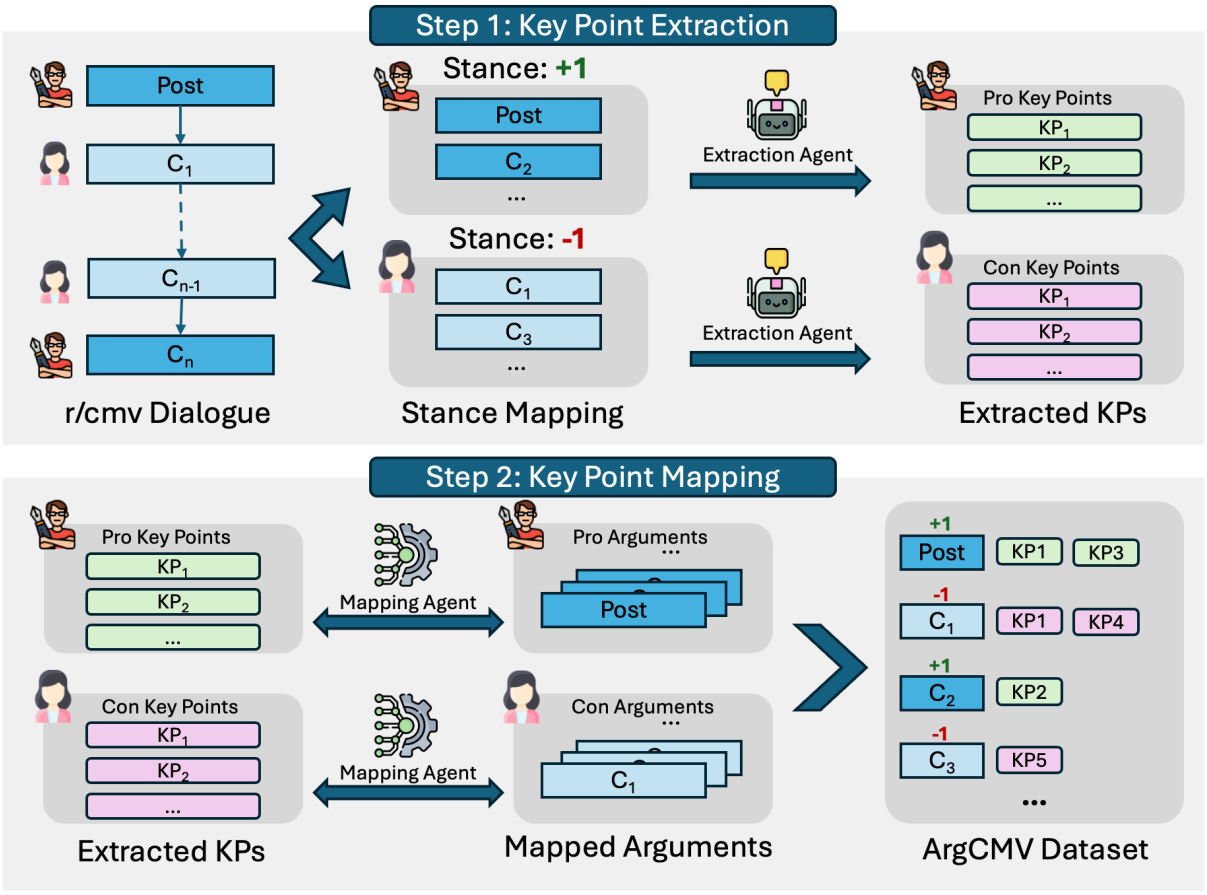

Figure 3: Our two-step LLM-based pipeline used to extract key points (KPs) for the ArgCMV dataset. In step 1, our ExtractionAgent processes all the arguments for a given topic and stance to generate a set of key points. In step 2, the MappingAgent identifies all the KPs which belong to a given argument. We use this approach to obtain the ground truth KPs for our ArgCMV dataset.

KP Precision: The proportion of the matched KPs which actually belong to the argument 2) KP Recall: The proportion of the matching candidate KPs which were actually matched to the arguments 3) KP-Redundancy: A binary label to check if any two KPs are semantically overlapping with each other. We report the results of the manual validation in Table 2. We find that there are almost no cases of redundant KPs in our sample. For precision and recall, we obtain very high values showing that GPT4 outputs are reliable. We calculate inter-annotator agreement using Pearson's correlation (Schober et al., 2018) and get moderate agreement in both cases.

Model Selection: In order to show that LLMs are effective at the KP extraction task and to select the best model for extraction and mapping steps, we perform a few-shot KP extraction on the standard test set on ArgKP21. We report these results in Table 4. For the extraction, we follow the prompting strategy proposed by (Altemeyer et al., 2025), where similar arguments are first clustered (using the USKPM strategy (Li et al., 2023)) and then the LLM is prompted to extract the representative KP for each cluster. The exact prompts are shown in the Appendix A. Following recent work which show GPT4 (Hurst et al., 2024) models as a reliable proxy to human annotations, we compare two variants `gpt-4o` and `gpt-4o-mini`. We compare these against two recent baselines the SKMP (Li et al., 2023) and (Li et al., 2024b), and report the standard metrics (described in detail in Section 6.2). We find that our `gpt-4o-mini` model outperforms (Li et al., 2024b) on the semantics-based soft metric. And, while it perfoms worse compared to the `FLanT5-xxl` model from (Li et al., 2023), the performance is still decent considering the few-shot setting. As a result, we use it as our ExtractionAgent. For the MappinAgent we use the `gpt-4o` model.

Metric	ArgCMV	ArgKP21
Number of arguments	12 262	6 549
Number of topics	3 131	31
KPs per argument ($\mu \pm \sigma$)	2.80 ± 1.76	0.76 ± 0.52
Number of debates	4 387	–
Turns per debate ($\mu \pm \sigma$)	1.68 ± 0.84	–

Table 3: Comparison of key metrics between ArgCMV and ArgKP21. ArgCMV contains substantially more arguments and far greater topical diversity.

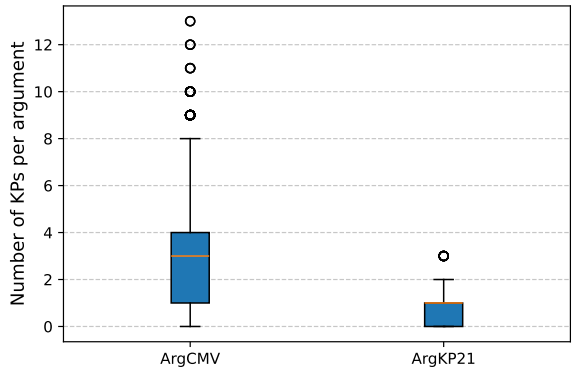

Figure 4: Box-plot showing the distribution of number of KPs for each argument for ArgCMV and ArgKP21 datasets. ArgCMV arguments have higher number of matched KPs.

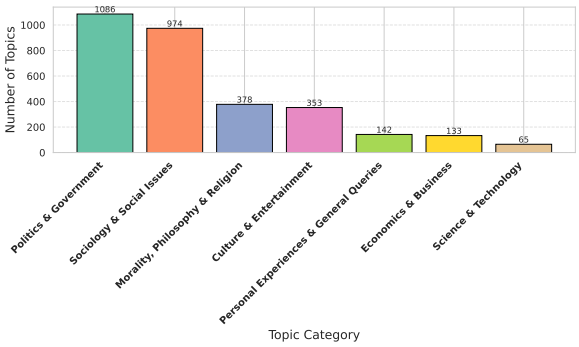

Figure 5: Distribution of topics across 7 high-level categories for ArgCMV. Politics and Sociology related posts have the highest frequency.

Dataset Statistics: Finally, our dataset contains a total of 12262 arguments, coming from 3131 topics. In Table 3 we provide a comparison between different statistics with ArgKP21. ArgCMV consists of almost twice the number of arguments, and 1000 times more topics compared to ArgKP21. As seen in Section 4, due to the larger length and complexity of arguments in our dataset, we also obtain a higher number of KPs for each argument. In Figure 4, we visualize this using a box-plot. We find that over 95% of arguments in ArgKP21 are associated with a single or no KP. On the contrary, over $\sim$ 75% arguments in ArgCMV have more than one KP associated with it. In order to further show the diversity and richness of our dataset, we label each topic with one of seven broad categories inspired by (Hidey and McKeown, 2018). For this, we perform few-shot prompting using the Gemma2 (Team et al., 2024) model. We report the distribution across topics in Figure 5. Thus, apart from offering a higher argument complexity, ArgCMV also has a much broader variety of topics making it more generalizable.

To create the splits, we randomly divide the topics in the ratio of 80/10/10. The final split sizes were: 9845/1172/1245 arguments for train/dev/test respectively.

6 ArgCMV Benchmarking

Having obtained the ground-truth KPs for our dataset, we compare the performance of different existing approaches on our new dataset. Additionally, we also compare the performance of different small language models (SLMs) to check if small-scale ($\sim 10B$ parameters) models are able to effectively extract KPs from our dataset.

6.1 Models

Graph Partitioning-based KPA: We use the approach descibed by (Li et al., 2024b) as our baseline model, as it achieves the best performance on ArgKP21 with the same model size (`FlanT5-large`). This approach first trains a Flan-T5 (Chung et al., 2024) model for the task of shared key point detection, where the input is a pair of arguments, along with their topic and stance and the model predicts whether the two arguments share a key point. In case they do, the model should also output the shared key point. In other words, $input_{ij} = topic \mid stance_i\ arg_i \mid stance_j\ arg_j$.

$$\text{output}_{ij} = \begin{cases} \text{Yes. } \{\text{kp}_{ij}\}, & \text{shared KP} \\ \text{No.}, & \text{no shared KPs.} \end{cases}$$

We convert our dataset into their input and output format to train the model. Since, most of the arguments in our datasets are mapped to multiple KPs, we frequently encounter cases where two arguments share more than one KP between them. As (Li et al., 2024b) don't specifically mention how they handle such cases (most likely since this is much rare in case of ArgKP21), we select the

Dataset	Model	Rouge-1	Rouge-2	sP	sR	sF1
ArgKP21	**Previous Approaches**					
	$\text{SKPM}_{\text{Flan-T5-large}}$	31.4	9.1	57.00	62.00	60.00
	$\text{SKPM}_{\text{Flan-T5-xxl}}$	32.8	9.7	**70.00**	**71.00**	**71.00**
	(Li et al., 2024b) Flan-T5-base	40.85	14.18	62.31	58.59	60.37
	(Li et al., 2024b) Flan-T5-large	**55.13**	**23.11**	61.65	61.69	61.66
	Few-Shot LLM					
	`gpt-4o-mini`	39.31	10.87	64.15	64.07	64.08
	`gpt-4o`	37.12	11.23	60.33	57.59	58.80
ArgCMV	(Li et al., 2024b) Flan-T5-base	21.96	6.62	47.74	42.0	44.48
	(Li et al., 2024b) Flan-T5-large	17.50	6.39	60.61	41.13	48.69
	`gemma-2-9b`	**51.77**	**21.20**	**60.76**	**60.85**	**60.76**
	`llama-3-8b`	41.81	14.93	57.07	56.92	56.95
	`mistral-nemo-2407`	47.67	17.91	54.92	54.47	54.62

Table 4: Performance comparison on the ArgKP21 and ArgCMV dataset using Rouge and soft matching metrics (sP, sR, sF1). On ArgKP21 `gpt-4o-mini` model is able to achieve close to SoTA performance with no training/fine-tuning. On ArgCMV existing methods fail to adapt, and `gemma-2-9b` shows the best few-shot performance. For the SKMP model, we take the results from (Li et al., 2023).

output KP using the following approach: while iterating on argument pairs, we keep a track of all the KPs which have been previously picked in the dataset. Then, when we encounter an argument pair with multiple shared KPs, we remove the ones which have been picked at least once and randomly sample a KP from the remaining. In case, all the shared KPs have been included, we simply sample a KP randomly.

The second adjustment we need to make is to their graph partitioning algorithm. Their initial K-Means based partitioning sets the initial number of clusters to be the number of ground truth KPs for the topic, stance combination. However, this doesn't work for our dataset where number of KPs is much higher than the number of arguments. To avoid this, we adjust the initial number of K-Means clusters to be half the number of arguments in the graph. Due to this change and less number of arguments in certain topics, we get cases where the algorithm fails to find any KPs for the topic. We do not consider these topics while calculating the overall metrics to ensure fair evaluation.

Small Language Model-based KPA: Given the long-context nature of our dataset, and the presence of multiple KPs for each argument make language models as the ideal choice for the KP extraction task. Given the recent popularity of open source small language models, we experiment with three candidates. We pick Gemma2 `gemma-2-9b` (Team et al., 2024), Llama3 (Grattafiori et al., 2024) `llama-3-8b`, and Mistral `Mistral-Nemo-2407` (MistralAI and NVIDIA, 2024) models for this experiment. We use the instruction-tuned versions of all the models. To generate the KPs, we base our prompt on (Altemeyer et al., 2025), however instead of generating a single KP per cluster, the model is now allowed to generate multiple KPs, if needed. Further, instead of using any clustering approach, we create clusters by grouping all the arguments from the same user together. We show the complete prompt in the Appendix A.

6.2 Metrics

Based on prior work (Li et al., 2023, 2024b), we report Rouge (Lin, 2004), soft-Precision/Recall/F1 metrics. To calculate Rouge metrics, we concatenate all the ground-truth KPs as well as the generated KPs, similar to (Li et al., 2023). The soft-precision (**sP**) score is calculated by finding the maximum similarity score with a reference KP for each generated KP, and then averaging all the values. Similarly, the soft-recall (**sR**) is calculated by the taking the mean of the maximum similarity score with a generated KP for each ground truth KP. The similarity score is calculated using the BLEURT-20 (Sellam et al., 2020) model. soft-F1 (**sF**) is the harmonic mean of **sP** and **sR**. Formally,

$$\text{sP} = \frac{1}{|KP_{\text{gen}}|} \sum_{kp_g \in KP_{\text{gen}}} \max_{kp_r \in KP_{\text{ref}}} \text{Sim}(kp_g, kp_r)$$

$$\text{sR} = \frac{1}{|KP_{\text{ref}}|} \sum_{kp_r \in KP_{\text{ref}}} \max_{kp_g \in KP_{\text{gen}}} \text{Sim}(kp_r, kp_g)$$

6.3 Implementation Details

For all the GPT models, we use OpenAI's API.[3] We estimate the cost to be ≈100 USD. We set

[3] https://openai.com/api/

the temperature parameter to be 0 to ensure reproducibility. For all the GPU experiments, we use NVIDIA A100 (40GB) GPUs which were accessed through a shared slurm cluster. For (Li et al., 2024b), we use the same hyper-parameters (except while training the Flan-T5-large on ArgCMV, where we had to reduce the batch size to 4 to avoid memory issues) as described in their paper and directly use their code for all the experiments. For running the open source models, we use Unsloth (Daniel Han and team, 2023) for fast inference. We use the 4-bit quantized versions of all the models. For these experiments, we set *temperature* = 0.1, *max_new_token* = 256, and *top_p* = 0.94, and perform un-batched inference.

6.4 Results

In the bottom section of Table 4, we report the results of the KP extraction task on our ArgCMV dataset. First, we observe that both the models from (Li et al., 2024b) achieve significantly lower performance on our dataset, compared to ArgKP21. This shows that existing KP extraction methods do not adapt well to our dataset. Specifically, we notice that the sP values are generally higher than sR for both the base and the large model. This means the KPs generated by their model have a close match with one of the reference KPs, however many of the reference KPs don't have a good match among the generated ones. We believe this is due to the fact that their approach is designed specifically for ArgKP21-style arguments where most of the arguments are mapped to a single KP.

Next, we compare the performance of our three SLMs. We find that the performance of all the three models is higher than our baseline, demonstrating the out-of-the-box ability of SLMs to extract KPs from our dataset. Also, the Gemma2 model outperforms all the other models across different metrics. Note that in the table we include the results for the run which results in the maximum $sF1$ value, although we don't observe large variations across runs.

7 Discussion and Implications

Summarization of online discussions: Recent research in human computer interaction (HCI) has underscored the challenges faced by end-users (Zhang and Cranshaw, 2018; Kumar et al., 2023) and content moderators (Jiang et al., 2019; Choi et al., 2023) in effectively navigating online discussions due to the large volume of content generated on online social media (OSM) platforms. Summarizing discussion threads on such platforms to highlight the core conversational outcomes and key points can help improve overall user experience by enabling effective consumption (Zhang and Cranshaw, 2018), curation (Choi et al., 2023), and engagement (Zhang et al., 2017; Im et al., 2020) in online discussions.

Our work has implications for the design of LLM-driven summarization tools for long-context online discussions. ArgCMV can be used to train summarization models for online forums like `r/ChangeMyView` or similar debate-oriented online platforms. Moreover, future work can apply our LLM-based KP extraction framework to other datasets and summarization tasks.

Data collection and labeling: Curation of ArgKP21 involved multiple professional debate experts and crowd-workers for argument generation and key point extraction/matching which involves high human effort. We remediate this issue using our hybrid dataset preparation pipeline which is shown to provide us with reliable ground truths. Collecting data from `r/ChangeMyView` allowed us to compile long-form debates without requiring any experts to manually write arguments for us. This approach not only saved us human-effort but it also helped us develop a benchmark which is more realistic than an expert-curated dataset. The effectiveness and flexibility of our approach demonstrates how future work can benefit by adopting our framework for other tasks as well, especially when data generation requires substantial human effort.

8 Conclusion

In this paper, we introduce ArgCMV, a new benchmark for key point extraction based on $\sim 12K$ arguments from real online debates across $\sim 3K$ topics. Unlike the widely used ArgKP21 dataset, ArgCMV reflects the messiness of actual human conversations, containing longer and co-referential arguments, more subjective language, and broader topical diversity. Our experiments show that existing methods fall short on our ArgCMV dataset, highlighting the need for stronger, more adaptable models for summarizing online debates. ArgCMV lays the groundwork for argument summarization and key point extraction in realistic, long-context settings like online discussions.

9 Limitations

We identify the following limitations in our work.

Limited number of arguments in certain topics: We find that many of our topics contain only a few arguments for each stance due to limited engagement on the Reddit thread. While including several topics helped us improve the diversity of our dataset, the presence of very few arguments can limit ArgCMV's effectiveness as a challenging benchmark. This issue can be addressed by collecting a larger pool of data and then filtering any low engagement threads.

Limited to discussions occurring in 2020: ArgCMV currently based on CMV discussions that occurred in the year 2020. Future work can include posts generated over a wider date range to expand our dataset and thereby enable research on temporal trends in online debate forums.

Data leakage during LLM pre-training: Given that contemporary LLMs are primarily trained on large-scale online data, there is a possibility that our dataset was part of the pre-training corpus of these LLMs which might affect the applicability of our results. This is an issue with any dataset based on publicly available online data. This can be potentially minimized by only collecting posts made after the model release dates.

Cost and reliability issues of LLMs: As our entire data labeling pipeline uses OpenAI's GPT models, the cost for the API usage might be significant for larger dataset sizes. Although we believe that this still does not offset the cost and effort involved in human annotation. Also, while we find LLM-generations to be mostly reliable based on our human-verification, we only validated a limited set which leaves room for some imperfections.

Ethical Considerations

We recognize that the use of naturally-occurring data generated as part of actual online discussions (here in the form of comments from Reddit) involves potential risks. We discussed these issues in detail before conducting this research and took the following measures to mitigate risks involved in the use of data from online communities like CMV. First, we collected data passively and post hoc, without any intervention, relying solely on naturally occurring discussions. Next, we actively worked to minimize potential risks to community members by not linking comments in ArgCMV back to their authors. We replaced all usernames with random strings and used these to compile dialogs, i.e., back-and-forth conversations between an OP and a replier, in our dataset. We also performed data cleaning to ensure any embedded URLs were removed. Additionally, we did not include any comments that were removed by moderators or deleted by their authors in this dataset, in an effort to respect moderators' and users' preferences respectively. Finally, we only collected public data through Reddit's official API, in an effort to protect Reddit itself from any harm. When releasing the data, we will perform a comprehensive set of checks to further ensure no personally identifiable data is released.

References

Milad Alshomary, Timon Gurcke, Shahbaz Syed, Philipp Heinrich, Maximilian Spliethöver, Philipp Cimiano, Martin Potthast, and Henning Wachsmuth. 2021. Key point analysis via contrastive learning and extractive argument summarization. arXiv preprint arXiv:2109.15086.

Moritz Altemeyer, Steffen Eger, Johannes Daxenberger, Tim Altendorf, Philipp Cimiano, and Benjamin Schiller. 2025. Argument summarization and its evaluation in the era of large language models. arXiv preprint arXiv:2503.00847.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020a. From arguments to key points: Towards automatic argument summarization. arXiv 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. arXiv preprint arXiv:2010.05369.

Sumit Bhatia, Prakhar Biyani, and Prasenjit Mitra. 2014. Summarizing online forum discussions–can dialog acts of individual messages help? In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 2127–2131.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. Booookscore: A systematic exploration of book-length summarization in the era of llms. arXiv preprint arXiv:2310.00785.

Liying Cheng, Lidong Bing, Ruidan He, Qian Yu, Yan Zhang, and Luo Si. 2022. Iam: a comprehensive and large-scale dataset for integrated argument mining tasks. arXiv preprint arXiv:2203.12257.

Frederick Choi, Tanvi Bajpai, Sowmya Pratipati, and Eshwar Chandrasekharan. 2023. Convex: A visual

conversation exploration system for discord moderators. Proceedings of the ACM on Human-Computer Interaction, 7(CSCW2):1–30.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70):1–53.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Charlie Egan, Advaith Siddharthan, and Adam Wyner. 2016. Summarising the points made in online political debates. In Proceedings of the 3rd Workshop on Argument Mining, The 54th Annual Meeting of the Association for Computational Linguistics, pages 134–143. Association for Computational Linguistics (ACL).

Roni Friedman, Lena Dankin, Yufang Hou, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. Overview of the 2021 key point analysis shared task. arXiv preprint arXiv:2110.10577.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 7805–7813.

Jia Guo, Liying Cheng, Wenxuan Zhang, Stanley Kok, Xin Li, and Lidong Bing. 2023. Aqe: argument quadruplet extraction via a quad-tagging augmented generative approach. arXiv preprint arXiv:2305.19902.

Christopher Hidey and Kathleen McKeown. 2018. Persuasive influence detection: The role of argument sequencing. In Proceedings of the AAAI conference on artificial intelligence, volume 32.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. arXiv preprint arXiv:2410.21276.

Jane Im, Sonali Tandon, Eshwar Chandrasekharan, Taylor Denby, and Eric Gilbert. 2020. Synthesized social signals: Computationally-derived social signals from account histories. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pages 1–12.

Jialun Aaron Jiang, Charles Kiene, Skyler Middler, Jed R Brubaker, and Casey Fiesler. 2019. Moderation challenges in voice-based online communities on discord. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW):1–23.

Manav Nitin Kapadnis, Sohan Patnaik, Siba Smarak Panigrahi, Varun Madhavan, and Abhilash Nandy. 2021. Team enigma at argmining-emnlp 2021: Leveraging pre-trained language models for key point matching. arXiv preprint arXiv:2110.12370.

Aman Kumar, Amit Shankar, Aviral Kumar Tiwari, and Hae-Jung Hong. 2023. Understanding dark side of online community engagement: an innovation resistance theory perspective. Information Systems and e-Business Management, pages 1–27.

Sung-Chul Lee, Jaeyoon Song, Eun-Young Ko, Seongho Park, Jihee Kim, and Juho Kim. 2020. Solutionchat: Real-time moderator support for chat-based structured discussion. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, page 1–12, New York, NY, USA. Association for Computing Machinery.

Hao Li, Viktor Schlegel, Riza Batista-Navarro, and Goran Nenadic. 2023. Do you hear the people sing? key point analysis via iterative clustering and abstractive summarisation. arXiv preprint arXiv:2305.16000.

Hao Li, Yuping Wu, Viktor Schlegel, Riza Batista-Navarro, Tharindu Madusanka, Iqra Zahid, Jiayan Zeng, Xiaochi Wang, Xinran He, Yizhi Li, et al. 2024a. Which side are you on? a multi-task dataset for end-to-end argument summarisation and evaluation. arXiv preprint arXiv:2406.03151.

Xiao Li, Yong Jiang, Shen Huang, Pengjun Xie, Gong Cheng, and Fei Huang. 2024b. Exploring key point analysis with pairwise generation and graph partitioning. arXiv preprint arXiv:2404.11384.

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

Nailia Mirzakhmedova, Johannes Kiesel, Khalid Al-Khatib, and Benno Stein. 2023. Unveiling the power of argument arrangement in online persuasive discussions. In 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), pages 15659–15671. Association for Computational Linguistics (ACL).

MistralAI and NVIDIA. 2024. Mistral-nemo-instruct-2407. https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407. Accessed: 2025-05-20.

Gaku Morio, Ryo Egawa, and Katsuhide Fujita. 2019. Revealing and predicting online persuasion strategy with elementary units. 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 6274–6279.

Karl Pearson. 1900. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 50(302):157–175.

Arturo Martínez Peguero and Taro Watanabe. 2024. Change my frame: Reframing in the wild in r/changemyview. arXiv preprint arXiv:2407.02637.

Allen Roush and Arvind Balaji. 2020. Debatesum: A large-scale argument mining and summarization dataset. arXiv preprint arXiv:2011.07251.

Allen Roush, Yusuf Shabazz, Arvind Balaji, Peter Zhang, Stefano Mezza, Markus Zhang, Sanjay Basu, Sriram Vishwanath, and Ravid Shwartz-Ziv. 2024. Opendebateevidence: A massive-scale argument mining and summarization dataset. Advances in Neural Information Processing Systems, 37:34270–34293.

Charlotte Schluger, Jonathan P Chang, Cristian Danescu-Niculescu-Mizil, and Karen Levy. 2022. Proactive moderation of online discussions: Existing practices and the potential for algorithmic support. Proceedings of the ACM on Human-Computer Interaction, 6(CSCW2):1–27.

Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. Anesthesia & analgesia, 126(5):1763–1768.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. arXiv preprint arXiv:2004.04696.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In Proceedings of the 25th international conference on world wide web, pages 613–624.

Liyan Tang, Zhaoyi Sun, Betina Idnay, Jordan G Nestor, Ali Soroush, Pierre A Elias, Ziyang Xu, Ying Ding, Greg Durrett, Justin F Rousseau, et al. 2023. Evaluating large language models on medical evidence summarization. NPJ digital medicine, 6(1):158.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. arXiv preprint arXiv:2408.00118.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerova, et al. 2023. Clinical text summarization: adapting large language models can outperform human experts. Research square, pages rs–3.

Anar Yeginbergen, Maite Oronoz, and Rodrigo Agerri. 2025. Dynamic knowledge integration for evidence-driven counter-argument generation with large language models. arXiv preprint arXiv:2503.05328.

Amy X Zhang and Justin Cranshaw. 2018. Making sense of group chat through collaborative tagging and summarization. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW):1–27.

Amy X Zhang, Lea Verou, and David Karger. 2017. Wikum: Bridging discussion forums and wikis using recursive summarization. In Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, pages 2082–2096.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024a. Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics, 12:39–57.

Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. 2024b. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. arXiv preprint arXiv:2403.02901.

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

A LLM Prompts

System Prompt for Key Point Generation

```
You are a professional debater and an expert at identifying concise, high-level reasoning patterns in extended argumentative discourse. You are given clusters of related arguments, where each cluster consists of multiple comments made by a **single user in a Reddit thread** responding in support or opposite to a debate topic. These comments are posted sequentially and may form a **logical progression** of thought or reasoning on the given topic and stance.

Your task is to extract a set of **salient, non-overlapping key points** that summarize the main lines of reasoning or sub-claims present in each cluster. Because the arguments within a cluster follow a logical flow, different parts of the cluster may correspond to different key points. A good key point captures a **distinct belief, rationale, or inference** made by the user that reflects a recurring or generalizable position on the topic. A key should should not exceed a length of {kp_token_length} tokens.

Each key point must:
- Stand on its own as a complete and clear claim
- Avoid restating or overlapping with other key points
- Capture reasoning shared across parts of the cluster, not isolated ideas

Here is an example of a good key point:
- "School uniform reduces bullying" is an opposing key point on the topic "We should abandon the use of school uniform."
```

User Prompt for Key Point Generation

```
Please generate a set of short (each ≤ {kp_token_length} tokens), salient, and non-overlapping stance key points on the topic "{topic}". Each cluster below contains a sequence of arguments made by a single user in a Reddit thread. These arguments are connected and built upon one another to form a coherent line of reasoning.

{clusters}

Instructions:
- For each cluster:
- Extract **multiple key points**, if the arguments contain more than one major idea or sub-claim.
- Do **not** include redundant or semantically overlapping key points.
- Do **not** force multiple key points if the cluster centers around a single idea.

Format: - Each key point should:
- Start on a new line
- Be preceded by a dash and a space ("- ")
- Be self-contained, with no references to the cluster or argument structure

Do not include any explanations or commentary. Return only the list of key points per cluster.
```

Prompt for Key Point Mapping

```
You are an expert debater and a professional at identifying concise, high-level, salient sentences called key points given an argument. You will be given argument related to a topic and stance. Additionally, you will be given a set of key points which were created by human experts for this topic and stance. Your task is to identify the key points that are present in the argument. A key point is considered present in the argument if the main idea is expressed clearly, even if reworded. Your output should be a **Python-style list** of the **indices of matching KPs**, e.g., [0, 2]
For example, the following argument and key points are given for the topic "We should ban the use of child actors" and opposing stance:

Argument: Banning child actors would ignore the fact that many children genuinely enjoy acting and choose to pursue it as a career. With appropriate regulation and adult supervision, they can work in safe environments where their well-being is prioritized. Moreover, child acting can offer early exposure to professional opportunities that build confidence, discipline, and creative skills. Instead of banning, we should focus on enforcing strict industry protections to prevent exploitation.

Key Points:
0 Child performers should not be banned as long as there is supervision/regulation.
1 Acting helps children build confidence and public speaking skills.
2 Child acting provides families with income opportunities.
3 Child actors have the right to choose their career.

Output: [0, 1, 3]

Given the argument and corresponding key points {stance} the topic "{topic}", identify the key points that are present in the argument. Carefully analyze each key point one by one and check if its contained in the argument. Your output should be a **Python-style list** of the **indices of matching KPs**, e.g., `[0, 2]`. Only output the list of matching KPs. Do not include any other text or explanation.

Argument: {argument}

Key Points:
{kps}

Output:
```