Designing Logic Pattern Templates for Counter-Argument Logical Structure Analysis

Shoichi Naito[*1,2,3], **Wenzhi Wang**[*1,2], **Paul Reisert**[4], **Naoya Inoue**[5,2], **Camélia Guerraoui**[1,2,6], **Kenshi Yamaguchi**[1], **Jungmin Choi**[2], **Irfan Robbani**[5], **Surawat Pothong**[5], **Kentaro Inui**[7,1,2],
[1]Tohoku University, [2]RIKEN, [3]Ricoh Company, Ltd.
[4]Beyond Reason, [5]JAIST, [6]INSA Lyon, [7]MBZUAI
shohichi.naitoh@jp.ricoh.com {wang.wenzhi.r7,guerraoui.camelia.kenza.q4}@dc.tohoku.ac.jp
beyond.reason.sp@gmail.com {naoya-i,robbaniirfan,spothong}@jaist.ac.jp
kenshi.yamaguchi.e7@tohoku.ac.jp jungmin.choi@riken.jp kentaro.inui@mbzuai.ac.ae

Abstract

Counterarguments (CAs) are a valuable tool in education for enhancing critical thinking skills. Despite their effectiveness, the logical attack structure of counterarguments in relation to their corresponding opponent argument remains unexplored due to its complexity. Towards tackling this challenge, in this work, we introduce **C**ounter-**A**rgument **L**ogical **S**tructure **A**nalysis (**CALSA**), a new task. We first define 10 new CA logic patterns, each comprised of a unique template and slots. We then conduct an annotation study on top of 778 CAs using our patterns to create a new dataset. Our dataset achieves high annotator agreement (Krippendorff α=0.50) and high coverage (86.5%). We perform preliminary experiments employing recent large language models to assess the feasibility of automating CA logical structure analysis. Our findings highlight the task's inherent complexity within a straightforward framework, indicating exciting opportunities for further exploration. We publicly release our dataset and model scripts at https://github.com/cl-tohoku/CALSA.

1 Introduction

Counterarguments (CAs) are employed in education as an effective means to improve critical thinking skills, especially for interactive discussions such as debates (Roy and Macchiette, 2005; Liu and Stapleton, 2014; Johnson and Johnson, 1993; Nussbaum, 2008). Constructing an effective CA not only requires the ability to comprehend the logical structure of an opponent's argument, but it also requires identifying where countering is most effective in order to construct a persuasive argument useful as an attack. Identifying vulnerabilities within an opponent's argument is a fundamental aspect of critical thinking, as affirmed by numerous educational studies, such as the well-regarded Watson-Glaser Critical Thinking Appraisal (Watson and Glaser, 1952).

In the context of argumentation-based education, feedback provided by teachers plays a crucial role in enabling learners to enhance their skills. Nonetheless, expecting all teachers to undertake this task is unrealistic, as it would require a significant amount of time to give tailored feedback to each student, and not all teachers undergo training in teaching argumentation (Driver et al., 2000). Developing a system that automatically gives feedback to argumentative texts would be highly beneficial, as it could assist teachers with the grading process. Envisaging the downstream task of providing informative feedback to learners on the CAs they produce, we focus on the design considerations for the task of CA logical structure analysis.

Various studies have explored the logical structure of arguments, with Argumentation Schemes (Walton et al., 2008) serving as a foundational framework for providing feedback on arguments (Macagno and Konstantinidou, 2013; Song and Ferretti, 2013; Song et al., 2014). Argumentation Schemes consists of 96 total schemes used to classify everyday discourse arguments into a claim and premise(s) structure. Each scheme is paired with critical questions that challenge the validity of the argument, essentially offering feedback. For instance, *Argument from Expert Opinion* is a scheme that concludes that proposition A is true based on the endorsement of an expert E. This scheme prompts critical questions such as, "Is E an expert in the field that A is in?" and "Is A consistent with what other experts assert?"

The notion of schemes typology proves advantageous in an educational context. Typology enables the generalization of individual cases, reducing cognitive load and allowing students to apply past experiences to new cases (Rosch, 1978). However, despite its efficacy, there is no typology tailored for CAs. By establishing a typology, we can provide

[*]Equal contribution.

Findings of the Association for Computational Linguistics: EACL 2024, pages 11313–11331

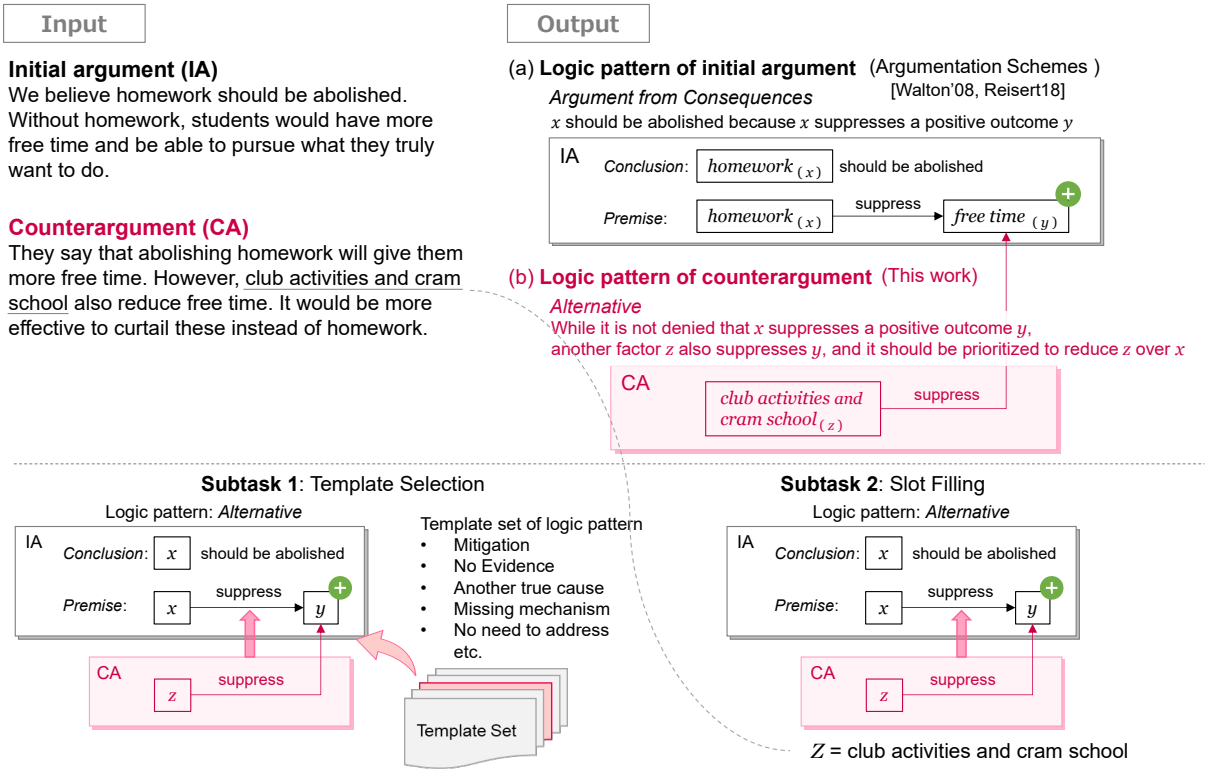


Figure 1: Our CA logical structure analysis task. Given a debate consisting of an initial argument (IA) and counterargument (CA), we i) select a pre-defined template to represent the CA's underlying logic in response to the IA and ii) fill in the template's slots (i.e., z written in red text).

feedback for CAs in a manner akin to the mechanism of Argumentation Schemes.

Figure 1 shows an example of an initial argument (henceforth, IA) and CA. In this specific example, the IA claims that *homework* suppresses *free time*, and although the CA agrees with this logic, it provides an alternative to indicate that *homework* is not the main issue regarding *free time*, but *club activities and cram school* are the more significant issues that suppress *free time*. This pattern of suggesting alternative solutions (i.e., reducing "club activities and cram school" as an alternative solution to the "free time" issue) to these IAs emerges irrespective of the topic. Once these logical patterns are identified, it becomes possible to provide informative feedback to the CAs, such as "Why is the alternative proposed by the CA superior?"

In computational argumentation, several existing research studies have focused on CA analysis. Bex and Reed (2011) formulated the Argument Interchange Format (AIF) core ontology to represent conflict and preference in argumentation. Afantenos and Asher (2014) demonstrate the use of Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides, 2003) to represent complex rhetorical moves, such as attacking the argument's claim (i.e., the statement expressing the position of the arguer) while agreeing with a premise (i.e., supporting statement for the conclusion). LPAttack (Mim et al., 2022) provides an annotation scheme on kinds of attacks, focusing on *Argument from Consequences* (Walton et al., 2008). However, no work has addressed the typology of logical patterns between a CA and its IA.

This work centers on CA logical structure analysis in a parliamentary debate setting. We first create a set of common logic patterns in CAs and construct a dataset on top of CAs. As the structure of logic patterns in a CA depends on the opponent's argument and encompasses diverse argument types, we consider defining patterns for CAs using *Argument from Consequences*, a frequently occurring argumentation scheme (Feng and Hirst, 2011; Reisert et al., 2018; Visser et al., 2022).

Our work consists of two research questions: (i) How can we create an inventory of logic patterns in CAs that offers sufficient coverage and is feasible for annotation? (ii) To what extent can a language model identify these logic patterns of CAs? Our main contributions are as follows:

- We introduce **C**ounter-**A**rgument **L**ogical **S**tructure **A**nalysis (**CALSA**), a new task for representing the logical structure between a CA and its corresponding IA.

- We develop 10 new, distinct logic patterns for CAs based on the *Argument from Consequences* (Walton et al., 2008) scheme and annotate our patterns on top of an existing dataset of 778 debates (i.e., IA-CA pairs).

- We design a cost-effective template selection phase for collecting high-quality templates with high IAA (Krippendorff's α=0.50) and high coverage (86.5%).

- We evaluate the feasibility of automating our task by applying several language models with few-shot and fine-tuning on the constructed dataset.

- We publicly release our dataset and model scripts to facilitate further research in CA logical structure analysis. [1]

2 Related work

2.1 Representation of attack relation

Several research studies exist on argumentation structure, particularly focusing on the representation of attacks. In *abstract argumentation frameworks* (Dung, 1995), arguments are represented as a directed graph with nodes and edges as attack relations, allowing for the analysis of the acceptability of arguments based on some criteria. *ASPIC*$^+$ (Modgil and Prakken, 2014) is a known means to generate an abstract argumentation framework. *ASPIC*$^+$, inspired by Pollock (1987)'s work, distinguish three kinds of abstract attack types: *rebuttal*, an attack on the conclusion of a defeasible inference, *undercut*, an attack on a defeasible inference step itself, and *undermine*, an attack on an ordinary premise. We go a step beyond these abstract types by understanding the more fine-grained reason for attack via our intricate logic templates.

In a study analyzing the discourse aspect of CAs, Bex and Reed (2011) formulated the Argument Interchange Format (AIF) ontology to represent conflict and preference in argumentation. Afantenos and Asher (2014) employed segmented discourse representation theory (SDRT) to describe what is both acknowledged and denied within a CA. LPAttack (Mim et al., 2022) focuses on *Argument from Consequences* and proposed an annotation scheme for defining how causality and value are attacked. While these works contributed to exploring CA structure representation, they did not focus on the typology of logical patterns between a CA and its IA.

Argumentation Schemes (Walton et al., 2008) is a fundamental work that focuses on logical structure representation of argumentation through schemes, where each scheme serves as a template for a common argument type. Each scheme focuses on a micro-structure (i.e., premise and conclusion) within a single argument, opposed to the relationship between two opposing arguments. We expand upon the fundamentals of Argumentation Schemes by representing the structure of an CA argument in relation to its corresponding IA.

2.2 Attack identification task

Research has been conducted in Argument Mining (Mochales and Moens, 2011; Peldszus and Stede, 2013; Stede and Schneider, 2018; Lawrence and Reed, 2019) to automatically analyze argument structures. In a study targeting monological arguments, Stab and Gurevych (2014, 2017) developed the Persuasive Essay Corpus to identify both support and attack relations among argument components. Peldszus (2014); Peldszus and Stede (2015) created a dataset annotating rebuttals and undercut relations for online micro-texts. Similar to Argumentation Schemes, these studies focus on identifying relations (i.e., *attack*) between the argument components within a single argument.

Previous studies have focused on dialogical arguments by introducing a task to determine whether two arguments are in (dis)agreement, utilizing data from online discussion forums (Murakami and Raymond, 2010; Walker et al., 2012; Boltužić and Šnajder, 2014; Sridhar et al., 2015; Rosenthal and McKeown, 2015), peer reviews (Cheng et al., 2020), and debate speeches (Menini et al., 2018; Orbach et al., 2020). Other studies have introduced a task to identify the conflict relation at a more fine-grained level in argument discourse units (Ghosh et al., 2014; Visser et al., 2019; Hidey et al., 2017). Such studies assist towards understanding which argumentative discourse units are being attacked, but they do not consider the details as to how and why the attacks are made.

[1] https://github.com/cl-tohoku/CALSA

3 Design of Logic Pattern Templates

As discussed in §1, we create common logic patterns of CAs. Following Argumentation Schemes (Walton et al., 2008; Reisert et al., 2018), our patterns are represented by templates and their placeholders (henceforth, slots). By observing actual CAs, we manually derive an inventory of templates.

3.1 Requirements

For creating an inventory of templates for representing the logical structure of a CA between its opposing IA, we establish the following key criteria:

First, we require that our logic templates are *comprehensive*, covering a wide-range of sentences in CAs. This is necessary to ensure that learners in a pedagogical setting receive appropriate feedback on a variety of different CAs. For this criteria, we observe the template coverage.

Second, our templates should be *expressive*, i.e., express logic details sufficiently, ensuring that feedback received by learners is specific and useful, especially for critical thinking skill improvement. We look to possible feedback comments for each template to ensure templates are expressive.

Third, we require that the templates are *simple*. This requirement is intended to simplify both human annotation and machine predictions. For testing this criteria, we observe the inner-annotator agreement for non-expert annotators.

Overall, creating a template set for all the requirements is challenging. For the *comprehensive* requirement, a generic template (e.g., the argument x claimed by IA lacks persuasiveness) could be applied to several CAs. However, our system then could not satisfy the *expressive* requirement, as a generic template would not allow for useful feedback due to a lack of capturing important details such as logic. On the other hand, templates that express the finer details of the logic could provide informative feedback, thus fulfilling our *expressive* requirement. However, the more similar templates with minor differences are considered, the more complex the annotation becomes, thus failing to meet our *simple* requirement. Given this criteria, we carefully craft our templates, aiming for reasonable annotator agreement and coverage.

3.2 Inducing Templates

With the goal of meeting our requirements, we manually design our template set. For analysis in template design, we use TYPIC (Naito et al., 2022). TYPIC includes a total of 1,000 CAs and 10 IAs, with 100 CAs per each IA. For some CAs, feedback comments provided by debate education experts are also included. TYPIC is ideal for our template creation, as it is suitable for deriving templates associated with feedback comments.

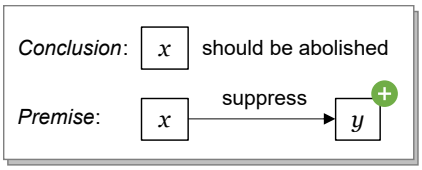


Figure 2: Argument from (Negative) Consequence scheme from Walton (Walton et al., 2008) which we utilize as a basis for our CA logical structure analysis. The template defined by Reisert et al. (2018) is used as a template for IA.

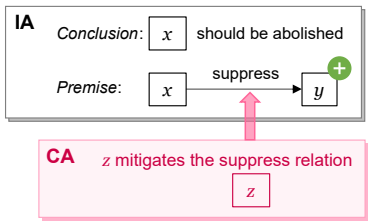
Observing TYPIC, we discover that the CA structure significantly varies depending on IA type. Therefore, we take the strategy of starting our exploration with a restricted range of IA types. Specifically, we utilize *Argument from Consequences* (Walton, 1999) for developing our templates. *Argument from Consequences* is a type of argument that concludes an action should or should not be brought about based on the good or bad consequences it may lead to. This scheme can be divided into Argument from Positive Consequences and Argument from Negative Consequences. The latter is defined as follows:

> **Premise**: If action x is brought about, bad consequences will plausibly occur.
> **Conclusion**: Therefore, x should not be brought about.

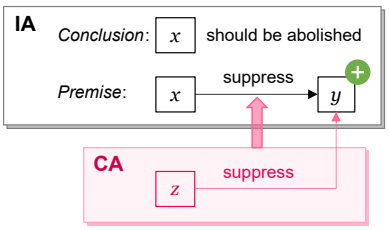
We selected this argument type initially for the following two reasons. First, it is observed that this type of argument is most common in discussions in which participants argue for or against a particular course of action (Walton, 1999; Feng and Hirst, 2011; Visser et al., 2019). Second, despite a limited range of argument types, it can still be highly beneficial in educational scenarios where students create CAs to a given prompt IA, as educators can control/choose the range of IA types to present as prompts.

To operationalize the Argument from Consequences scheme, allowing for a both annotation and model friendly, fine-grained logical representation, Reisert et al. (2018) created *argument templates*.

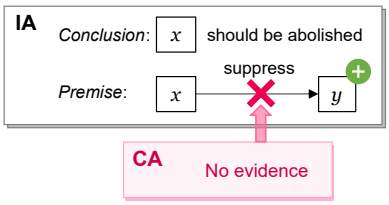
Mitigation (Mig)
While it is not denied that x suppresses a positive outcome y, the causal relationship can be mitigated through the means of z.



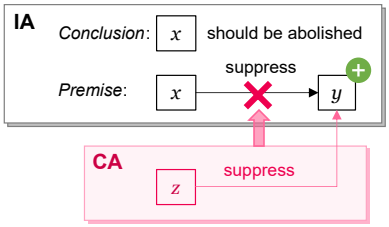
Alternative (Alt)
While it is not denied that x suppresses a positive outcome y, another factor z also suppresses y, and it should be prioritized to reduce z over x.



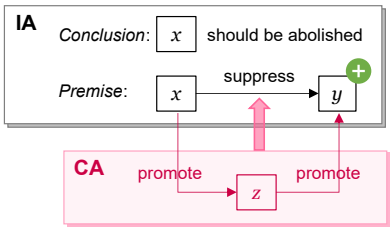
No evidence (No Evi)
There is no evidence that x suppresses y.



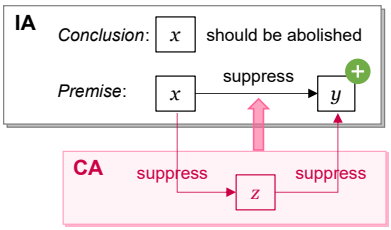
Another true cause (ATC)
The true cause of suppressing y is z, not x; therefore, abolishing x will not solve the problem.



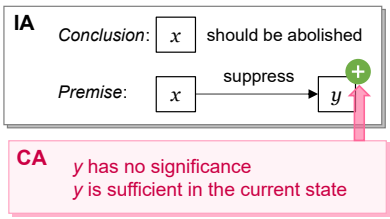
Missing mechanism #1 (MM1)
x promotes another factor z, and z promotes y. Therefore, x does not suppress y but rather promotes y.



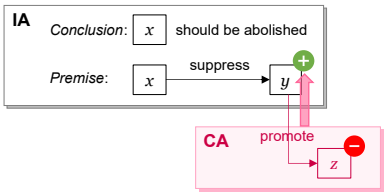
Missing mechanism #2 (MM2)
x suppresses another factor z, and z suppresses y. Therefore, x does not suppress y but rather promotes y.



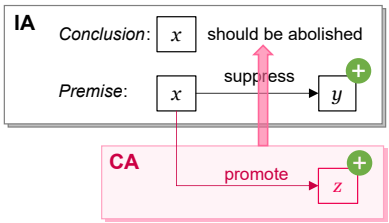
No need to address (NNA)
y is sufficient as it is, and there is no need to take action for y.



Negative effect due to y (Neg eff)
Since y leads to a negative outcome z, y is not a positive outcome.



Positive effects of a different perspective from y #1 (Dif Per1)
Since x promotes a positive outcome z, which is a different perspective from y, x should not be abolished.



Positive effects of a different perspective from y #2 (Dif Per2)
Since x suppresses a negative outcome z, which is a different perspective from y, x should not be abolished.

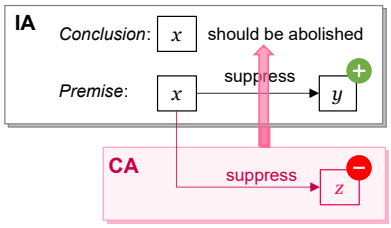


Table 1: Inventory of logic patterns we create for our new task of CA logical structure analysis. Our patterns were heavily inspired by *Argument from Consequences* scheme (Walton et al., 2008) and Reisert et al. (2018). The templates represent how a CA attacks the logic of the IA, shown in red. Red arrows indicate which part of the IA (causality of premise, value of premise, conclusion) is being attacked.

Consider the following argument template for Argument from Negative Consequences, as illustrated in Figure 2:

> **Premise:** x suppress y. AND pos(y).
> **Conclusion**: neg(x).

Above, x and y are *action* and *consequence* slots, respectively. neg(x) refers to the sentiment of x, and suppress refers to the relation between x and its triggering of consequence y. For creating our templates, we adopt similar ingredients, allowing for fine-grained CA logic representation.

In building an inventory of logic templates for CAs, our primary focus is on the practical application of providing feedback to CAs. Distinct templates are designed based on the principle that varying feedback indicates different templates. To this end, we investigate the CAs and the feedback provided to them in TYPIC, analyzing the factors that cause variations in the feedback. This analysis reveals that feedback tends to differ based on both the part being attacked (the attack point) and the manner of the attack. When attacking an argument of *Argument from Consequences*, the attack points can be broadly categorized into three types: attacking the causal relationship (the premise that x suppresses y in Table 1), attacking the value (the premise that y is good), and attacking the conclusion (that x should be abolished). In terms of the manner of attack, some attacks negate the opponent's claim, some acknowledge it but impose limitations, and others contradict the opponent's claim. These different approaches also lead to variations in feedback. Through the analysis of actual examples of CAs and feedback comments, we organize different types of attacks into an inventory of 10 templates, as shown in Table 1.[2]

3.3 Task Setting

As depicted in Figure 1, we formulate CA logical structure analysis as a sentence-level sequence labeling task consisting of two subtasks: i) *template selection* and ii) *slot-filler extraction*.

Template Selection Given an IA and its CA, each sentence in the CA has the ability to incorporate multiple logic templates. However, both time and annotation for exhaustively collecting all templates per each CA sentence is costly. In addition, it would be challenging to have annotators label multiple templates simultaneously per sentence. To mitigate these issues, we carefully design our annotation by temporarily treating template selection as single-template per CA sentence task, where a small number of non-expert annotators provide a single template per each sentence. Afterwards, we can collect high-competent templates (i.e., single template per CA sentence) while simultaneously creating a cost-effective method of collecting multiple templates by having disagreeing instances (i.e., multiple templates per CA sentence) verified by an expert annotator. Ultimately, this allows our task to be treated as multi-template per CA.

Slot-Filler Extraction The next step is to fill in the selected template slots. The template contains fillers (i.e., slot-fillers) that serve as a representation of key points for CAs. This task aims to extract phrases from the sentence that can be used as a template's slot-filler within the CA.

4 Annotation Study

We describe the creation of our dataset using the TYPIC debate corpus and crowdsourcing.

4.1 Dataset

We utilize TYPIC (Naito et al., 2022), a dataset consisting of a debate topic, an IA, and a CA annotated with feedback comments. To be consistent with classrooms, where teachers use the same topics annually, TYPIC focused on a *micro-domain* approach (i.e., many CAs per few topics). This allowed for the dataset to be used for feedback construction while mitigating task difficulty. Datasets like TYPIC are useful for implicit discourse recognition, a task that currently struggles with current LLMs (0.16 Macro F1) (Chan et al., 2024). For our task, we can cover a large range of typical CA logic patterns specific to a topic.

We additionally choose TYPIC for its reasonable size of IAs and paragraph-level CAs with multiple arguments, making the corpus appealing for our task of logic pattern analysis. In total, the corpus consists of two debate topics: *Ban death penalty (DP)* and *Ban homework (HW)*. We select 5 IAs and 490 CAs from the TYPIC corpus and extend upon it with a new topic: *Students should have a part-time job* with 3 new IAs and 288 new CAs, totaling 778 CAs (5,172 sentences).

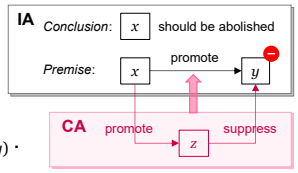
[2]Please see Appendix A for details on the relationship between our templates and Critical Question, Appendix B for examples of each pattern, and Appendix C for possible feedback comments for each pattern.

IA: Today's topic is "Homework should be abolished". Our first point is that forcing students to do homework will make them passive in character... Therefore, for these reasons, homework should be abolished.

CA: ... However, in fact, compared to completing assignments in the classroom, homework is ideal for requiring students to work on their own initiative...By requiring students to work without the physical presence and real-time help of the teacher, homework promotes independence and self-discipline....

Interpretation 1: Missing Mechanism #1

homework (x), on the contrary, suppresses passive in character (y) because it promotes own initiative (z), which suppresses passive in character (y).



Interpretation 2: Positive effects of different perspective from y #1

homework (x), should not be abolished because it promotes independence and self-discipline (z), which are positive outcomes from a perspective different than passive in character (y).

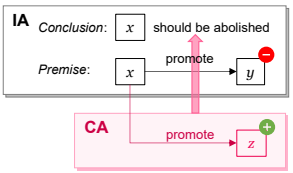


Figure 3: Examples of multiple reasonable interpretations possible.

4.2 Annotation Process

For each IA and CA pair, our annotation process is as follows. First, annotators carefully read both IA and CA. Next, annotators select all relevant templates per sentence in the CA, choosing "Not Applicable" if a template cannot be selected. Finally, annotators label the corresponding segments (i.e., consecutive sentences with same template label) for each chosen template, extracting phrase(s) from the CA that match the template slot-filler(s). [3]

Given the complexity of our task, we implemented thorough procedures to eliminate simple errors during the annotation process. For annotations performed by crowdworkers, we selected seven reliable workers through Amazon Mechanical Turk [4]. We established communication channels with them to address any uncertainties or questions as they arose. After reading the annotation guideline, we conducted three preliminary calibration sessions to align our understanding. In these sessions, all workers annotated several examples, after which we showed the gold labels and opened a phase to discuss questions and doubts. Any inquiries and suggestions for improvement raised during these calibrations were promptly reflected in the annotation guidelines. The main annotation was conducted by 3 non-expert annotators per each of the 778 CA instances, averaging 5-10 minutes per CA.[5]

4.3 Annotation Results

To evaluate annotation feasibility, we measure inter-annotator agreement (IAA) and the coverage of the template set. We discuss the results below.

Quality of Annotation For collecting high-quality single template per CA sentence instances and multiple templates per CA sentence, we first aggregate template labels using MACE (Hovy et al., 2013). For all 778 CAs, we filter by MACE using a threshold of 0.8, resulting in 469 CAs where each sentence has a gold label. We calculate IAA for these instances and achieve a high Krippendorff's α (Krippendorff, 1980) of 0.50, comparable to pioneering works in argumentation (Wachsmuth et al., 2017; Miller et al., 2019). Furthermore, Heinisch et al. (2023) shows that IAA is often reported to be fair to moderate in annotations of argumentative tasks that include subjectivity. To further test the quality, experts sampled 50 of the 469 CAs and agreed with 88.0% of the selected templates.

Next, we collect high-quality, multi-template labels per CA sentence. We gather sentences with a missing MACE gold label from the remaining 309 CAs and have non-expert labels verified by an expert (i.e., expert manual correction) [6]. If the expert did not agree with any of the labels, the expert annotated with a template from the full inventory. In total, 872 sentences were verified by the expert, and 844 sentences had at least one agreement with non-experts. This indicates that non-experts were significantly able to perform the task.

After verification, 50 CAs from the expert's annotation were verified by another expert annotator, resulting in 78.5% of CAs being agreed upon.[7] In total, 134 CAs contained at least one sentence with multiple templates. The remaining 175 CA instances were combined with the high-

[3] Please see Appendix D for an annotated example.
[4] https://www.mturk.com/
[5] Please see Appendix E for annotation cost details.
[6] Experts reviewed instances in the bottom 20% of entropy for aggregated label probabilities calculated using MACE.
[7] Since multiple interpretations were deemed valid and the gold standard is multi-template per CA, Krippendorff's alpha could not be applied.

competent 469 CAs, totaling 644 CAs with only single-template labels per sentence. In total, we achieve 4,264 high-quality single-template CA sentences and 709 multi-template sentences.

We conduct a small, in-house study to observe the quality of annotator slot-fillers. Two expert annotators reviewed 43 slot-fillers across 16 IA/CA pair instances. After discussing the disagreements, both agreed that 38 of the 43 slot-fillers were correct. Disagreeing instances included those requiring further context surrounding the slot-filler in the original text. As a result, only 2 of the 43 slot-fillers were agreed to be incorrect.

Coverage of Template Set We calculate the coverage of the template set and determine that 87.5% of sentences in our CAs contained a template besides "Not Applicable". The majority of CAs can be represented within our template set.[8]

4.4 Disagreement Analysis

Figure 3 shows an example of two disagreeing annotations which were labeled as correct by our expert annotator. Here, the logic can be interpreted as both "Missing Mechanism" and "Positive Effects of a Different Perspective from y". Interpretation 1 suggests that by promoting "their own initiative", homework actually suppresses "passive in character", contrary to the IA's claim that it promotes passivity. Interpretation 2 suggests that homework offers other benefits, such as promoting "independence and self-discipline", which are distinct from the IA's perspective of promoting passivity. This disagreement arises depending on whether the IA's "passive in character" and the CA's "independence and self-discipline" are seen as opposing concepts (Interpretation 1) or as distinct concepts (Interpretation 2). These two patterns differ in the strength of their CAs (with Missing Mechanism being a stronger counterargument), and thus should be distinguished.

5 Model Experiments

We aim to investigate how well current LLMs can perform on our dataset. For our target models, we select a few recent, high-performance open LLMs based on Huggingface's Open LLM leaderboard [9], and OpenAI's GPT family models.

[8] Please see Appendix F for template distribution.

[9] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

5.1 Automatic CA logical structure analysis

Settings We employ fine-tuning and few-shot learning for training generative models to perform *1) template selection* and *2) slot-filler extraction* together in a general text2text manner. The format of the model's input and output is shown in Figure 4. Regarding the dataset in use, we divide CA instances that have single-template annotations into both *train* and *dev* sets, where *train* consists of 516 CAs and *dev* consists of 128 CAs. For testing the models, we utilize all 134 CAs with multi-template annotations (henceforth, *test*), where each sentence in a CA may have one or more gold template(s). We train our models until training loss converges and evaluate each checkpoint on *dev*. We select the checkpoint with the highest evaluation score for *test*. Regarding few-shot learning, we randomly select 20 CAs (20-shot) for the same IA as the target CA from *train* for in-context examples, for each CA in *test*. To ensure robust evaluation, we conduct all experiments 3 times. We conduct the fine-tuning experiments with 3 different seeds.

Evaluation We evaluate the task of *template selection* with sentence-level F1 score. Thus, for each sentence in a CA, we have LLMs predict one template and consider it correct if it is within the gold label(s) for that sentence. For our task of *slot-filler extraction*, we evaluate it using the precision score. We utilize RougeL to evaluate the lexical overlap between the generated slot and the label. We first consider a slot incorrect when either 1) the predicted logic pattern template for which the slot is generated is incorrect since slots are dependent on logic pattern templates, or 2) the generated slot cannot be found in the counter-argument passage. If none of the above applies, we then determine whether a generated slot is correct by comparing the RougeL score between the target slot and each slot in the gold list. If the RougeL score is higher than a pre-defined threshold, we consider the target slot correct. For our experiments, we set the threshold to be 0.5.

Results As shown in Table 2, scores are low for both *template selection* and *slot-filler extraction*, even when evaluating models' generation with multi-template annotations (i.e. the generation is considered correct if it matches any one of the multi-template annotations). Specifically, precision for ***slot-filler extraction*** is exceedingly low. We attribute this to our strict evaluation method, which

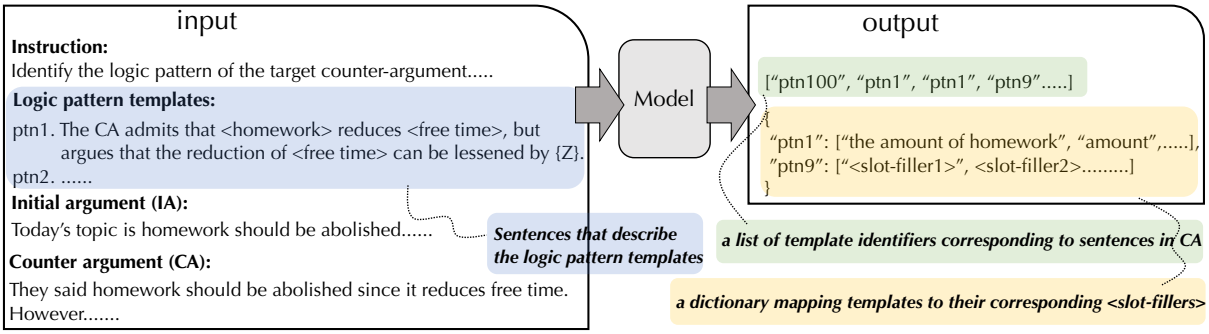


Figure 4: The input/output format of our model experiments.

models	F1 (templates)	Precision (slots)
GPT-4	0.549 (0.018)	0.153 (0.012)
GPT-4-turbo	0.545 (0.012)	0.147 (0.011)
GPT-3.5-turbo	0.604 (0.045)	0.204 (0.040)
Mixtral8×7b	0.522 (0.030)	0.092 (0.004)
Yi-34b	0.564 (0.015)	0.095 (0.005)
Llama-2-13b	0.491 (0.023)	0.047 (0.013)

Table 2: Averaged F1 and precision scores over results of three runs for template selection (*templates*) and slot-filler extraction (*slots*), respectively. The results above the line are for few-shot learning experiments, while those below the line are for fine-tuning experiments. The numbers in the parenthesis indicate the standard deviation of three runs.

solely rewards lexical overlaps without factoring in semantic information, as the slot-fillers are intended to be extracted directly from the original text. Nevertheless, the results indicate that our proposed task is significantly challenging for current LLMs to address using a general text2text setting.

5.2 Error Analysis

To investigate whether models' predicted templates are wrong or correct but not included in our gold labels, we randomly sample 30 instances, with a template selection F1 score below 0.5 to increase the possibility that sampled instances have predictions that are not included in the gold labels (henceforth, out-of-label predictions), per each of the 4 models. We observe that, on average, 71% of out-of-label predictions are incorrect.[10] The investigation results indicate that it is indeed challenging for the current LLMs to identify our logic templates. We believe the model's poor performance is due to the complex nature of our task which heavily involves underlying, implicit reasoning.

[10]Please see Appendix H for specific detail.

6 Conclusion

In this work, we explored the new task of CA logical structure analysis. We first defined a new task setting and created a list of templates based on a popular Argumentation Scheme (Walton et al., 2008). We then conducted an annotation study to categorize 778 CAs with our templates and created a new dataset, achieving remarkable IAA and coverage. Finally, we investigated the feasibility of utilizing LLMs to automatically identify templates and their corresponding slot-fillers. We observed that it is challenging to automate our CA logical structure analysis task with current LLMs utilizing a general text2text generation paradigm.

In our future work, we aim to create a feedback template set for our logic pattern templates. We also plan to extend our logic pattern template set to include other Argumentation Schemes. Finally, we plan to explore better modeling methods for our task, such as decomposing it into more fine-grained sub-tasks.

7 Limitations

Pattern creation For our task of CA logical structure analysis, we create our patterns based on the argumentation scheme *Argument from Consequences* (Walton et al., 2008) due to its frequency of usage in texts (Reisert et al., 2018). However, there are more than 60 argumentation schemes, so there is still room to explore other common argumentation scheme logic patterns.

Data Our work is limited to 3 topics for our experiments. Ideally, we would like to explore a wider range of topics. However, the cost of annotation prohibits us from currently expanding our corpus even further. Furthermore, while the progression of LLMs has improved performance in many tasks, implicit discourse recognition still re-

mains a challenge. Chan et al. (2024) reported that in-context learning with LLMs achieved a Macro-F1 score of 0.16 for identifying implicit discourse relations in Penn Discourse Tree Bank. Opposed to collecting counterarguments on various topics in a broad and shallow manner, we aim to focus on a specific topic with a large amount of counterarguments. Such datasets collected with this approach are scarce and serve as valuable resources to assess the current limits of LLMs, as such datasets make it possible to cover typical CA logic patterns specific to each topic, which simplifies the problem by reducing cases that a model has never encountered before. Therefore, for the purpose of CA logical structure analysis, we require a dataset with a small number of topics with a large amount of CAs. Additionally, when considering actual usage scenarios, often only a few topics are immediately useful. For example, in a high school classroom, three topics may be sufficient. There is no need to prepare different topics for each class, nor is it necessary to change them every year.

Prompt Formatting for Model Experiments In this work, we experiment with only one type of input and output format for fine-tuning our model and few-shot learning. Thus, our work is limited in that there are more input and output formats that we can explore.

8 Ethical Statement

Crowdsourcing In this work, we conduct crowdsourcing experiments. For these experiments, the reward for annotating one CA was $4.00. The work time per annotation was about 10 minutes, and workers were paid more than the minimum wage. Additionally, bonuses were given for their hard work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 22H00524.

References

Stergos Afantenos and Nicholas Asher. 2014. Counter-argumentation and discourse: A case study. *CEUR Workshop Proceedings*, 1341.

N. Asher and A. Lascarides. 2003. *Logics of Conversation*. Studies in Natural Language Processing. Cambridge University Press.

Floris Bex and Chris Reed. 2011. Schemes of inference, conflict and preference in a computational model of argument. *Studies in Logic, Grammar and Rhetoric*, 36.

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.

Chunkit Chan, Cheng Jiayang, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2024. Exploring the potential of ChatGPT on sentence level relations: A focus on temporal, causal, and discourse relations. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 684–721, St. Julian's, Malta. Association for Computational Linguistics.

Liying Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. APE: Argument pair extraction from peer review and rebuttal via multi-task learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7000–7011, Online. Association for Computational Linguistics.

Rosalind Driver, Paul Newton, and Jonathan Osborne. 2000. Establishing the norms of scientific argumentation in classrooms. *Science Education*, 84:1–312.

Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357.

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.

Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 39–48, Baltimore, Maryland. Association for Computational Linguistics.

Philipp Heinisch, Matthias Orlikowski, Julia Romberg, and Philipp Cimiano. 2023. Architectural sweet spots for modeling human label variation by the example of argument quality: It's best to relate perspectives! *arXiv preprint arXiv:2311.03153*.

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.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

David W. Johnson and Roger T. Johnson. 1993. Creative and critical thinking through academic controversy. *American Behavioral Scientist*, 37(1):40–53.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Methodology*. Sage Publications, Inc., Beverly Hills, CA.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Fulan Liu and Paul Stapleton. 2014. Counterargumentation and the cultivation of critical thinking in argumentative writing: Investigating washback from a high-stakes test. *System*, 45:117–128.

Fabrizio Macagno and Aikaterini Konstantinidou. 2013. What students' arguments can tell us: Using argumentation schemes in science education. *Argumentation*, 27:225–243.

Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2018. Never retreat, never retract: Argumentation analysis for political speeches. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Tristan Miller, Maria Sukhareva, and Iryna Gurevych. 2019. A streamlined method for sourcing discourse-level argumentation annotations from the crowd. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1790–1796.

Farjana Sultana Mim, Naoya Inoue, Shoichi Naito, Keshav Singh, and Kentaro Inui. 2022. LPAttack: A feasible annotation scheme for capturing logic pattern of attacks in arguments. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2446–2459, Marseille, France. European Language Resources Association.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Sanjay Modgil and Henry Prakken. 2014. The aspic + framework for structured argumentation: A tutorial. *Argument & Computation*, 5.

Akiko Murakami and Rudy Raymond. 2010. Support or oppose? classifying positions in online debates from reply activities and opinion expressions. In *Coling 2010: Posters*, pages 869–875, Beijing, China. Coling 2010 Organizing Committee.

Shoichi Naito, Shintaro Sawada, Chihiro Nakagawa, Naoya Inoue, Kenshi Yamaguchi, Iori Shimizu, Farjana Sultana Mim, Keshav Singh, and Kentaro Inui. 2022. TYPIC: A corpus of template-based diagnostic comments on argumentation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5916–5928, Marseille, France. European Language Resources Association.

Michael Nussbaum. 2008. Using argumentation vee diagrams (avds) for promoting argument-counterargument integration in reflective writing. *Journal Of Educational Psychology*, 100:549–565.

Matan Orbach, Yonatan Bilu, Assaf Toledo, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2020. Out of the echo chamber: Detecting countering debate speeches. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7073–7086, Online. Association for Computational Linguistics.

Andreas Peldszus. 2014. Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 88–97, Baltimore, Maryland. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *Int. J. Cogn. Informatics Nat. Intell.*, 7:1–31.

Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pages 801–815.

John L. Pollock. 1987. Defeasible reasoning. *Cognitive Science*, 11(4):481–518.

Paul Reisert, Naoya Inoue, Tatsuki Kuribayashi, and Kentaro Inui. 2018. Feasible annotation scheme for capturing policy argument reasoning using argument templates. In *Proceedings of the 5th Workshop on Argument Mining*, pages 79–89, Brussels, Belgium. Association for Computational Linguistics.

Eleanor Rosch. 1978. Principles of categorization. In Eleanor Rosch and B. B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Erlbaum, Hillsdale, NJ.

Sara Rosenthal and Kathy McKeown. 2015. I couldn't agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 168–177, Prague, Czech Republic. Association for Computational Linguistics.

Abhijit Roy and Bart Macchiette. 2005. Debating the issues: A tool for augmenting critical thinking skills of marketing students. *Journal of Marketing Education*, 27(3):264–276.

Yi Song and Ralph P. Ferretti. 2013. Teaching critical questions about argumentation through the revising process: effects of strategy instruction on college students' argumentative essays. *Reading and Writing*, 26:67–90.

Yi Song, Michael Heilman, Beata Beigman Klebanov, and Paul Deane. 2014. Applying argumentation schemes for essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 69–78, Baltimore, Maryland. Association for Computational Linguistics.

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 116–125, Beijing, China. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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*. Morgan & Claypool, San Rafael, CA, USA.

Jacky Visser, John Lawrence, Chris Reed, Jean Wagemans, and Douglas Walton. 2022. *Annotating Argument Schemes*, pages 101–139. Springer Nature Switzerland, Cham.

Jacobus H Visser, Barbara Konat, Rory Duthie, Marcin Koszowy, Katarzyna Budzynska, and Chris Reed. 2019. Argumentation in the 2016 us presidential elections: annotated corpora of television debates and social media reaction. *Language Resources and Evaluation*, 54:123 – 154.

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

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

Douglas Walton. 1999. Historical origins of argumentum ad consequentiam. *Argumentation*, 13:251–264.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

Goodwin Watson and Edward M. Glaser. 1952. Watson-glaser critical thinking appraisal: Manual. new york: Harcourt.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Argument from Consequence Critical Questions

Below is the list of critical questions (CQ) for the argumentation scheme, Artgument from Consequences:

- CQ1: How strong is the likelihood that the cited consequences will (may, must) occur?
- CQ2: What evidence supports the claim that the cited consequences will (may, must) occur, and is it sufficient to support the strength of the claim adequately?
- CQ3: Are there other opposite consequences (bad as opposed to good, for example) that should be taken into account?

Our CA patterns can be broadly categorized into attacks on causality, value, and conclusion. Patterns that attack causality (6 patterns from Mig to MM2) correspond to the subdivisions of CQ1 and CQ2 of "Argument from Consequences". Patterns that attack the conclusion (Dif Per1 and Dif Per2) correspond to subdivisions of CQ3. While there is no CQ that corresponds to attacks on value (NNA and Neg eff), these patterns address the goodness or badness of the outcomes.

B Example of CA Pattern

Table 3 shows examples of each CA pattern.

C Feedback Comments

Table 4 shows potential feedback comments for each CA pattern.

D Annotation Example

Figure 5 shows an example of the annotation.

E Annotations Costs

The reward for crowd workers was set to $4 per annotation, ensuring an hourly wage of $24, which exceeds US minimum wage. Expert annotations were conducted by the authors, so no wages were incurred for that. Since we allocated three crowd workers for 778 data instances, the total cost was approximately $10,000.

F Dataset Statistics

Table 5 shows the number of CA sentences for each IA. Table 6 shows the template distribution of the dataset measured by the number of CA sentences. Note that since the *test* set has multi-template sentences, the number of total templates is larger than the number of CA sentences. Non-arg has the largest number of sentences since every CA essay begins with at least one Non-argumentative sentence.

G Addtional information on modeling experiments

We utilized LLaMA-Factory (Zheng et al., 2024) for fine-tuning the Open LLMs. We utilize LoRA and 4-bit quantization techniques to fine-tune large models due to the GPU resource limit. Table 7 shows the hyperparameters used in our experiments. For few-shot learning, we aim to evaluate the models in their best condition. Therefore, we provided the maximum number of examples allowed within the context length limitation of GPT-4 and GPT-4-turbo, which is 20.

H Details of Error Analysis of Modeling

Table 8 shows the percentage of incorrect predictions in out-of-label predictions in the sampled instances for all the models we investigated. For the task of slot-filling, we observed that at least one slot-filler was reasonable for all the correct out-of-label predictions. However, the generated slot-fillers tend to be too lengthy to include not only the target phrase but also the surrounding context (sometimes including a whole sentence).

Furthermore, since our task encompasses a wide range of subtasks, from fundamental NLP tasks such as coreference resolution to high-level reasoning tasks that utilize context information and background knowledge, it is not obvious to pinpoint specific aspects responsible for the model's incorrect predictions and categorize them into a fixed set of error patterns. However, for the sake of discussion, We show an example that demonstrates the need for various abilities to infer the correct template. Figure 6 illustrates a CA essay where the fourth sentence explicitly states that 'it (homework) is beneficial to promoting students' understanding (of the class)', which aligns with the template Dif per1 (i.e., homework promotes a beneficial outcome distinct from free time). This identification may seem obvious to humans, as we implicitly link the pronoun 'it' in the sentence to the actual subject 'homework', and we are also able to infer that 'student's understanding' pertains to 'the class', and

CA template	Example
Mitigation	IA: Homework should be abolished because it would give students more free time. CA: Most teachers will extend deadlines or reduce the amount of homework if you explain your situation.
Alternative	IA: Homework should be abolished because it would give students more free time. CA: Rather than abolishing homework, we should reduce the time spent on club activities and hanging out with friends.
No evidence	IA: School uniforms should be abolished because they suppress individuality. CA: There is no evidence that uniforms suppress individuality.
Another true cause	IA: Homework should be abolished because it would give students more free time. CA: Even if homework is abolished, it won't lead to more free time. In the end, students will still be made to study by their parents, so nothing will really change.
Missing mechanism #1	IA: The death penalty should be abolished because it denies criminals the opportunity for rehabilitation. CA: The death penalty is actually appropriate for the rehabilitation of criminals. Facing death forces them to confront the severity of their crimes.
Missing mechanism #2	IA: Homework should be abolished because it would give students more free time. CA: Without homework, students have to figure out what they need to learn on their own. This takes more time and, in the end, actually leaves them with less free time.
No need to address	IA: The death penalty should be abolished because it denies criminals the opportunity for rehabilitation. CA: Violent offenders sentenced to death are unlikely to be rehabilitated, so it makes no sense to give them that opportunity.
Negative effect due to y	IA: Homework should be abolished because it would give students more free time. CA: With more free time, students are more likely to go out and play, which can lead to involvement in misconduct and dangerous activities.
Positive effects of a different perspective from y #1	IA: Homework should be abolished because it would give students more free time. CA: Homework should not be abolished, as that would lead to a decline in academic performance.
Positive effects of a different perspective from y #2	IA: The death penalty should be abolished because it denies criminals the opportunity for rehabilitation. CA: Abolishing the death penalty and replacing it with life imprisonment would increase the costs of keeping criminals incarcerated.

Table 3: The examples of each CA pattern.

CA template	Feedback comments
Mitigation	• Is z an appropriate way to mitigate the causal relationship between x and y? • If z mitigates but does not completely eliminate the effects, can the remaining impacts be ignored?
Alternative	• Is the reason why z suppresses y explained? • If both x and z suppresses y, why should z be prioritized for reduction over x?
No evidence	• To consider counterarguments from various perspectives, let's also explore patterns other than "No evidence".
Another true cause	• Is the reason why x does not suppress y explained? • Is the reason why z suppresses y explained?
Missing mechanism #1	• Is the reason why x promotes z explained? • Is the reason why z promotes y explained? • Is the reason explained why x promotes y through z to a greater extent than it suppresses y?
Missing mechanism #2	• Is the reason why x suppresses z explained? • Is the reason why z suppresses y explained? • Is the reason explained why x promotes y through z to a greater extent than it suppresses y?
No need to address	• Can the negative impact of taking no action be ignored?
Negative effect due to y	• Is the reason why y causes the result z explained? • Why is disadvantage of z more important than the advantage of y?
Positive effects of a different perspective from y #1	• Is there any direct attack on the IA stating that x suppresses y or that y is a good outcome? • Why is the advantage of promoting z more important than disadvantage of suppressing y?
Positive effects of a different perspective from y #2	• Is there any direct attack on the IA stating that x suppresses y or that y is a good outcome? • Why is the advantage of suppressing z more important than disadvantage of suppressing y?

Table 4: Potential feedback comments for each CA pattern.

IA ID	main point of IA	#sentences of CA
HW1	Abolishing homework gives students more free time	696
HW2	Homework makes students passive in character	690
HW3	Students memorizing the incorrect way to study with homework	749
DP1	Abolishing death penalty prevents misjudgement	622
DP2	Abolishing death penalty relieves executioners' stress	728
PJ1	Part-time job helps students learn responsibility and manners	572
PJ2	Part-time job helps improve academic performance	533
PJ3	Part-time job helps students recognize the importance of money	582
Total		5,172

Table 5: The number of sentences for CAs associated with each IA.

Temp	*train*	*dev*	*test*	all
Non-arg	665	153	161	979
Others	515	88	132	635
Mig	489	128	119	736
Alt	268	72	70	410
No Evi	88	30	11	129
ATC	157	21	101	279
MM1	154	71	111	336
MM2	26	10	8	44
NNA	317	113	247	677
Neg eff	112	29	87	228
Dif Per1	334	51	221	606
Dif Per2	308	65	240	613

Table 6: *train*, *dev*, and *test* template distribution measured by the number of CA sentences. The template distributions of the *train*, *dev*, and *test* data are different since we split the data at the CA essay level to ensure all sentences from the same CA belong to the same split. One can re-split the dataset in a sentence level for further experiments in the future.

Parameter	Value
lora_target	q_proj, v_proj
lora_alpha	16
lora_rank	8
learning_rate	5e-5
learning_schedule_type	cosine
train_batch_size	8
gradient_accumulation_steps	8

Table 7: Hyperparameters used in the fine-tuning modeling experiments for all models.

"understanding the class" is a good outcome, drawing on context information and our commonsense knowledge. However, for a model to identify the underlying template, it must possess the ability to implicitly perform the tasks mentioned above: 1) connecting "it" to "homework" (i.e., coreference resolution), 2) relating "students' understanding" to "the class", and 3) recognizing that "understanding of the class" is a positive outcome based on its own knowledge obtained both from pertaining and fine-tuning phases or the context information obtained from the essay, in a sense that it is not directly prompted to generate or predict the answers to those tasks. We believe that performing all of these tasks simultaneously remains highly challenging for current LLMs, at least within the context of argumentation, based on our experimental results. Furthermore, the results also highlight the intrinsic value of our proposed task and dataset. It not only possesses societal and educational value, as discussed in the introduction, but also can be considered as a benchmark for testing LLMs' ability to comprehend argumentative text at a deep level.

Model	% of incorrect predictions
GPT-4	66% (114 / 172)
GPT-4-turbo	65% (90 / 139)
Mixtral8×7b	72% (118 / 164)
Yi-34b	79% (130 / 165)

Table 8: Percentage of incorrect predictions in out-of-label predictions from the sampled instances (rounded off). The actual number of incorrect predictions (sentences) and out-of-label predictions (sentences) are shown in the parenthesis.

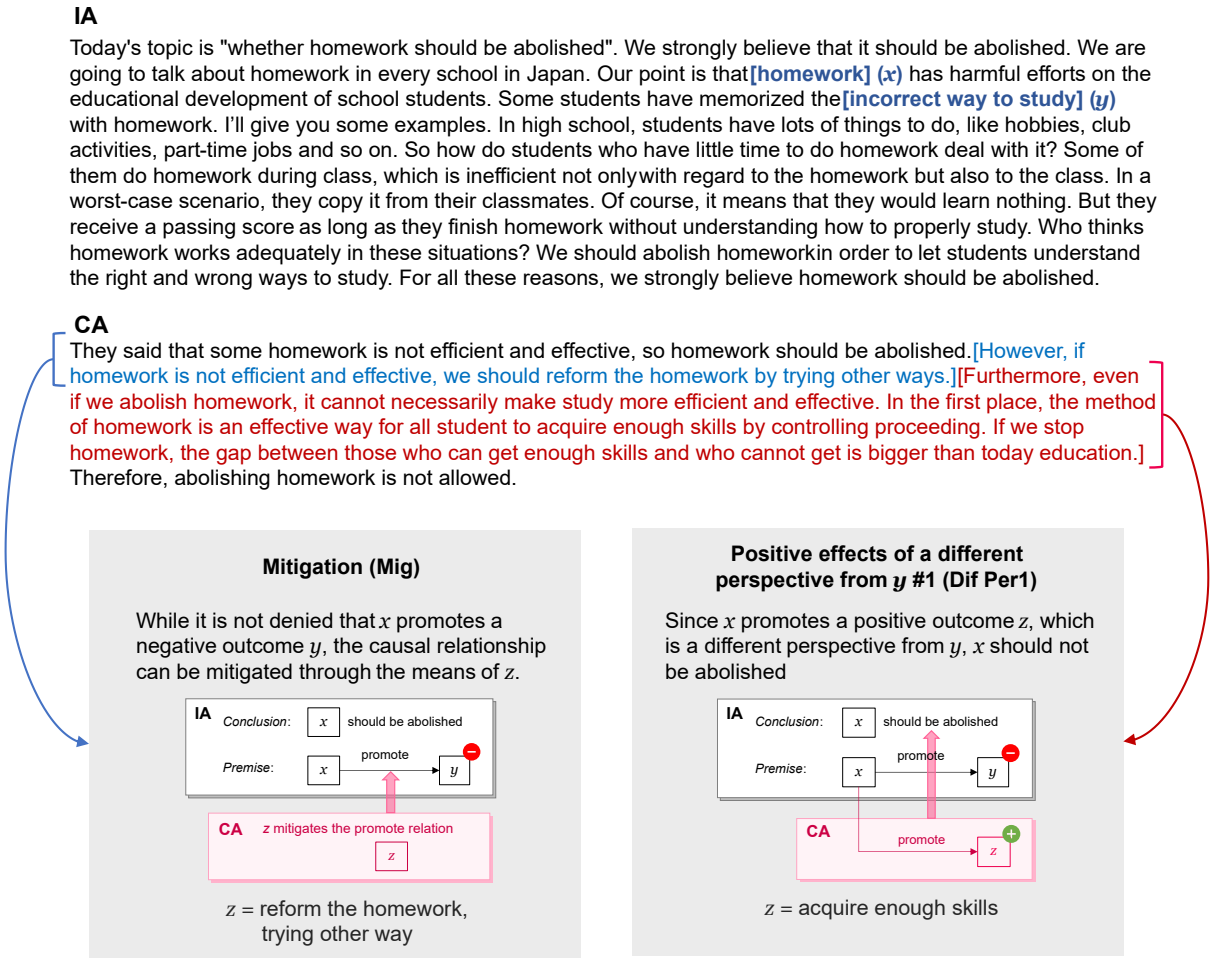


Figure 5: Annotation example of logic pattern templates and slot-fillers

I An example of the prompt used in few-shot learning experiments

Table 9 shows one example prompt we used for few-shot learning experiments.

IA:

Today's topic is "Homework should be abolished". The point is "free time". We believe that if **[homework] (*x*)** were to be abolished, we could have more **[free time] (*y*)**. As a result, we could do more of what we really wanted like club activities, hobbies, or playing with friends. In my case, I go to tennis club after class until 5:00 pm and then I go to cram school until 8:00 pm. After this full day, I arrive at my home around 8:40 pm to eat dinner and take a shower. At nearly 10:00 pm I start my homework. I have a lot of homework. As a result, I go to bed late at night at nearly 1:00 am in the morning and I don't have the opportunity to sleep for a long period of time. It is not healthy. Therefore, homework should be abolished.

CA:

However, it is rather counter productive, because it can cause the decline of students' study ability.
For example, teachers put on preparation for next class as homework.
Through homework, students can know substances of next class in advance.
It is so beneficial to promote students' understanding.
However, if homework were to be abolished, teachers cannot do it.
Still time is limited.
Thus, each class are not so meaningful.
Only students who are good at study can understand class's substance.
To sum up, their study ability will decline.

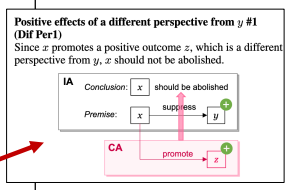


Figure 6: An example of CA with template Diff Per1. It requires various abilities to identify the correct template.

Identify the logic pattern of the target counter-argument based on the given initial-argument. the logic pattern should be one of the patterns given in the logic pattern templates. Your output should consist of two parts. 1) a list of logic pattern template identifiers, each of which represents the logic pattern of the corresponding sentence in the counter-argument. 2) a dictionary where keys are the logic pattern template identifiers you selected and values are the corresponding slot-fillers extracted from the counter-argument. You will shown 20 examples below, the format of your output should be exactly the same as that in the examples. The initial-argument is the same for counter-arguments in the examples and the target counter-argument.

Logic pattern templates:
ptn1. The CA admits that <death penalty> promotes <misjudgement>, but argues that this promotion of <misjudgement> can be lessened by Z.
ptn2. The CA admits that <death penalty> promotes <misjudgement>, but argues that there is another factor Z that also promotes <misjudgement>, and it is more important to address Z.
ptn3. The CA asserts that there is no evidence to support the claim that <death penalty> promotes <misjudgement>.
ptn4. The CA argues that <death penalty> does not promote <misjudgement> because there is another true factor Z that promotes <misjudgement>.
ptn5. The CA argues that <death penalty>, on the contrary, suppresses <misjudgement> because it promotes the missed factor Z and Z suppresses <misjudgement>.
ptn6. The CA argues that <death penalty>, on the contrary, suppresses <misjudgement> because it suppresses the missed factor Z and Z promotes <misjudgement>.
ptn7. The CA argues that <misjudgement> is not a bad outcome because <misjudgement> is not severe or is acceptable and therefore does not require addressing.
ptn8. The CA argues that <misjudgement> is not a bad outcome because <misjudgement> leads to a positive outcome Z.
ptn9. The CA argues that <death penalty> promotes a positive outcome Z from a different perspective than that of <misjudgement>.
ptn10. The CA argues that <death penalty> mitigates a negative outcome Z from a different perspective than that of <misjudgement>.
ptn11. The CA employs logic, but none of the aforementioned positions apply.
ptn100. No logical argument is presented, such as a greeting, introduction, or statement of stance.

Initial-argument: Today's topic is "whether the death penalty should be abolished". We believe it should be abolished. Our point is that misjudgment in the death penalty is very dangerous. Humans make mistakes. This is the nature of a human and we cannot deny it completely. Of course, technology has been developed such as DNA testing, surveillance camera, and so on. However, still there is still a possibility of misjudgment. If misjudgment happens when the court sentences the suspected criminal the death penalty, it will be irreparable. When we deprive the life of a suspected criminal, they will never be able to live again. It' s very terrible. Imagine the case where your family member is mistakenly arrested by the police and unfortunately suspected to be sentenced to the death penalty. Would you be able to endure this situation? The answer is definitely "No". Therefore, at least, by abolishing death penalty, we can prevent the situation of ending the life of innocent people. At least, we will have more time to investigate the truth when the innocent in jail are suspected of a serious crime. Then, there will be a greater chance that the innocent, suspected person can be saved in the future from losing their life. It' s much safer society. Remember, even if we carefully investigate the incident, there is always a risk of mistakes. For these reasons, death penalty should be abolished.

Example 1
Counter-argument:
They said that there is always a risk of mistakes.
However, I will oppose the argument that there is always a risk of mistakes.
Mistakes means occurs unknowing and occasionally and not always.
If you take Bin ladens case as an example and if you left him in the prison for life imprisonment then he will create so many terrorist in the country to act on behalf of him from the prison itself.
So based on the type of crimes and criminals the death penalty must be altered.
Output:
{'logic pattern template': "['ptn100', 'ptn7', 'ptn7', 'ptn10', 'ptn11']", 'slot-fillers': "'ptn10': ['create so many terrorist in the country to act on behalf of him from the prison itself'], 'ptn11': 'no slots', 'ptn7': 'no slots'"}
.....
Example 20
....
....
Target counter-argument:
They said that there is the possibility of misjudgement.
However, dNA evidence is very strong.
It is over 99.9% accurate.
The chance of someone with the exact same DNA being in that same place at the same time is close to zero.
If the DNA evidence and other evidence is strong then the death penalty should be an option that needs to be considered.
This way the person who committed the crime will not be able to harm anyone else ever again.
Output:

Table 9: An example prompt for GPT-4-turbo. We used json mode to ensure the output can be validly parsed into a json object.