Identification of Multiple Logical Interpretations in Counter-Arguments

Wenzhi Wang[1,2], Paul Reisert[3], Shoichi Naito[4,2], Naoya Inoue[5,2], Machi Shimmei[1,2], Surawat Pothong[5], Jungmin Choi[2], Kentaro Inui[7,1,2],
[1]Tohoku University, [2]RIKEN, [3]Beyond Reason
[4]Ricoh Company, Ltd., [5]JAIST, [7]MBZUAI
wang.wenzhi.r7@dc.tohoku.ac.jp beyond.reason.sp@gmail.com shohichi.naitoh@jp.ricoh.com
{naoya-i,spothong}@jaist.ac.jp machi.shimmei.e6@tohoku.ac.jp
jungmin.choi@riken.jp kentaro.inui@mbzuai.ac.ae

Abstract

Counter-arguments (CAs) are a good means to improve the critical-thinking skills of learners, especially given that one has to thoroughly consider the logic of initial arguments (IA) when composing their CA. Although several tasks have been created for identifying the logical structure of CAs, no prior work has focused on capturing multiple interpretations of logical structures due to their complexity. In this work, we create CALSA$^+$, a dataset consisting of 134 CAs annotated with 13 logical predicate questions. CALSA$^+$ contains 1,742 instances annotated by 3 expert annotators (5,226 total annotations) with good agreement (Krippendorff α=0.46). Using CALSA$^+$, we train a model with Reinforcement Learning with Verifiable Rewards (RLVR) to identify multiple logical interpretations and show that models trained with RLVR can perform on par with much bigger proprietary models. Our work is the first to attempt to annotate all the interpretations of logical structure on top of CAs. We publicly release our dataset to facilitate research in CA logical structure identification.

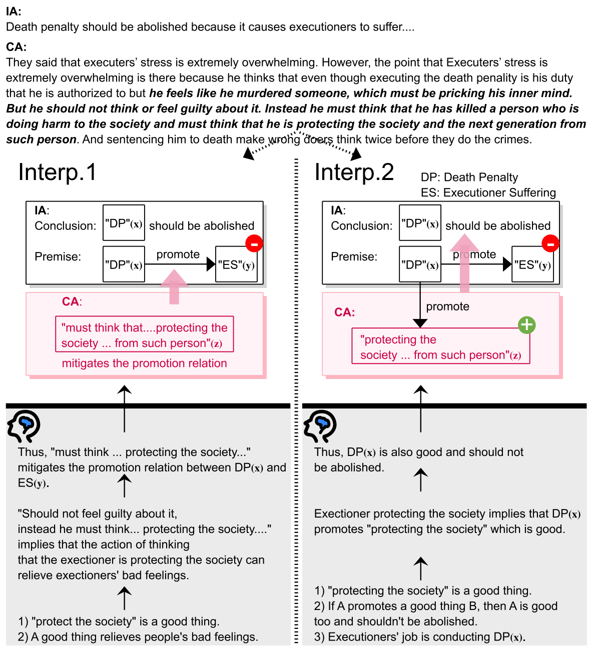

Figure 1: Example of multiple logical interpretations for a CA. The bold, italic text is a segment with, at least, two different logical interpretations. The gray boxes represent possible background knowledge and implicit reasoning useful for identifying the respective interpretation.

1 Introduction

Counter-arguments (CAs) serve as an effective tool for enhancing learners' critical thinking skills (Liu and Stapleton, 2014), especially given that one has to thoroughly consider the logic of initial arguments (IA) when composing a CA. In order to maximize learning efficiency, tailored feedback from teachers is extremely valuable (Hattie and Timperley, 2007). However, it is difficult to provide every learner tailored feedback due to limited human resources and heavy workloads (Paris, 2022). Therefore, developing a system that can automatically provide feedback to learners' CAs for improving their critical-thinking skills would be a beneficial way of applying artificial intelligence (AI) technology to the educational field.

In general, written texts can be interpretted multiple ways due to the potential discrepancy between writers' intent and readers' understanding (Iser, 1972; Tierney et al., 1983; Rosenblatt, 1988; Mart et al., 2019). Moreover, readers can also have different interpretations of the same text due to different background knowledge. In the context of providing feedback for CAs, this becomes a challenge given the overall complexity of CA logical structures. Towards identifying these logical structures, inspired by Argumentation Schemes (Walton et al., 2008), Naito et al. (2024) proposed a typology of patterns to capture common CA logical structures adhering to the Argument from Consequences scheme, where each pattern was associated with unique feed-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6430–6445

back. They framed the identification of logical structures as the Counter-Argument Logical Structure Analysis (i.e., CALSA) task. Although their work is groundbreaking, it only interpreted one logical structure for a given argumentative segment.

Figure 1 shows an example of a written text (CA) with at least two possible interpretations of the logical structure, henceforth *logical interpretations* for the argumentative segment in bold (Here, we use *logical interpretation* to refer to a logical structure defined in the CLASA typology). One interpretation is that the CA acknowledges the *premise* in the IA's logic (x promotes y, where x="death penalty", y="executioner's suffering), but it claims that the relation between x and y can be mitigated by a z="executioners considering themselves protecting society" (Interp. 1). The other interpretation can be that the *conclusion* of the IA's logic is attacked, as the CA claims that x should not be abolished since it promotes a good thing z="protecting the society" (Interp. 2). When creating a system for providing feedback to learners, it is important that system accounts for such multiple interpretations of logical structures within a CA.

We argue that it is crucial to collect and identify multiple logical interpretations for the following reasons: 1) *correctness*: in the educational context, providing feedback that is not aligned with the intent of a learner's writing may cause them to become less motivated towards articulating their own thoughts, and even cause them to become disinterested in writing (Brannon and Knoblauch, 1982; Treglia, 2008). Therefore, collecting as many interpretations as possible could prevent potential misalignment caused by only having a single interpretation; 2) *informativeness*: Showing multiple possible interpretations of CA logical structures to writers can potentially facilitate the reflection on their own writings (i.e., learners can be aware of additional interpretations and revise accordingly).

To the best of our knowledge, no resource exists for identifying multiple logical interpretations in CAs. Towards tackling this issue, we aim to answer the following research questions: **(i)**: How can we collect as many multiple logical interpretations in a CA as possible?, and **(ii)**: To what extent can we utilize current Large Language Models (LLMs) to identify multiple logical interpretations in a CA? To answer these questions, we explore decomposing the original logical structures defined in Naito et al. (2024) into several finer-level, logical predicates, each representing an independent partial structure of the CA logic. We conduct an annotation study to create CALSA$^+$, a new dataset of predicates annotated on top of a CA to collect as many logical interpretations as possible. Utilizing CALSA$^+$, we conduct model experiments using two modeling methods: Prompt Engineering (PE) and Reinforcement Learning from Verifiable Rewards (RLVR). Although we discover that the models can solve the task to some extent, many challenges still remain.

Our contributions can be summarized as follows:

- We create CALSA$^+$, a dataset composed of multiple logical interpretations. CALSA$^+$ consists of 134 CAs annotated with 13 predicate-related questions. In total, it contains 1,742 instances annotated by 3 expert annotators (5,226 total annotations) with good agreement (Krippendorff α=0.46 [1]). We publicly release our dataset [2]. Our dataset is the first to include multiple CA logical interpretations and can be further studied by the broader community.

- We conduct model experiments with various methods and establish a baseline for the CALSA$^+$ dataset. To the best of our knowledge, our work is the first to explore RLVR for improving the reasoning abilities for a CA logic parsing task.

2 Related Work

Various works have focused on subjectivities embedded in NLP tasks. Pavlick and Kwiatkowski (2019) argues that the disagreements in NLI tasks are not noise, but instead, useful information and models trained to do NLI tasks should produce the distribution of human ratings instead of a single aggregated label. Some works focus on utilizing the opinions of different annotators for the downstream tasks, e.g., creating an annotator embedding or predicting the ratings of individual annotators (Fleisig et al., 2023; Deng et al., 2023). Ferracane et al. (2021) studies the subjective judgments on conversational acts and intents of response in congressional hearing settings. Many other works also build on the idea that a single ground truth label cannot reflect the true nature of the target task (Das

[1]Although we obtain a good IAA, our annotation does not expect all annotators to agree with each other. The disagreement in our annotations is an important signal indicating the different background knowledge and implicit reasoning annotators use for interpreting CAs (Section 3.4)

[2]https://github.com/cl-tohoku/ca-multi-ptn

et al., 2017; Poesio et al., 2019; Nie et al., 2020; Jiang and de Marneffe, 2022; Plepi et al., 2022; Heinisch et al., 2023; Jiang et al., 2023).

There have been many works that focus on CAs in the field of computational argumentation. Several focus on automatically creating CAs through retrieval, generation, or combining both (Wachsmuth et al., 2018; Hua et al., 2019; Alshomary and Wachsmuth, 2023; Alshomary et al., 2021; Lin et al., 2023; Yeginbergen et al., 2025). In comparison, the main focus of this paper is on the automatic identification of logical structures of CAs. Furthermore, some of the previous works focus on creating datasets of CAs, and annotating CAs with structural information (Reisert et al., 2019; Mim et al., 2022; Naito et al., 2024). Among them, the most relevant work is Naito et al. (2024) due to their typology created for counter-arguments. While they focus on annotating one logical structure per CA segment, we focus on collecting multiple interpretations of logical structures in CAs.

3 CALSA$^+$ Creation

We briefly introduce an existing task and dataset for CA logical structure identification and describe the details regarding the construction of CALSA$^+$.

3.1 CALSA Task and Dataset

CALSA is a task that focuses on analyzing the logical structure of a CA in relation to an IA (Naito et al., 2024). Inspired by Reisert et al. (2018), they utilize a template-based approach where a CA logical structure is considered to be a combination of a logic template and slot-fillers. For the task, a dataset was created which consisted of more than 700 CAs written in English, each of which was annotated with sentence-level logic templates. The templates were selected from a pre-defined set of 10 templates created based on the Argument from Consequences scheme in Walton et al. (2008). If a template had a placeholder for a slot-filler, the annotators were instructed to extract a suitable slot-filler. For a CA, which consists of multiple mutually exclusive segments $\{S_i\}_{i=1}^{n}$, where each segment S_i contains multiple consecutive sentences $\{e_j\}_{j=k}^{k+m}$, their annotation protocol allowed for one logical structure label per sentence e_j. As a result, the CALSA dataset contains a single logical structure label for each segment S_i (taking the label of its constituent sentences). They conducted several baseline experiments to exhibit that it is challenging for the current LLMs to identify the underlying logical structure embedded in the CA, but their experiments were limited to training LLMs to directly generate the target label of the logical structure. Furthermore, while they claim that multiple logical structures may exist in a single segment S_i due to multiple possible interpretations of the CA segment, their annotation restricted their dataset to one CA logical structure, which also hinders the reliability of evaluating LLMs' abilities in solving the task.

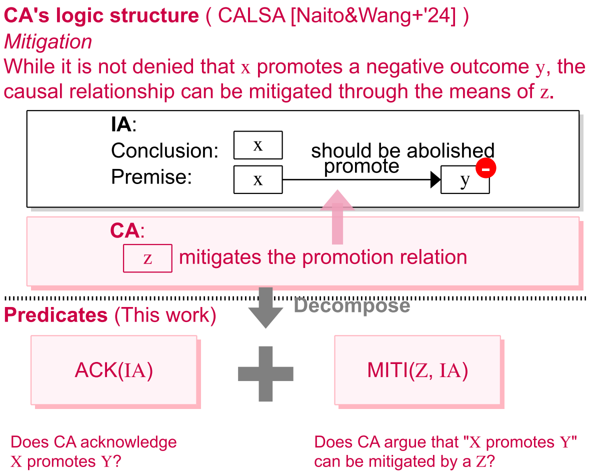

Figure 2: Example of decomposing one CA logical structure into its constituent predicates. We show the mapping between predicates and all logical structures in Appendix B.

As discussed in Section 1, we opt for utilizing the CALSA dataset since it is the only dataset that has labels of deep logical structure of argumentative texts (CAs), which can ultimately be used for constructive feedback. Furthermore, given the high coverage reported in Naito et al. (2024), the CALSA dataset is capable of covering common CA logical structures, which renders it suitable for our purpose. For carrying out our annotation, we utilize a portion of IA-CA pairs in the CALSA dataset.

3.2 Extending CALSA with multiple logical intepretations

In order to provide correct and informative feedback, we aim to create a dataset with as many CA logical interpretations as possible. We formalize collecting as many logical interpretations of a CA as possible as a feasible problem, in which the number of interpretations is bounded to the total number of the unique logical structures defined in CALSA (Naito et al., 2024) (i.e., 10). We discuss our annotation in detail as follows.

Predicate P	Example
ACK(IA)	acknowledge that death penalty promotes misjudgment
DENY(IA)	death penalty does not promote misjudgment
REVERSE(IA)	death penalty suppresses misjudgment
MITI(Z,IA)	"death penalty promotes misjudgment" can be mitigated by a thing Z.
ANO(Z,Y)	there is a thing Z other than death penalty that also promotes misjudgment
NO-EVI(IA)	IA doesn't mention any evidence to support that death penalty promotes misjudgment
NO-NEED-ADDRESS(Y)	misjudgment is not important or is not a problem that requires any action to be taken
SUP(Y,Z)	misjudgment suppresses a bad thing Z
PRO(Y,Z)	misjudgment promotes a good thing Z
PRO(X,Z)	death penalty promotes a good thing Z.
SUP(X,Z)	death penalty suppresses a bad thing Z
TRANS1(X,Z,Y)	death penalty suppresses a thing Z that promotes misjudgment
TRANS2(X,Z,Y)	death penalty promotes a thing Z that suppresses misjudgment

Table 1: Example of one set of decomposed predicates for an IA. Given the IA logic IA (e.g., IA=death penalty promotes misjudgment which is bad), X and Y refer to slot-fillers taken from IA (e.g., X="death penalty" and Y="misjudgment"). Z refers to a slot-filler that satisfies the respective predicate logic and should be extracted from the CA. Each IA has its own set of predicates. Predicates across IAs have the same logical structure but slightly different wordings depending on the actual IA logic.

CALSA^{+} Predicate Inventory We explore decomposing the 10 CA logical structures from the CALSA dataset into finer-level predicates. At a high level, we consider that there are three parts where a CA logic can interact with the corresponding IA: the *conclusion of the IA*, the *causal relation between variables* (concept slot-fillers X and Y in the IA), and the *variables* themselves (X or Y). Each CA logical structure interacts with one or more parts in the IA. We consider such interaction as a *partial logic* and decompose the 10 CALSA logical structures into a set of partial logics. We refer to each unique partial logic as a *predicate*. However, since a valid CA must deny IA's conclusion, the interaction with the conclusion is not informative for distinguishing different CA logical structures. Thus, for our task, we only consider the predicates that represent an interaction with the causal relation or with variables. Examples of the decomposition procedure are shown in Appendix H. As a result, we decompose the 10 CALSA logical structures into 13 finer-level predicates, where each predicate P represents an independent partial logical structure of a CA.

Each predicate, along with an instantiated example, is shown in Table 1. The original labels for CALSA logical structures can thus be obtained by aggregating the labels for all predicates (a logical structure exists if all of its constituent predicates exist). An example is shown in Figure 2. The reason for the our decomposition is three-fold: 1) Given the complexity of annotating deep logical structures, we anticipate that decomposing the complex logical structures into small predicates and annotating them individually would ease the annotation procedure and consequently improve the quality of the final results; 2) Given that the CALSA logical structures can be considered as a combination of multiple finer-level logics, it is possible for annotators to be biased towards one part and overlook another. By explicitly considering each finer-level predicate, we could alleviate such bias and improve the quality of annotations; 3) While there are unlimited approaches for decomposition, our approach ensures that there is unique feedback associated with each predicate, which can be derived from the original feedback provided in the CALSA dataset.

Annotation Procedure We convert each predicate P to a binary question (e.g., "Does the CA acknowledge that death penalty promotes misjudgment?" for ACK(IA) in Table 1). Given an IA logic (e.g., "death penalty promotes executioner's suffering") and CA pair, annotators are also given 13 total binary questions. For each question, the answer is YES if the given predicate exists (the partial logical structure exists) in the CA; otherwise, NO. If the predicate P has a slot-filler Z, annotators are also required to extract a slot-filler from the CA; otherwise, the annotators are required to select sentences from the CA as evidence when answering YES to a given binary question. Annotators are allowed to provide optional reasoning to support their annotation. We create a custom annotation interface for achieving the annotation process as shown in Appendix A. Given that our purpose is to create a high-quality dataset, opposed to utilizing crowdsourcing, we chose to perform the annotation

Predicate	3 YES	2 YES	1 YES	All NO
ACK(IA)	37	29	18	50
ANO(Z,Y)	20	20	27	67
DENY(IA)	12	25	26	71
MITI(Z,IA)	8	14	28	84
NO-EVI(IA)	0	1	2	131
NO-NEED-ADDRESS(Y)	14	18	33	69
REVERSE(IA)	5	11	29	89
TRANS1(X,Z,Y)	1	7	22	104
TRANS2(X,Z,Y)	10	17	31	76
PRO(X,Z)	42	41	18	33
SUP(X,Z)	25	37	30	42
PRO(Y,Z)	11	6	19	98
SUP(Y,Z)	1	7	17	109
Total	186	233	300	1023

Table 2: Distribution of answers per predicate P in our dataset. Each P in our dataset is comprised of 134 answers corresponding to 134 CAs respectively.

with expert annotators. Following previous works that utilize expert annotators (Wachsmuth et al., 2017; Robbani et al., 2024), we employ three expert annotators, one native, and two fluent English speakers, all experts in the field of argumentation.

3.3 Results

For determining the final labels [3], we want to collect as many logical interpretations as possible, even the less obvious ones. Therefore, we want to utilize as many YES labels at the predicate level as possible as it allows us to aggregate more logical interpretations. To examine the quality of the answer YES and analyze the results, we sample 50 predicate-level instances that do not have full agreement. Annotators discussed the sampled results and provided reasoning for their answers. As a result, we found that the answer YES is considered reasonable for 49/50 sampled instances. The slot-fillers associated with YES labels were also considered reasonable. Therefore, we decided to use YES as the final label for a predicate if any of the annotators selected YES, otherwise, the final label was NO. For slot-fillers, we consider all of the selected slot-fillers for a YES answer as the label. The distribution of predicate-level labels of answers is shown in Table 2. We aggregated the results of predicate-level annotations to obtain all labels of logical structures for a CA to create CALSA$^+$ dataset. We show basic statistics of CALSA$^+$ dataset in Table 3.

For all predicates that are labeled as YES and have a slot-filler, we report the results of collected slot-fillers. In total, annotators select 745 unique slot-fillers[4], with 11 slot-fillers agreed between all annotators (i.e., the three annotators select the lexically identical slot-fillers for 11 instances), 77 slot-fillers agreed by two annotators, and the remaining 657 slot-fillers selected once by one annotator. Additionally, we calculate the similarity between slot-fillers using ROUGE scores (Lin, 2004). For each predicate that has two or more YES, we calculate it between every pair of slot-fillers for all combinations of the selected slot-fillers. In total, we obtain 503 slot-filler pairs. The distribution of ROUGE-1 scores above a certain threshold is shown in Figure 3. Out of the 503 pairs, although we did not expect annotators to select the same slot-fillers, more than half have a score above 0.5 which indicates that annotators tend to focus on the same slot-fillers for certain predicates. We also compare the similarity between the selected evidence. The results are shown in Appendix C.

# CAs	134
# Predicate level annotations	1742
# Logic structures per CA on average	3.63

Table 3: The basic statistics of the CALSA$^+$ dataset.

3.4 Analysis

Given the existence of multiple logical interpretations in CALSA$^+$, we ask the following question: *What are the main sources of the different annotators' understanding that lead to multiple interpretations?* We conduct a manual analysis on 50 of the disagreed predicate-level instances from CALSA$^+$, as the predicates are the finer-level constituents of logic structures. For instances in which the optional reasoning was not provided in the original annotation, annotators discussed their reasoning together. Following both the discussion and analysis, we found that the sources of different logical structures can be located at the predicate level, namely, the answers YES and NO can simultaneously be reasonable for certain instances. Overall, we found three main sources for the phenomenon (henceforth, *ambiguity source*): scalar implicature (SOURCE1), different interpretations of concepts (SOURCE2), and utilizing different background knowledge to complement implicit information (SOURCE3). Each source further includes finer categories. We show representative examples in Table 4.

[3] We achieve Krippendorff α=0.46 for the overall annotations.

[4] We consider lexically identical slot-fillers from different instances different slot-fillers.

Example CA and P	**2 Agree**	**1 Agree**	**Disagreement Reason**
CA: .. however, ... every job is not meant like this where one can keep themself updated with the real world or what is happening in society.... P: SUP(X,Z) **Q:** CA argues that "*part-time jobs*"$_{(X)}$ suppresses a good thing?	NO, "*every* ..." means that there are jobs that promote "*keep...*".	YES, "*every* ..." means that there are jobs that suppress "*keep them self updated with the real world...*".	(SOURCE1) "*every*" can mean both sides; scalar implicature
CA: .. homework is a compulsory exercise that allows students to make decisions about what to do now and what not to do... P: TRANS2(X,Z,Y) **Q:** CA argues that "*homework*"$_{(X)}$ promotes a thing that suppresses "*being passive in character*"$_{(Y)}$?	YES, it promotes "*make decisions about what to do now and...*" which is *active learning* that suppresses Y.	NO, nothing is promoted.	(SOURCE2) The boundary between two concepts: whether "*making decisions*" is equivalent *active learning* (i.e., *not passive*).
CA: .. however, I think it's just as likely that forcing students to do homework makes students rebellious and resistant to authority...... P: REVERSE(IA) **Q:** CA argues that "*homework*"$_{(X)}$ suppress "*being passive in character*"$_{(Y)}$?	YES, homework makes students active in character ("*rebellious and..*"), which suppresses Y.	NO, "*forcing students to do homework*" instead of "*homework*" promotes active character which suppresses Y.	(SOURCE2) Different interpretation of X.
CA: .. homework can establish the basic foundation of studying because homework is a good guideline of what we should review and study. Studying is based on the accumulation of understanding, once we get behind, catching up classes is difficult.... P: TRANS1(X,Z,Y) **Q:** CA argues that "*homework*"$_{(X)}$ suppresses a thing that suppresses "*free time*"$_{(Y)}$?	NO, CA does say "*It would be difficult to catch up*" but it doesn't state that that would take up more free time.	YES, homework suppresses "*get behind*" which suppresses free time because if someone were to get behind, they would use more time to catch up.	(SOURCE3) The degree to which annotators read from context is different.
CA: .. even if some students are cheating by copying friends' homework, they will easily notice such kind of ways to study are not beneficial....... P: SUP(Y,Z) **Q:** CA argues that "*incorrect ways of studying*"$_{(Y)}$ suppresses a bad thing?	NO, nothing bad is suppressed.	YES, Y promotes "*notice such kind of....not beneficial*" which stops students from doing Y which is bad.	(SOURCE3) Implicit circular logic chain.
CA: .. students also learn that copying work is counter to actually studying and learning the material. Abolishing testing or homework deprives students of the chance to learn this. P: PRO(X,Z) **Q:** CA argues that "*homework*"$_{(X)}$ promotes a good thing?	NO, nothing good is promoted.	YES, abolishing homework suppresses the chance to *learn this* which means homework promotes the *chance to learn this*.	(SOURCE3) Reversed reasoning.

Table 4: Examples of P with disagreements. "2 Agree" and "1 Agree" refer to two annotators agreeing and one annotator agreeing, respectively.

3.5 Discussion on ambiguity sources

Anjali and Babu (2014) present a study of different types of ambiguities in NLP (e.g., lexical ambiguity, semantic ambiguity, etc.). Our findings of the sources of different logical interpretations can be categorized using their typology. Specifically, the first source, scalar implicature (SOURCE1), can be seen as scope ambiguity where the usage of quantifiers causes ambiguity. It has also been studied as a linguistic phenomenon by language experts (Geurts, 2009). SOURCE2 and SOURCE3 can be broadly categorized as pragmatic ambiguity, where context causes ambiguity. Recently, Li et al. (2024) propose a taxonomy for analyzing ambiguity in the modern NLP context. Although their taxonomy has more fine-grained types, they are more related to the semantic meaning of the sentences/words themselves, whereas our SOURCE2 and SOURCE3 are more related to the background knowledge and level of implicit reasoning of the readers. Furthermore, previous work has also focused on addressing specific ambiguity types in the context of presenting a focused study (Itankar and Raza, 2020; Abeysiriwardana and Sumanathilaka, 2024; Kamath et al., 2024), creating datasets (Liu et al., 2023; Yuan et al., 2023; Kamath et al., 2024), and proposing a new modeling method (Kim et al., 2023, 2024). However, they only focus on lexical, syntactic, or semantic ambiguities of the target sentences themselves, whereas, as aforementioned, our SOURCE2 and SOURCE3 are rooted in the differences in the implicit reasoning of the background of the readers. Given the differences between ambiguity sources in our work compared to existing work, we believe our findings can also facilitate future studies on ambiguity. Overall, the main sources are closely relevant to the long-standing challenge of NLP tasks, knowledge and implicit reasoning, especially for SOURCE2 and SOURCE3.

4 Addressing CALSA$^+$ with LLMs

We conduct experiments on identifying multiple logical interpretations of CA using our new dataset, CALSA$^+$. Our experiments aim to answer our aforementioned research question: **(ii)**: To what extent can we utilize current LLMs to identify multiple logical interpretations in a CA?

4.1 General Design

We conduct our experiments at the predicate level, consistent with the annotation procedure. We consider the task a binary Question-Answering task (QA). Given an IA, a CA, and a question for a predicate P, models are expected to output an answer YES or NO, generate a slot-filler Z if applicable, and generate the reasoning. We set the expected format of the models' output to be: <think>{reason}</think><answer>{answer}</answer><slot>{slot-filler}</slot>. We focus on predicate-level experiments for the following reasons: 1) We anticipate that identifying predicates would be easier than directly identifying all the logical structures simultaneously since the predicates are the decomposition of the complex logical structures. 2) Identifying predicates would provide finer-level insights as to which part of the logic models identify both correctly and incorrectly, which facilitates finer-level analysis. Through aggregation of the results from our experiment on predicate identification, we can obtain the results for multiple logical interpretations identification. All the experiments are conducted with 3-fold cross-validation. The basic statistics for a fold are shown in Table 6. We conduct our experiments with Reinforcement Learning from Verifiable Rewards (RLVR), Supervised Fine-Tuning (SFT) and Prompt Engineering (PE), respectively.

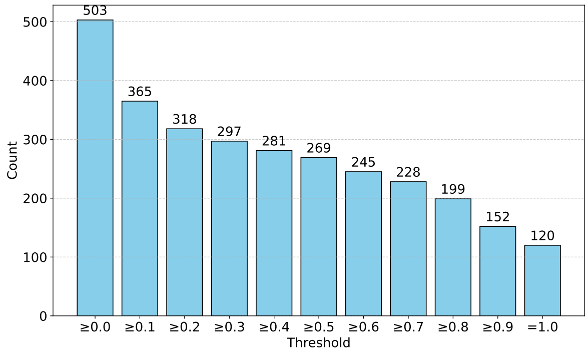

Figure 3: Number of pairs of slot-fillers whose ROUGE-1 score is above a certain threshold.

4.2 Modeling methods

We explore fine-tuning models with RLVR (Lambert et al., 2024) since the RL approach is reported to significantly enhance the general reasoning abilities of pre-trained LLMs, especially in solving math and coding problems (Lambert et al., 2024; Guo et al., 2025; Wei et al., 2025). Given that the implicit reasoning ability is the key part to identifying a logical structure, we want to examine to what extent the RL approach is effective in our

Methods	Models	Ptn.Acc	Predicate.F1	Slot.ROUGE-1.All	Slot.BertScore.All	Slot.ROUGE-1.R	Slot. BertScore.R
RLVR	Qwen2.5-7B-Instruct	.545 (.05)	**.783** (.02)	**.791** (.03)	**.964** (.00)	.415 (.08)	.683 (.04)
	Mistral-7B-Instruct-v0.3	**.560** (.10)	.761 (.05)	.760 (.02)	.959 (.00)	.377 (.04)	.686 (.03)
	Falcon3-7B-Instruct	.544 (.02)	.752 (.02)	.752 (.01)	.959 (.00)	.381 (.03)	.678 (.03)
SFT	Qwen2.5-7B-Instruct	.559 (.01)	.747 (.03)	.749 (.04)	.960 (.01)	.353 (.02)	.697 (.07)
	Mistral-7B-Instruct-v0.3	.449 (.05)	.673 (.03)	.545 (.07)	.930 (.00)	.264 (.17)	.780 (.13)
	Falcon3-7B-Instruct	.525 (.05)	.756 (.01)	.747 (.01)	.959 (.00)	.326 (.01)	.644 (.00)
Zero-shot	Qwen2.5-7B-Instruct	.342 (.00)	.659 (.02)	.696 (.03)	.943 (.01)	.248 (.01)	.561 (.03)
	Mistral-7B-Instruct-v0.3	.202 (.07)	.534 (.03)	.426 (.01)	.872 (.01)	.244 (.02)	**.830** (.01)
	Falcon3-7B-Instruct	.358 (.04)	.607 (.04)	.535 (.02)	.907 (.01)	.223 (.01)	.667 (.03)
	gpt-4.1-2025-04-14	.483 (.02)	.748 (.01)	.766 (.02)	.960 (.00)	.452 (.06)	.726 (.01)
	o4-mini-2025-04-16	.462 (.02)	.665 (.01)	.659 (.01)	.942 (.00)	**.498** (.05)	.821 (.01)

Table 5: Averaged 3-fold experiment results. The standard deviation is shown in parentheses. Ptn.Acc: the multi-label accuracy of multiple logical structure identification; Predicate.F1: the macro F1 score for predicate identification; Slot.{target}.All: the average target score of slot-fillers calculated for all instances at the predicate level; Slot.{target}.R: the average target score of slot-filler calculated for instances where that label has a slot-filler. The results are rounded up to the respective decimal position. All scores range from 0 to 1. The higher the better.

Split	#CAs	#Predicate-level instances
train	84	1092
val	25	325
test	25	325

Table 6: The basic statistics of a fold. All three folds contain the same number of instances.

task. We utilize the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024) for training our models. Following their work, our reward function is also designed to contain two parts: the format and the correctness of the content. We also conduct zero-shot and STF experiments as baselines. We show more details in Appendix D.

4.3 Results

We evaluate the models' performance through various metrics. For logical structure identification, we use multi-label accuracy, for predicate identification, we use macro-F1 score to account for both YES and NO labels. For slot-fillers, we consider the ROUGE score and BertScore for evaluating both surface-level similarity and semantic-level similarity respectively. We consider the predicted slot-filler correct if it matches any of the selected slot-fillers by annotators. Specifically, for each CA, we consider the highest score as the score for that instance and then average the scores for all CAs for both ROUGE and BertScore. We also consider two scenarios: 1) since not all predicates have slot-fillers, when the target predicate does not have a slot-filler, we do not want models to generate one. Thus, we evaluate for all instances, denoted as **All** (i.e., if the label does not contain a slot-filler, models should not generate one, otherwise it will be penalized). 2) We observed that 245 out of 325 test instances do not contain a slot-filler (i.e., models are not supposed to generate a slot-filler for those instances), thus, it makes us wonder about the quality of the actually generated slot-fillers. We thus calculate both ROUGE score and BertScore for instances that have a slot-filler (80/325), denoted as **R**ecall. The results are shown in Table 5. Overall, the high Predicate.F1 scores show that the models tested here are capable of identifying the logical structure to some extent. For slot-fillers, the promising results for type **All** show that models generally do not predict a slot-filler when the label does not have one. However, the comparably low scores for type **R** show that the actual generated slot-fillers differ from the annotated ones to a certain extent. Furthermore, though trained on a comparably small dataset, models trained with RLVR constantly display better performances than their zero-shot and SFT counterparts. The performances are also consistent with OpenAI's larger models under zero-shot settings. This verifies the effectiveness of RLVR in solving tasks other than math or coding. Our RLVR modeling results per predicate are shown in Appendix I.

4.4 Analysis

To further analyze the results and to understand the challenges regarding modeling, we perform two levels of manual investigations on model outputs. Given that the automatic evaluation results do not distinguish significantly across different models and methods, we use the outputs of Qwen2.5-7B-Instruct as a representative for analysis.

First, we conduct analysis on the finer-level predicate identification task. We manually check 71 test instances in which the model's predicted answer is different than the label (henceforth, non-aligned instances). For all 28 non-aligned instances where that label is NO (i.e., all three annotators agree that the target predicate does not exist in the given CA) and model's prediction is YES, we found that, interestingly, the predicted answers for 19 non-aligned instances could be considered reasonable, which indicates that model sometimes identifies implicit logic that even annotators did not realize during the annotation. We show an example in Appendix E (the first one). For the 19 reasonable instances, the main source behind the non-alignment is consistent with that found in annotation results (Section 3.4). However, for the 8 instances where the predicted answer is not reasonable, we found that the main reason (5/8) is that the model generates correct reasoning, which can semantically induce the correct label, but the actual generated answer is the opposite. Furthermore, for 44 non-aligned instances where the label is YES whereas model's predicted answer is NO, we found that model has difficulty identifying non-obvious logic, consistent with the obviousness of our labels (i.e., how many annotators select YES). Specifically, 23 non-aligned instances have 1-YES labels, 18 non-aligned instances have 2-YES labels, and only 2 non-aligned instances have 3-YES labels. We also show an instance of a non-obvious logic that the model fails to identify in Appendix E (the second one).

To assess the quality of the generated slot-fillers, we conducted an additional annotation study on top of 40 generated slot-fillers for instances that were not an exact match with any of the gold slot-fillers selected by annotators. Specifically, two expert annotators labeled whether the generated slot-filler is correct or incorrect in terms of representing a slot-filler for the given binary predicate question. As a result, annotators agreed on 35/40 (87.5%) instances, where 30 instances were agreed to be correct and five instances were agreed to be incorrect. After the annotation, the five disagreeing instances (12.5%) were discussed amongst annotators. Both agreed that two of the remaining instances could be considered correct or incorrect depending on how one interprets the concept Y in the original question. This is consistent with our findings in Section 3.4, specifically, the SOURCE2 for different interpretations. The other three disagreements were attributed to human error, and after the discussion, two were agreed to be incorrect and one correct. The final resulsts indicate that 33/40 instances can be considered correct and that the generated slot-fillers can be considered high quality.

We conduct an analysis on the results for identifying multiple logical interpretations. Given that logical interpretation labels are obtained through aggregating the predicate-level labels, the findings for predicate-level results are closely related to the results here. We found that since the model struggles to identify non-obvious logic, the average number of logical structures (2.88) identified is smaller than the labels (3.56). For identifying different logical interpretations for certain CAs, the model must think "differently". Nevertheless, it usually provides similar reasonings to predicates of different logical interpretations. This also contributes to failing to identify diverse interpretations of a CA.

Overall, our experiments demonstrate that while current LLMs show potential in identifying multiple interpretations of a CA, challenges still remain. Specifically, identification of the non-obvious logic, which can also be challenging for human annotators, without explicit explanations is still a challenge. Furthermore, the inconsistencies between the generated reasoning and the final predicted answer is another challenge that must be addressed.

5 Conclusion and Future Work

In this work, we addressed the task of identifying multiple logical interpretations of a CA. We created CALSA^{+}, a dataset containing 134 CAs with multiple logical interpretations collected via 13 predicate-related questions. We conduct an analysis on our dataset and find three main sources of disagreement. Using our dataset, we conducted modeling experiments and found that models trained with RLVR show effectiveness in addressing the task. Meanwhile, challenges, especially the inconsistencies between the generated reasoning and the final answer, along with identifying the non-obvious logic, still remain. We hope our dataset and the experiment results will inspire further research in identifying multiple logical interpretations in argumentative texts. Furthermore, while our work is based on the CALSA dataset, which was originally created in the context of a debate setting, we believe our predicates can also be leveraged to analyze other types of CAs, such as those in the legal domain (e.g., court cases), general domain (e.g., Reddit threads), etc. In our future work, we plan to utilize our dataset to determine which logical interpretations align with the original writer's intention and whether modeling is possible for determining the most likely plausible interpretations.

Limitations

Dataset Size The number of overall annotations for CALSA^{+} can be considered abundant from the perspective of total annotations. We collected 1,742 annotations per annotator, resulting in 5,226 total annotations. However, from the perspective of unique CAs and predicates, our dataset can be considered small. Each annotator only annotated 134 CAs. For each CA, although 13 total predicates are annotated, this means that we only collect, at most, 134 instances for each predicate type.

Number of Annotators Ideally, for each CA, we would like to collect an abundant amount of annotations from several annotators with different background knowledge through means such as crowdsourcing. However, given the complexity of the task and the time to train crowdworkers, our annotation is limited to only three expert annotators.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 22H00524. We sincerely thank all the reviewers and meta-reviewers for the time and effort devoted to reviewing our work.

References

Miuru Abeysiriwardana and Deshan Sumanathilaka. 2024. A survey on lexical ambiguity detection and word sense disambiguation. *arXiv preprint arXiv:2403.16129*.

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

Milad Alshomary and Henning Wachsmuth. 2023. Conclusion-based counter-argument generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 957–967, Dubrovnik, Croatia. Association for Computational Linguistics.

M Kl Anjali and Anto P Babu. 2014. Ambiguities in natural language processing. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(5):392–394.

Lil Brannon and Cyril H Knoblauch. 1982. On students' rights to their own texts: A model of teacher response. *College Composition & Communication*, 33(2):157–166.

Debopam Das, Manfred Stede, and Maite Taboada. 2017. The good, the bad, and the disagreement: Complex ground truth in rhetorical structure analysis. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 11–19, Santiago de Compostela, Spain. Association for Computational Linguistics.

Naihao Deng, Xinliang Zhang, Siyang Liu, Winston Wu, Lu Wang, and Rada Mihalcea. 2023. You are what you annotate: Towards better models through annotator representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12475–12498, Singapore. Association for Computational Linguistics.

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2021. Did they answer? subjective acts and intents in conversational discourse. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1644, Online. Association for Computational Linguistics.

Eve Fleisig, Rediet Abebe, and Dan Klein. 2023. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6715–6726, Singapore. Association for Computational Linguistics.

Bart Geurts. 2009. Scalar implicature and local pragmatics. *Mind & Language*, 24(1):51–79.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of Educational Research*, 77(1):81–112.

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! In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11138–11154, Singapore. Association for Computational Linguistics.

Xinyu Hua, Zhe Hu, and Lu Wang. 2019. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, Florence, Italy. Association for Computational Linguistics.

Wolfgang Iser. 1972. The reading process: A phenomenological approach. *New literary history*, 3(2):279–299.

Prashant Y Itankar and Nikhat Raza. 2020. Ambiguity resolution: An analytical study. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 6(2):471–479.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374.

Nan-Jiang Jiang, Chenhao Tan, and Marie-Catherine de Marneffe. 2023. Understanding and predicting human label variation in natural language inference through explanation. *arXiv preprint arXiv:2304.12443*.

Gaurav Kamath, Sebastian Schuster, Sowmya Vajjala, and Siva Reddy. 2024. Scope ambiguities in large language models. *Transactions of the Association for Computational Linguistics*, 12:738–754.

Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sanggoo Lee, and Taeuk Kim. 2024. Aligning language models to explicitly handle ambiguity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1989–2007, Miami, Florida, USA. Association for Computational Linguistics.

Jong Myoung Kim, Young-jun Lee, Sangkeun Jung, and Ho-jin Choi. 2023. Semantic ambiguity detection in sentence classification using task-specific embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 425–437, Toronto, Canada. Association for Computational Linguistics.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Margaret Y Li, Alisa Liu, Zhaofeng Wu, and Noah A Smith. 2024. A taxonomy of ambiguity types for nlp. *arXiv preprint arXiv:2403.14072*.

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

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

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

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.

Cagri Mart and 1 others. 2019. Reader-response theory and literature discussions: A springboard for exploring literary texts. *The New Educational Review*, 56(2):78–87.

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.

Shoichi Naito, Wenzhi Wang, Paul Reisert, Naoya Inoue, Camélia Guerraoui, Kenshi Yamaguchi, Jungmin Choi, Irfan Robbani, Surawat Pothong, and Kentaro Inui. 2024. Designing logic pattern templates for counter-argument logical structure analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11313–11331, Miami, Florida, USA. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Brit Paris. 2022. Instructors' perspectives of challenges and barriers to providing effective feedback. *Teaching and Learning Inquiry*, 10.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Joan Plepi, Béla Neuendorf, Lucie Flek, and Charles Welch. 2022. Unifying data perspectivism and personalization: An application to social norms. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7391–7402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Massimo Poesio, Jon Chamberlain, Silviu Paun, Juntao Yu, Alexandra Uma, and Udo Kruschwitz. 2019. A crowdsourced corpus of multiple judgments and disagreement on anaphoric interpretation. 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 1778–1789, Minneapolis, Minnesota. Association for Computational Linguistics.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Paul Reisert, Benjamin Heinzerling, Naoya Inoue, Shun Kiyono, and Kentaro Inui. 2019. Riposte! a large corpus of counter-arguments. *arXiv preprint arXiv:1910.03246*.

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.

Irfan Robbani, Paul Reisert, Surawat Pothong, Naoya Inoue, Camélia Guerraoui, Wenzhi Wang, Shoichi Naito, Jungmin Choi, and Kentaro Inui. 2024. Flee the flaw: Annotating the underlying logic of fallacious arguments through templates and slot-filling. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20524–20540, Miami, Florida, USA. Association for Computational Linguistics.

Louise M Rosenblatt. 1988. Writing and reading: The transactional theory. technical report no. 416.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Robert J Tierney and 1 others. 1983. Author's intentions and readers' interpretations. technical report no. 276.

Maria Ornella Treglia. 2008. Feedback on feedback: Exploring student responses to teachers' written commentary. *Journal of Basic Writing*, pages 105–137.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

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, Valencia, Spain. Association for Computational Linguistics.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

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

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. 2025. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*.

Anar Yeginbergen, Maite Oronoz, and Rodrigo Agerri. 2025. Dynamic knowledge integration for evidence-driven counter-argument generation with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22568–22584, Vienna, Austria. Association for Computational Linguistics.

Yuewei Yuan, Chaitanya Malaviya, and Mark Yatskar. 2023. AmbiCoref: Evaluating human and model sensitivity to ambiguous coreference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1023–1030, Dubrovnik, Croatia. Association for Computational Linguistics.

A Annotation Interface

Our annotation interface is shown in Figure 4.

B Mapping between predicates and all CA logical structures

We show the mapping between the original logical structures defined in CALSA and the predicates used in the work in Table 7.

C Similarity between selected evidence

For predicates that do not have a slot-filler, we report the Jaccard similarity (Jaccard(X,Y)=$\frac{|X \cap Y|}{|X \cup Y|}$) between all pairs of evidence for instances that have 2 more YES (if 3 annotators selected evidence, we compare every pair of them). We show the distribution of the similarity scores in Figure 5. Similar to slot-fillers, we observed that around 40% of the pairs have a similarity score above 0.5, which indicates that annotators tend to focus on similar segments in a CA as evidence for their answers.

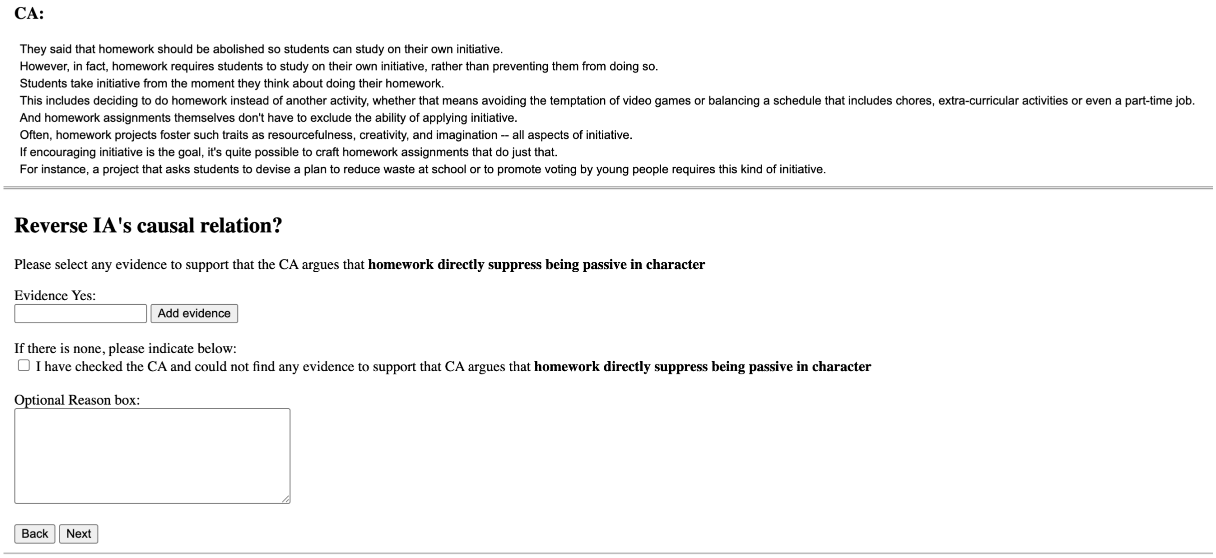

Figure 4: Our interface used for collecting predicate-level annotations for CAs.

Predicates	**Patterns of logical structure**
ACK(IA), MITI(Z,IA)	Mitigation
ACK(IA), ANO(Z,Y)	Alternative
NO-EVI(IA)	No evidence
DENY(IA), ANO(Z,Y)	Another true cause
TRANS1(X,Z,Y)	Missing mechanism #1
TRANS2(X,Z,Y)	Missing mechanism #2
NO-NEED-ADDRESS(Y)	No need to address
PRO(Y,Z)	Negative effect due to y
PRO(X,Z)	Positive effects of a different perspective from y #1
SUP(Y,Z)	Positive effects of a different perspective from y #2

Table 7: The mapping between the original patterns of CA logic structures defined in the CALSA paper (Naito et al., 2024) and the predicates explored in this paper. We ignore the predicate regarding the attack on the conclusion of IA in all logical structures, for the reason discussed in Section 3.2.

D Training and evaluation details

We use NVIDIA H200 GPU to train all the models 60 to 90 epochs till the overall rewards for both training and validation sets converge. We use Huggingface TRL library (von Werra et al., 2020) for training. We use DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) strategy to reduce memory usage. For testing our models, we use the checkpoint which has the highest overall reward on the validation set. We use the Huggingface evaluate library for calculating the ROUGE and BertScore scores.

E Examples of model generation

We show examples of Qwen2.5-7B-Instruct model's generation in Table 8.

F The distribution of logic structures in CALSA^{+} dataset

The distribution of the patterns of logic structure in CALSA^{+} dataset is shown in Figure 6. The x-axis represents the labels of logical structures and the y-axis represents the number of CAs.

G Reward function design

Our reward function rewards the format of the output and the generated contents. Specifically, we design two separate reward functions for rewarding the answer to the binary question and the slot-filler. For rewarding the answer to the binary question, first, if the answer is generated within the answer tag <answer></answer>, and the answer tag is generated only once, we give a reward of 0.5 for the

Input	Model generation
IA: homework reduces free time. free time is good. thus homework should be abolished.... **CA:** They said that if homework would be abolished, we could more free time. However, it is not important, in particular, to students. Because, not all students spend their free time wisely. It is supposed that most of students waste time in vain. As government team speaker said students are very busy because of club activities ,cram school and so on. Students are very tired, so they tend to play or something in free time. For example, net surfing, playing video games, and so on. As a result, most of students are lack of sleep because of meaningless things. **Q:** Does CA argue that 'homework' suppresses a bad thing? If yes, what is that thing? Please extract that from the counter-argument. (label: NO)	**Reasoning:** the counter-argument does not directly argue that 'homework' suppresses a bad thing, but rather it suggests that if homework were abolished, students might engage in more activities that are harmful or unproductive in their free time. the phrase that captures this idea is 'meaningless things', which could represent the bad thing that students might do without homework. **Answer:** yes **Slot-filler:** meaningless things
IA: homework reduces free time. free time is good. thus homework should be abolished. **CA:** They said that homework interferes with sleep because the student is doing homework at night. However, the only reason homework interferes with the student's sleep schedule is because the student procrastinates too much. If, as the argument states, the student doesn't begin homework until 10 PM, then this student has put off the important homework task and has instead filled the after-school hours with other less important activities. Doing away with homework because students have a habit of procrastinating just reinforces a bad habit that will cause problems in school and beyond. For instance, if a boss assigns some work to be completed after hours, the employee needs to make time to do this work or risk getting fired. In this sense, assigning homework teaches an important life skill: how to put important tasks first instead of putting them off until it's too late. **Q:** Does CA argue that 'free time' promotes a bad thing? If yes, what is that thing? Please extract that from the counter-argument. (label: YES)	**Reasoning:** the counter-argument does not directly state that 'free time' promotes a bad thing, but rather implies that a bad habit (procrastination) is the cause of the disruption to free time (or sleep, in this case). therefore, the answer to the question is no. **Answer:** no

Table 8: Examples of generation of Qwen2.5-7B-Instruct for the first fold. The generation is formatted for readability.

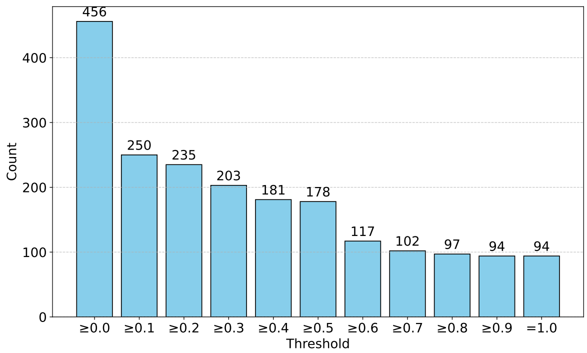

Figure 5: Number of pairs of evidence whose Jaccard Similarity score is above a certain threshold.

correctness of the format; otherwise, we give no reward. We then give a reward of 1 if the answer is correct; otherwise, 0. For rewarding the slot-filler, similar to the answer, we first reward the format. If the model generates a slot-filler within the tag <slot></slot>, and the tag is only generated once, we give a reward of 0.5; otherwise, 0. Then, for the content, we use the slot-fillers obtained from all three annotators as references, and calculate the ROUGE-1 score between the generated slot-filler and each reference slot-filler. We then aggregate the scores and use the final score as a reward. On top of that, if the current predicate does not have a slot-filler and the model generates one, we penalize the model by giving it a reward of -0.5.

H Examples of the procedure for decomposition

Consider an IA logic: X should be abolished, since X suppresses a good Y. Below, we show examples to explain the decomposition using two CA logical structures that attack this IA logic.

1. Mitigation, an original logic structure, is defined as: while it is not denied that X suppresses a positive outcome Y, the causal relationship can be mitigated through the means of Z. This logical structure has two interactions with the causal relation in the corresponding IA: (1) CA acknowledges IA's causal relation (2) while claiming that the relation can be mitigated by Z. The logical structure also has one interaction with the conclusion of the IA since CA's claim is that (3) X should not be abolished. Thus, this logical structure can be decomposed into three predicates (partial logics) (1) ACK(IA), (2) MITI(Z,IA), and (3) DENY(IA_CON). However, as all CAs must deny IA's conclusion, DENY(IA_CON) is not informative to distinguish different logical structures since it exists in every logical structure. Thus, we only consider ACK(IA) and MITI(Z,IA) as the constituent predicates for the logical structure Mitigation.

2. Similarly, Positive effects of a different perspective from y #1, another logical structure, is defined as: since X promotes a positive outcome Z, which is a different perspective from Y, X should not be abolished. The pattern has a total of two interactions with the original IA: (1) with the conclusion, since it claims that X should not be abolished, (2) with the variable X, since it claims that X promotes a positive Z. However, as aforementioned, interaction with conclusion is not informative enough for distinguishing different logical structures, since, as a valid CA, it must deny IA's conclusion, we only consider interaction (2) as the predicate PRO(X,Z) for this logical structure.

I RLVR modeling results per predicate

We show the RLVR modeling results per predicate for all predicates in Table 9. Note that for predicates that do not have a slot-filler by design, **Slot.<metric>.All** scores represent whether the model succeeds in not generating any slot-filler for the predicate (e.g., a score of 1.0 indicates that for all the instances of the current predicate, the model did not generate any unwanted slot-fillers). **Slot.<metric>.R** will be N/A for such predicates since the label is None.

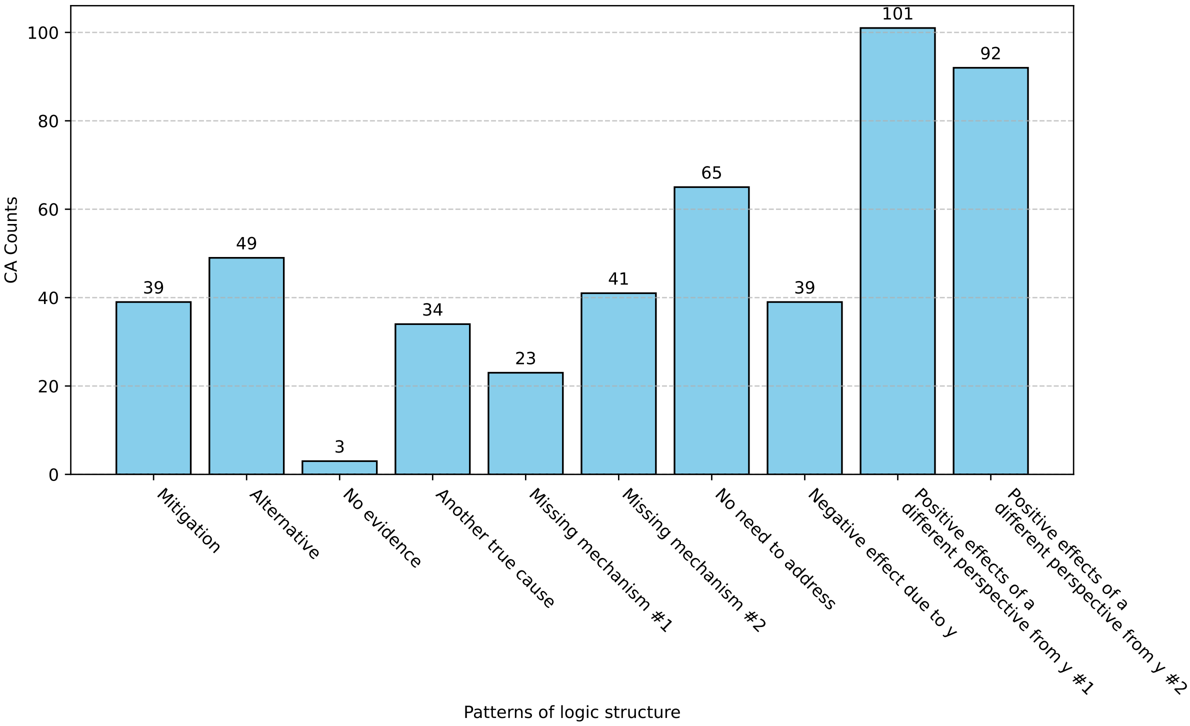

Figure 6: The distribution of patterns of logic structure in CALSA+ dataset.

Model	Predicate	Predicate.F1	Slot.ROUGE-1.All	Slot.BertScore.All	Slot.ROUGE-1.R	Slot. BertScore.R
Qwen2.5-7B-Instruct	ACK(IA)	.717 (.04)	1.0 (.00)	1.0 (.00)	N/A	N/A
	DENY(IA)	.741 (.03)	1.0 (.00)	1.0 (.00)	N/A	N/A
	REVERSE(IA)	.806 (.10)	1.0 (.00)	1.0 (.00)	N/A	N/A
	MITI(Z,IA)	.739 (.09)	.663 (.06)	.941 (.01)	.395 (.19)	.576 (.19)
	ANO(Z,Y)	.838 (.04)	.629 (.01)	.936 (.01)	.434 (.13)	.764 (.09)
	NO-EVI(IA)	.486 (.01)	1.0 (.00)	1.0 (.00)	N/A	N/A
	NO-NEED-ADDRESS(Y)	.539 (.21)	1.0 (.00)	1.0 (.00)	N/A	N/A
	SUP(Y,Z)	.698 (.05)	.809 (.16)	.971 (.02)	.198 (.27)	.386 (.11)
	PRO(Y,Z)	.705 (.07)	.722 (.08)	.951 (.01)	.258 (.13)	.412 (.17)
	PRO(X,Z)	.714 (.06)	.591 (.05)	.923 (.01)	.615 (.07)	.834 (.06)
	SUP(X,Z)	.698 (.05)	.469 (.12)	.913 (.01)	.477 (.18)	.837 (.09)
	TRANS1(X,Z,Y)	.767 (.05)	.793 (.03)	.967 (.00)	.263 (.02)	.520 (.15)
	TRANS2(X,Z,Y)	.772 (.09)	.613 (.07)	.935 (.01)	.266 (.10)	.608 (.08)
Mistral-7B-Instruct-v0.3	ACK(IA)	.673 (.14)	1.0 (.00)	1.0 (.00)	N/A	N/A
	DENY(IA)	.698 (.01)	1.0 (.00)	1.0 (.00)	N/A	N/A
	REVERSE(IA)	.796 (.03)	1.0 (.00)	1.0 (.00)	N/A	N/A
	MITI(Z,IA)	.684 (.10)	.692 (.02)	.943 (.00)	.362 (.10)	.515 (.11)
	ANO(Z,Y)	.758 (.04)	.552 (.08)	.927 (.02)	.396 (.08)	.705 (.11)
	NO-EVI(IA)	.486 (.01)	1.0 (.00)	1.0 (.00)	N/A	N/A
	NO-NEED-ADDRESS(Y)	.567 (.24)	1.0 (.00)	1.0 (.00)	N/A	N/A
	SUP(Y,Z)	.613 (.02)	.710 (.15)	.956 (.02)	.153 (.19)	.460 (.08)
	PRO(Y,Z)	.774 (.12)	.673 (.14)	.945 (.02)	.200 (.18)	.470 (.13)
	PRO(X,Z)	.763 (.08)	.645 (.08)	.934 (.01)	.644 (.05)	.840 (.02)
	SUP(X,Z)	.681 (.04)	.304 (.02)	.884 (.00)	.336 (.03)	.849 (.07)
	TRANS1(X,Z,Y)	.762 (.05)	.744 (.03)	.958 (.00)	.290 (.04)	.665 (.21)
	TRANS2(X,Z,Y)	.733 (.06)	.558 (.06)	.923 (.01)	.288 (.14)	.634 (.10)
Falcon3-7B-Instruct	ACK(IA)	.719 (.06)	1.0 (.00)	1.0 (.00)	N/A	N/A
	DENY(IA)	.745 (.06)	1.0 (.00)	1.0 (.00)	N/A	N/A
	REVERSE(IA)	.734 (.12)	1.0 (.00)	1.0 (.00)	N/A	N/A
	MITI(Z,IA)	.712 (.07)	.571 (.05)	.925 (.01)	.309 (.02)	.597 (.02)
	ANO(Z,Y)	.784 (.09)	.599 (.07)	.944 (.02)	.406 (.12)	.704 (.15)
	NO-EVI(IA)	.597 (.20)	1.0 (.00)	1.0 (.00)	N/A	N/A
	NO-NEED-ADDRESS(Y)	.624 (.08)	1.0 (.00)	1.0 (.00)	N/A	N/A
	SUP(Y,Z)	.692 (.09)	.769 (.08)	.961 (.01)	.271 (.12)	.515 (.07)
	PRO(Y,Z)	.756 (.03)	.676 (.03)	.943 (.01)	.231 (.07)	.551 (.10)
	PRO(X,Z)	.670 (.08)	.559 (.07)	.923 (.01)	.617 (.06)	.852 (.01)
	SUP(X,Z)	.687 (.10)	.435 (.13)	.909 (.01)	.395 (.08)	.786 (.06)
	TRANS1(X,Z,Y)	.602 (.10)	.637 (.05)	.943 (.00)	.174 (.09)	.476 (.32)
	TRANS2(X,Z,Y)	.671 (.01)	.536 (.05)	.922 (.01)	.299 (.07)	.579 (.09)

Table 9: Averaged 3-fold RLVR experiment results per predicate. The standard deviation is shown in parentheses.