Flee the Flaw: Annotating the Underlying Logic of Fallacious Arguments Through Templates and Slot-filling

Anonymous ACL submission

Abstract

Prior research in computational argumentation has mainly focused on scoring the quality of arguments, with less attention on explicating logical errors. In this work, we introduce four sets of explainable templates for common informal logical fallacies designed to explicate a fallacy's implicit logic. Using our templates, we conduct an annotation study on top of 400 fallacious arguments taken from LOGIC dataset and achieve a high agreement score (Krippendorf's α of 0.54) and reasonable coverage (0.829). Finally, we conduct an experiment for detecting the structure of fallacies and discover that state-of-the-art language models struggle with detecting fallacy templates (0.31 F_1). To facilitate research on fallacies, we make our dataset publicly available.

1 Introduction

A *fallacy* is an invalid or weak argument supported by unsound reasoning (Hinton, 2020). The automatic detection of fallacies has important applications, including providing constructive feedback to learners in writing. The assessment of argument quality, including fallacy detection, is considered an important topic in the fields of computational argumentation and argumentation mining (Wachsmuth et al., 2017; Ke and Ng, 2019).

Previous work on quality assessment has focused on numerical scoring (Carlile et al., 2018; Ke et al., 2019) and fallacy type-labeling tasks (Jin et al., 2022; Sourati et al., 2023), without aiming to analyze *fallacy logic structures*, namely the representation of *how* given arguments are weak. In the field of argumentation theory, a typology of invalid arguments has been long studied and compiled into an inventory (Walton, 1987; Bennett, 2012). The inventory typically includes semi-formal definitions and some examples for each type of fallacy. For example, *Faulty Generalization* is a widely recognized fallacy type, characterized by "Drawing a conclusion based on a small sample size, rather than looking at statistics that are much more in line with the typical or average situation." (Bennett, 2012). The semi-formal definition is as follows: "(i) Sample S is taken from population P. (ii) Sample S is a very small part of population P. (iii) Conclusion C is drawn from sample S and applied to population P". Although such inventory provides insights into how the analysis of fallacy logic structure can be formulated as an NLP task, several important questions remain: (i) How should the annotation scheme for fallacy logic structure identification be designed? (ii) Can humans consistently annotate fallacy logic structures? (iii) Is the automatic identification of fallacy logic structure a challenging task for machines?

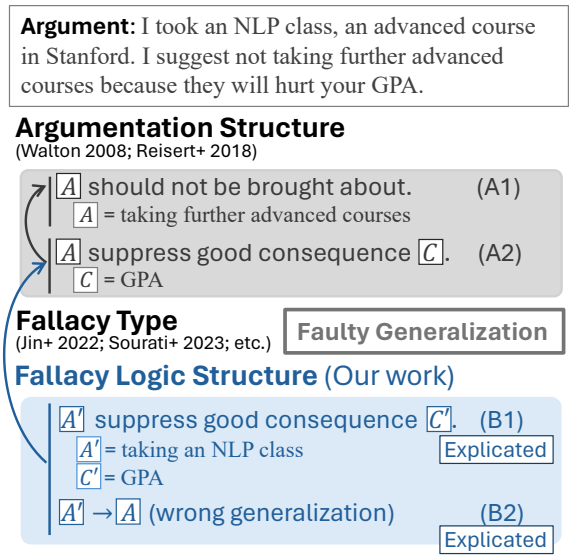

Figure 1: An overview of the proposed fallacy logic structure identification task.

To address this issue, we propose *fallacy logic structure identification*, a new task of identifying the underlying logic structure of fallacies. Specifically, we design an annotation scheme for this task and conduct an annotation study to examine

its feasibility. The key idea behind our annotation scheme is as follows. Capturing the fallacy logic structure of arguments requires two types of representations: (A) core argumentation structure and (B) which argumentative component is fallacious in what manner. Consider the argument in Fig. 1, where the writer persuades people *not* to take advanced courses in Stanford (A2) because it will hurt their GPA (A1). To represent this core argumentation structure, we employ Walton et al. (2008)'s Argumentation Schemes, a well-known typology of everyday arguments (it falls under *Argument from Consequence*). Now, A1, the universal claim made towards people, is further supported by the writer's own single experience based on his NLP class (B1). This is a faulty generalization, where the writer *implicitly* assumes that his single experience can be generalized to everyone (B2). To represent this fallacy logic structure, we leverage an inventory of logical fallacies developed in the field of argumentation theory.

Our main contributions are as follows:

- We are the first to formulate an inventory of logical fallacies as fallacy templates and conduct an extensive annotation study on top of 400 fallacious arguments, yielding high inter-annotator agreement (Krippendorf's α of 0.54) and coverage of 0.828 (§3).
- We annotate 400 arguments from LOGIC (Jin et al., 2022) with fallacy logic structures and publicly release the corpus (§3).[1] This is the first corpus of fallacy logic structures including implicit components.
- Our experiments show that fallacy logic structure identification poses a significant challenge for state-of-the-art language models like GPT-3.5 and GPT-4 (§4).

2 Fallacy Logic Structure

2.1 Template-based Formulation

The underlying logical structure of arguments has been represented previously with *argument templates* (Reisert et al., 2018).

- **Premise**: If $[A]$ is brought about, *GOOD* (*BAD*) consequences $[C]$ will plausibly occur.
- **Conclusion**: Therefore, $[A]$ should (not) be brought about.

[1] https://anonymous

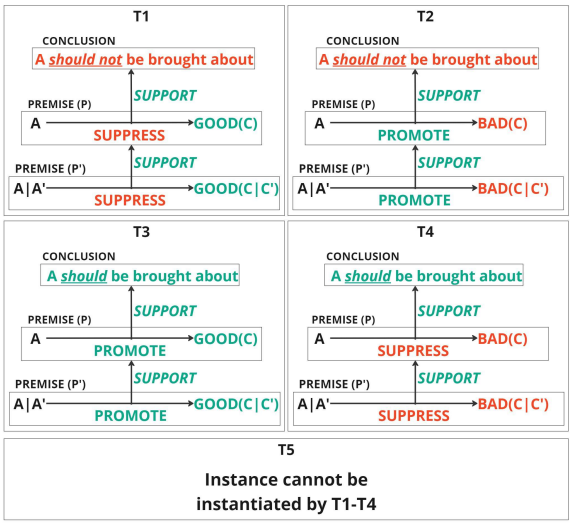

Figure 2: Five distinct templates designed for annotating the *faulty generalization* fallacy's logical structure.

Argument templates are annotation-friendly templates based on Walton et al. (2008)'s Argument from Consequence scheme, shown above, where $[A]$ and $[C]$ represent event/entity slot-fillers, and *GOOD*/*BAD* represent the sentiment of slot-fillers. In addition, two relations, *PROMOTE* and *SUPPRESS*, between slot-fillers are considered. *PROMOTE* refers to triggering the consequence and *SUPPRESS* refers to preventing the consequence (Hashimoto et al., 2012).

Consider the *faulty generalization* argument in Fig. 1. With argument templates, an instantiation could be represented with slot-fillers $[A]$="taking further advanced courses" and $[C]$="GPA", conclusion="$[A]$ should not be brought about", and premise (P)="$[A]$ *SUPPRESS* $[C]$". Such argument templates are a simple, efficient way to represent underlying logic.

2.2 Our Fallacy Template Inventory

For representing fallacy logical structure, we extend Reisert et al. (2018)'s argument templates with new argumentative components. In total, we create 20 new templates (4 fallacy types, 5 templates each) for the task of fallacy structure parsing. Fig. 2 shows an example of all fallacy templates we create for *faulty generalization* arguments.[2].

For the argument in Fig. 1, we can use the first template in our *Hasty Generalization* inventory for capturing the underlying fallacious structure. Namely, the argument is fallacious with the ad-

[2] Due to space limitations, we have included all of our fallacy templates in the Appendix.

Fallacy Type	GWET AC1	Krippendorff's α
False Dilemma	0.628	0.435
Faulty Generalization	0.395	0.360
False Causality	0.710	0.653
Fallacy of Credibility	0.578	0.491
Average	0.569	0.536

Table 1: Template selection Inter-Annotator Agreement.

Fallacy Type	Annotator 1	Annotator 2
False Dilemma	0.900	0.910
Faulty Generalization	0.680	0.760
False Causality	0.950	0.960
Fallacy of Credibility	0.640	0.828
Average	0.793	0.828

Table 2: Coverage of fallacy templates.

Fallacy Type	Disagree Rate
False Dilemma	0.32
Faulty Generalization	0.49
False Causality	0.24
Fallacy of Credibility	0.35

Table 3: Analysis of disagreement.

dition of premise P', where P'="[A] *SUPPRESS* [C]" with slot-fillers [A']="taking an NLP class" and [C]="GPA", which supports P.

3 Flee the Flaw (FtF) Dataset

We discuss the creation of our dataset *Flee the Flaw*, hereby referred to as *FtF*. We first use LOGIC, an existing dataset of annotated fallacious arguments, and build our corpus on top of it.

3.1 LOGIC Dataset

To build a dataset of fallacious argument template instantiations, we require fallacious arguments which cover our target fallacy types. Therefore, we use LOGIC (Jin et al., 2022), an English fallacy dataset consisting of 2,449 fallacious arguments spanned across multiple fallacy types, including our four target template types. We sampled 400 arguments from LOGIC, equally split between its development (LOGIC-DEV$_{200}$) and training sets (LOGIC-TRAIN$_{200}$), with 200 arguments each. Missing fallacy instances in the development set were supplemented from the training set, ensuring no overlap by segmenting the training set before distribution.

3.2 FtF Dataset

We utilized 400 instances, with 200 samples sourced from the LOGIC-DEV$_{200}$ set to establish annotation guidelines and another 200 samples from the LOGIC-TRAIN$_{200}$ set, utilizing these guidelines. As a result, we have allocated 200 instances for each set, henceforth designated as the 'Few-Shot-Example-Set' (FTF-TRAIN) and the 'Test-Sample-Set' (FTF-DEV), respectively.

Annotation Process Given a fallacious argument, its fallacy type, and our templates, annotators selected the appropriate template and slot-fillers. Annotators provided their confidence level for each annotation and comments, if necessary.

Annotation Quality To check the quality of the FtF dataset, we performed Inter-Annotator Agreement (IAA) Analysis and coverage assessments on the selected templates. The IAA, depicted in Table 1, confirms a respectable consensus among annotators using our proposed template. GWET AC1 (Gwet, 2008) scores from 0.395 to 0.710 indicate moderate to the substantial agreement. Krippendorff's alpha (Hayes and Krippendorff, 2007) supports these observations, with similar scores indicating consistent annotator reliability.

Table 2 provides a comparison of annotation coverage between two annotators. The coverage score reflects how comprehensively each annotator has identified and applied the appropriate annotation template to instances of fallacies.

3.3 Disagreement Analysis

After implementing the annotation guidelines, we noted discrepancies between two annotators, as depicted in Table 3, which outlines the levels of disagreement in FtF dataset. *Faulty Generalization* recorded the highest disagreement rate at 49%, as certain dataset instances could be interpreted as different fallacy types. For example, the statement "James, the company you work for just filed for bankruptcy! How can I trust you with our money?" may be categorized as a *Fallacy of Division*, although the intended label was *Faulty Generalization*. This ambiguity has contributed to the elevated disagreement rates in template instantiation.

Secondly, we observe that the *Faulty Generalization* fallacy has the lowest IAA. So, we conduct an additional analysis on all disagreements for *Faulty Generalization* and discover that 60% of disagreements were caused when one annotator labeled '5'

and the other instantiated a template, where reasons annotators labelled '5' was due to complicated instances and implicitness of the argument. Lastly, some instances in LOGIC were found to be other types of fallacies, namely *Slippery Slope*.

4 Experiments

We explore the effectiveness and adaptability of LLMs in fallacy structure parsing.

4.1 Setup

We use GPT-3.5-turbo (Abdullah et al., 2022) and GPT-4-1106-preview (Achiam et al., 2023), a state-of-the-art language model, on zero-shot, 1-shot, and 5-shot prompt settings. We ran this experiment one time for each setting. For data, we use FTF-TRAIN to sample a few-shot demonstrations for few-shot prompting and FTF-DEV as a test set. We use Macro F1 for template section. For slot-filling, we use (i) Partial Match accuracy, the percentage of test instances where *all* predicted slot fillers have more than 50% word overlap with the gold standard, (ii) Exact Match Accuracy, the percentage of test instances where *all* predicted slot fillers perfectly match the gold-standard slot fillers.

4.2 Results and Analysis

Template Selection Table 4 shows our results of the template selection. Both models achieved the highest overall results when using a 5-shot prompt setting. The trend of the results for the prompt settings appears to improve with each shot. However, GPT-3.5 yielded the best results for *false causality*, and GPT-4 performed the best for *faulty generalization* in the 1-shot prompt setting. Nevertheless, for 3 out of 4 types, the best results were obtained using a 5-shot. This indicates the importance of providing examples for template instantiation.

Slot-Filling Table 5 shows the results of the performance of slot-filling. The 1-shot prompt setting for GPT-3.5 achieved the highest accuracy for Exact Match and Partial Match. Both models exhibited good performance in matching slot-filler gold labels for the *false dilemma* and *faulty generalization*. However, they struggled when dealing with *false causality* and the *fallacy of credibility* types, where most of the results were below 0.50. It remains a question why both models, across all prompt settings, were unable to perform better for *false causality* and the *fallacy of credibility*, especially considering that instances of those two types are straightforward.

Model	FD	FG	FC	FCr	Overall
Random	0.20	0.20	0.20	0.20	0.20
Majority	0.21	0.09	0.11	0.10	0.10
GPT3.5 zero	0.05	0.22	0.25	0.18	0.18
GPT3.5 1-shot	0.05	0.24	**0.33**	0.20	0.20
GPT3.5 5-shot	0.23	**0.37**	0.26	0.39	**0.31**
GPT4 zero	0.11	0.24	0.13	0.30	0.20
GPT4 1-shot	0.13	0.33	0.18	0.33	0.24
GPT4 5-shot	**0.27**	0.32	0.22	**0.45**	**0.31**

Table 4: Performance of template selection (Macro F1) for False Dilemma (FD), Faulty Generalization (FG), False Causality (FC), and Fallacy of Credibility (FCr).

Model	FD	FG	FC	FCr	Overall
Exact Match:					
GPT3.5 zero	0.75	0.50	0.08	0.21	0.39
GPT3.5 1-shot	**0.80**	**0.78**	0.20	0.36	**0.53**
GPT3.5 5-shot	0.64	0.67	0.37	0.32	0.50
GPT4 zero	0.67	0.47	0.11	0.40	0.41
GPT4 1-shot	0.75	0.50	**0.38**	**0.41**	0.51
GPT4 5-shot	0.50	0.44	0.20	0.26	0.35
Partial Match:					
GPT3.5 zero	0.75	0.50	**0.50**	0.21	0.49
GPT3.5 1-shot	**1.00**	**0.78**	0.47	0.43	**0.67**
GPT3.5 5-shot	0.91	0.71	0.37	0.32	0.58
GPT4 zero	**1.00**	0.53	0.33	**0.50**	0.59
GPT4 1-shot	0.75	0.50	0.50	0.45	0.55
GPT4 5-shot	0.56	0.50	0.33	0.39	0.45

Table 5: Performance of slot filling.

5 Conclusion and Future Work

We have formulated an inventory of logical fallacies as fallacy templates and conducted an extensive annotation study on top of 400 fallacious arguments. We have constructed the first, publicly avaialble corpus of fallacy logic structures-annotated arguments. Our experiments show that the fallacy logic structure identification task poses a significant challenge for state-of-the-art language models, highlighting future automation challenges.

Clearly, our dataset holds numerous possibilities beyond the scope of Inter-Annotator Agreement (IAA) analysis. Our immediate next step involves studying the underlying patterns and reasoning errors in arguments by analyzing the logical structure of fallacies. We also plan to conduct a large-scale annotation of a fallacy template on larger and more natural arguments.

Limitations

In this research, we mainly focus on the proposed explainable fallacy template with a focus on only 4 fallacy types which are mainly the informal fallacy while leaving behind the fallacy of logic which is the extension from the informal fallacy to formal fallacy. In addition, our fallacy templates do not cover every possible combination of ingredients (e.g. *NOT PROMOTE*, *NOT SUPPRESS*) which limits the amount of instantiations we can acquire. Furthermore, we use patterns inspired by Walton (2008)'s Argument from Consequence scheme, which also limits the full range of fallacy instantiations we can produce.

We limit ourselves to four types of fallacies which only represents a small subset of all known fallacies. Primarily, we target common informal logical fallacies as a start for fallacious template structure instantiation.

Regarding our experiments, we only experiment with two LLMs: GPT4 and GPT3.5.

Given the structure of *False Dilemma* fallacy, which follows an *either-or* structure, we obtain an unbalanced partition for our false dilemma templates. As shown in Fig. 10, both annotators mainly annotated with template 2.

Ethical Considerations

Each author of this paper ensured that all ethical considerations were upheld. All results are reported as accurately as possible. Given that we conducted an annotation, we adhere to constructing a high quality dataset as exemplified by our annotator agreement results.

References

Malak Abdullah, Alia Madain, and Yaser Jararweh. 2022. Chatgpt: Fundamentals, applications and social impacts. In *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–8. IEEE.

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

B. Bennett. 2012. *Logically Fallacious: The Ultimate Collection of Over 300 Logical Fallacies*. Ebookit.com.

Winston Carlile, Nishant Gurrapadi, Zixuan Ke, and Vincent Ng. 2018. Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, Melbourne, Australia. Association for Computational Linguistics.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jong-Hoon Oh, et al. 2012. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 619–630.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

Martin Hinton. 2020. *Evaluating the Language of Argument*, 1 edition, volume 37. Springer Cham.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schoelkopf. 2022. Logical fallacy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7180–7198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zixuan Ke, Hrishikesh Inamdar, Hui Lin, and Vincent Ng. 2019. Give me more feedback II: Annotating thesis strength and related attributes in student essays. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3994–4004, Florence, Italy. Association for Computational Linguistics.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

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.

Zhivar Sourati, Filip Ilievski, Hông-Ân Sandlin, and Alain Mermoud. 2023. Case-Based Reasoning with Language Models for Classification of Logical Fallacies.

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.

D.N. Walton. 1987. *Informal Fallacies: Towards a Theory of Argument Criticisms*. Companion series. J. Benjamins Publishing Company.

Douglas Walton. 2008. *Informal logic: A pragmatic approach*. Cambridge University Press.

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

A Appendix

A.1 Templates

Fig. 2 is a list of *faulty generalization* templates. The fallacy occurs because applying a belief to a large population without having sufficient sample and non-biased. For example, "I know five people from Kentucky. They are all racists. Therefore, Kentuckians are racist."

Fig. 7 is a list of *false causality* templates. The fallacy occurs when assuming two events are correlated, they must also have a cause-and-effect. For example, "I drank bottled water and now I am sick, so the water must have made me sick."

Fig. 8 is a list of *fallacy of credibility* templates. The fallacy occurs when an appeal is made to some form of ethics, authority, or credibility. For example, "We are going to protest and not get in trouble because Mr. Iglesias said it is okay."

Fig. 9 is a list of *false dilemma* templates. The fallacy occurs due to restrictions on the available choices without considering any potential options. For example, "We either have to cut taxes or leave a huge debt for our children."

A.2 Guideline Creation Process

First, two expert annotators in the field of argumentation, both authors of this paper, conducted an annotation study on top of LOGIC-DEV$_{200}$. The annotation was divided into multiple rounds in which agreements were calculated and disagreements were discussed amongst both annotators. After the first round, an initial set of guidelines were created. The guidelines were then discussed and updated after each subsequent round until all 200 instances in LOGIC-DEV$_{200}$ were annotated. All notes during each round were aggregated to create our final guidelines.[3]

A.3 Reducing Annotation Complexities

During guideline construction, annotators discovered that multiple templates could be instantiated for a single argument. In order to reduce the complexity of annotation, many important conditions were created, such as i) *preservation of argument's original, explicit intent*, ii) *paraphrase arguments in terms of Argument from Consequences*, and iii) *preference of entities over events.* We demonstrate such conditions with the following example of False Dilemma: "*We either have to cut taxes or leave a huge debt for our children.*".

[3]Our guidelines our available at `http://anonymous`

Opposed to selecting the entity A="*taxes*" which satisfies the third condition, annotators were encouraged to select the event A="*Cut taxes*" as it maintains the explicit intention of the argument, satisfying the first condition. Given that this is a *false dilemma* fallacious argument which follows an *either-or*, the annotators satisfied the second condition by considering that the argument can be thought of in terms of argument from consequence, where the conclusion "Cut taxes should be brought about" is good as it suppresses the premise "leave a huge debt for our children", a bad thing.

A.4 False Dilemma Example

An example of a false dilemma is shown in Fig. 3.

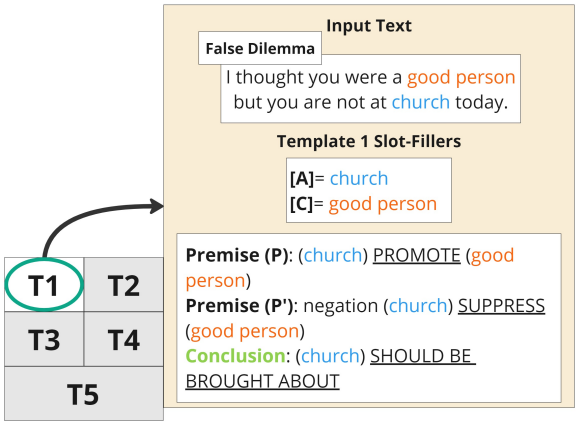

Figure 3: Template instantiation for a *false dilemma* argument. The instantiation represents the underlying logical structure of the fallacy.

A.5 Faulty Generalization Example

An example of a faulty generalization is shown in Fig. 4.

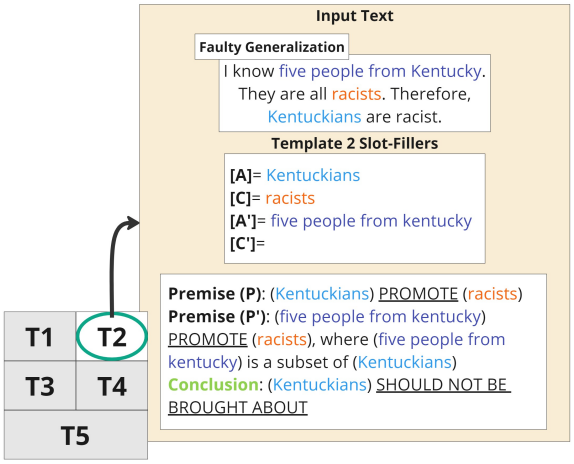

Figure 4: Template instantiation for a *faulty generalization* argument. The instantiation represents the underlying logical structure of the fallacy.

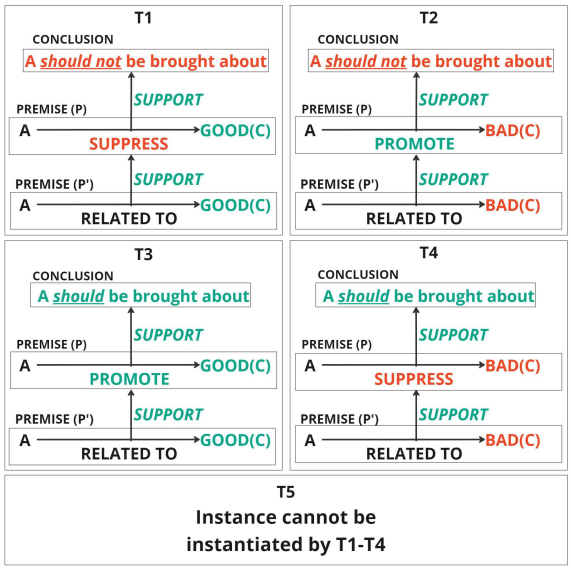

Figure 7: Five distinct templates designed for annotating the *false causality* fallacy's logical structure.

A.6 False Causality Example

An example of a false causality is shown in Fig. 5.

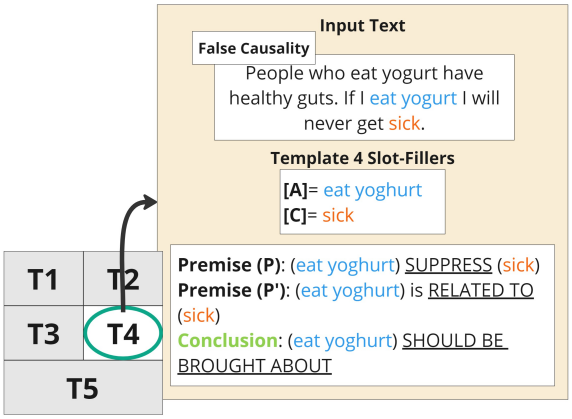

Figure 5: Template instantiation for a *false causality* argument. The instantiation represents the underlying logical structure of the fallacy.

A.7 Fallacy of Credibility Example

An example of a fallacy of credibility is shown in Fig. 6.

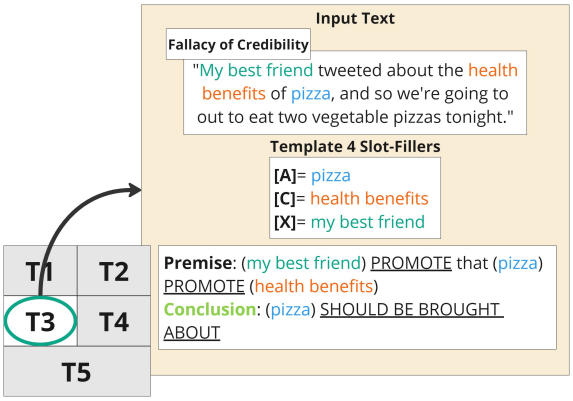

Figure 6: Template instantiation for a *fallacy of credibility* argument. The instantiation represents the underlying logical structure of the fallacy.

A.8 Distribution of fallacy templates

Fig. 10 shows the distribution of fallacy templates.

A.9 Prompt

Table 6 shows the prompt of zero-shot, one-shot, and five-shot for the LLM experiments. The instances that were used for one-shot and five-shot prompts are randomly selected from LOGIC-TRAIN$_{200}$.

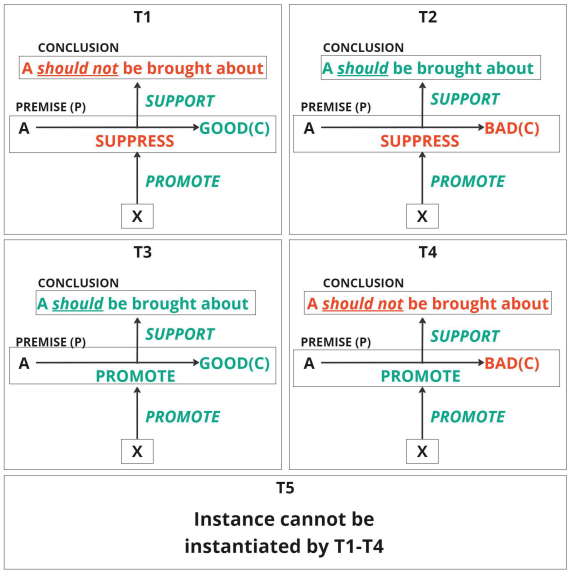

Figure 8: Five distinct templates designed for annotating the *fallacy of credibility* fallacy's logical structure.

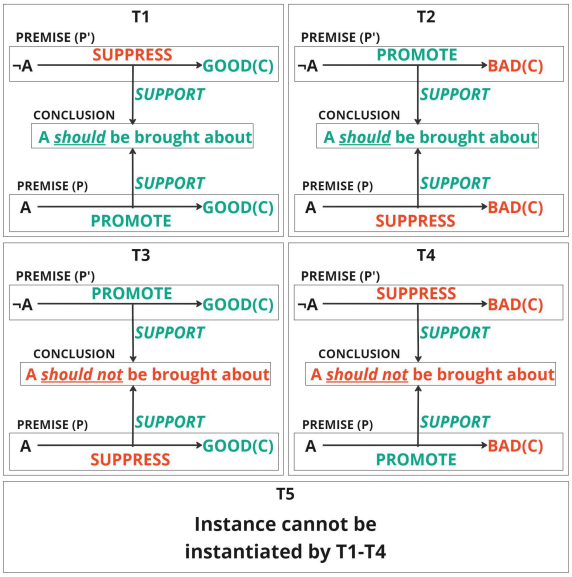

Figure 9: Five distinct templates designed for annotating the False Dilemma fallacy's logical structure.

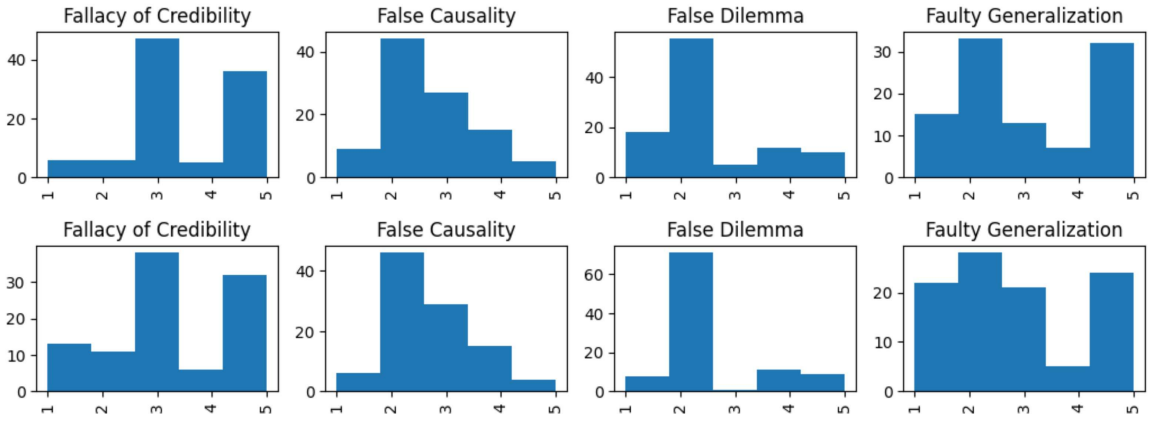

Figure 10: The distribution of fallacy templates in our final dataset, as provided by two annotators (annotator 1 above and annotator 2 below), encompasses 400 instances, with each fallacy type represented by 100 instances.

Zero-shot	One-shot	Five-shot
# Task Identify the underlying structure of an argument of False Causality. Given a list of fallacy templates, your task is to choose a template that best describes the underlying fallacy structure, filling the template's placeholders. Please follow the Output Format!!! # List of Templates Template No.1: Premise 1: An entity/action [A] suppresses a good entity/action [C]. Premise 2: An entity/action [A] is related to an entity/action [C]. This premise supports Premise 1. Conclusion: Premise 1 supports that [A] should not be brought about. Template No.2: Premise 1: An entity/action [A] promotes a bad entity/action [C]. Premise 2: An entity/action [A] is related to an entity/action [C]. This premise supports Premise 1. Conclusion: Premise 1 supports that [A] should not be brought about. Template No.3: Premise 1: An entity/action [A] promotes a good entity/action [C]. Premise 2: An entity/action [A] is related to an entity/action [C]. This premise supports Premise 1. Conclusion: Premise 1 supports that [A] should be brought about. Template No.4: Premise 1: An entity/action [A] suppresses a bad entity/action [C]. Premise 2: An entity/action [A] is related to an entity/action [C]. This premise supports Premise 1. Conclusion: Premise 1 supports that [A] should be brought about. Template No.5: There is either no consequence in the argument, or the argument cannot be instantiated with one of the templates above. # Output Format Template No.=[No.] $[A]$= $[C]$= # Query {}	# Zero-shot prompt # Example I had a real bad headache, then saw my doctor. Just by talking with him, my headache started to subside, and I was all better the next day. It was well worth the $200 visit fee. Template No.=4 $[A]$=talking with him $[C]$=headache # Query {}	# Zero-Shot prompt # Example1 You oversleep and then fail a test; so you assume that oversleeping causes you to fail tests" Template No.=1 $[A]$=oversleep $[C]$=test # Example2 The accident was caused by the taxi parking in the street Template No.=2 $[A]$=taxi parking in the street $[C]$=accident # Example3 I have flipped heads five times in a row. As a result, the next flip will probably be tails. Template No.=3 $[A]$=flipped heads five times in a row $[C]$=next flip will probably be tails # Example4 I had a real bad headache, then saw my doctor. Just by talking with him, my headache started to subside, and I was all better the next day. It was well worth the $200 visit fee. Template No.=4 $[A]$=talk with doctor $[C]$=headache # Example5 â€œSince event Y followed event X, event Y must have been caused by event Xâ€. What fallacy is described in this logical form? Template No.=5 $[A]$= $[C]$= # Query {}

Table 6: The prompts for the LLM experiments