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

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

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 83%. 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 (0.47 accuracy). To facilitate research on fallacies, we make our dataset, guidelines, and code 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., 2023a), 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

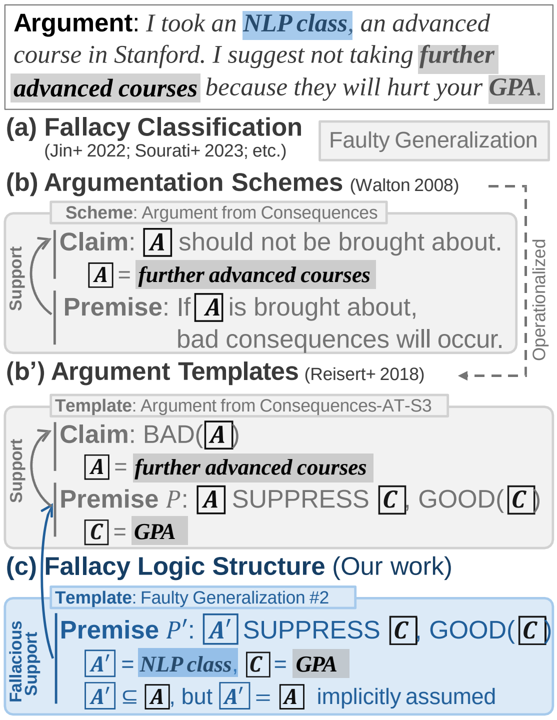

Figure 1: Overview of our proposed fallacy logic structure. We extend (b') existing argumentative representation (Reisert et al., 2018) consisting of Claim and Premise P by adding (c) Premise P', which explains what makes the argument fallacious. The example annotation shows: (i) the claim "(A=*further advanced courses*) are BAD" is supported by "P: (A=*further advanced courses*) SUPPRESS (C= *GPA*), a GOOD thing", and (ii) P is then further supported by "P': (A'=*NLP class*) SUPPRESS (C=*GPA*), a GOOD thing", where A'=*NLP class*(their own experience) is implicitly generalized to A=*further advanced courses*(advanced courses in general), which makes the overall argument fallacious.

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)

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 20524–20540
November 12-16, 2024 ©2024 Association for Computational Linguistics

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) To what extent is the automatic identification of fallacy logic structure a challenging task for machines?

To address this issue, we propose *fallacy logic structure identification*, a new task for identifying the underlying logical structure of fallacies. For this task, we design an annotation scheme and conduct an annotation study to examine its feasibility. The key idea behind our annotation scheme is to enrich previous work on the argumentative structure with a fallacy structure from an inventory of common fallacy types.

Consider the argument in Fig. 1, where the writer persuades people *not* to take advanced courses at Stanford because they claim it will hurt their GPA. The claim is further supported by the writer's own, single experience based on their NLP class. This is a faulty generalization caused by the writer *implicitly* assuming that their single experience can be generalized to everyone. Previous work in fallacy identification (Sourati et al., 2023b; Jin et al., 2022) would identify this argument as *Faulty Generalization* (Fig. 1 (a)), but no additional information such as logical structure or fallacious reasoning is provided. Argumentation Schemes (Walton et al., 2008), a well-known typology for the representation of arguments, would categorize this argument as *Argument from Consequences* (Fig. 1 (b)), and Reisert et al. (2018)'s *Argument Templates*, an operationalized version of Argumentation Schemes, represent this argument with a more fine-grained, logical representation by structured templates (Fig. 1 (b')). To represent the committed fallacy structure, our work further enriches this representation by adding an additional premise that indicates how the given argument is fallacious (Fig. 1 (c)).

Our main contributions are as follows:

- We conduct the first study of formulating logical fallacy structure by creating an inventory of fallacy templates (§3).
- We create the first dataset of fallacy logical structures which consists of 400 arguments from LOGIC (Jin et al., 2022) annotated with our templates (§4). We publicly release the dataset, along with the guidelines and code.[1] Our dataset achieves high inter-annotator agreement (Krippendorf's α of 0.54) and coverage (0.83).
- We show that the fallacy logic structure identification task poses a significant challenge for state-of-the-art language models (§5).

2 Related Work

Fallacy Annotation Several studies address creating benchmarks for fallacy identification, including game facilitation (Habernal et al., 2017) and argumentation corpora validation (Ruiz-Dolz and Lawrence, 2023). Particularly, Jin et al. (2022) focused on logical fallacies within climate change discourse, emphasizing the challenges posed by complex scientific data. They developed detailed annotation guidelines to aid in consistent identification of fallacies within climate arguments. Similarly, Goffredo et al. (2023) analyzed fallacious reasoning in U.S. presidential debates, highlighting common fallacies. They employed advanced computational techniques and the INCEpTION platform for structured annotation, ensuring reliability through cross-verification and Krippendorff's α. In addition to the current benchmarks, our research proposes benchmark resources aimed at capturing fallacy structure rather than solely identifying fallacies. Our research fills the gap, extending previous work by focusing on template annotation to capture the underlying structure of fallacious arguments.

Argumentation Structure Argumentation theory examines how arguments, including those about daily exercise, are constructed and evaluated. To begin with, Stab and Gurevych (2017) established methods for parsing argumentation structure in persuasive essays by identifying and classifying argument components and their relationships. Toulmin (2003) provided a framework for analyzing arguments by decomposing them into components like Claim, Grounds, Warrant, and Rebuttal. Walton (2013) focused on specific argumentation schemes, such as Argument from Analogy, which compares similar situations to infer outcomes but risks failure with irrelevant similarities (false analogy). The Argument from Consequences (Walton et al., 2008) scheme emphasizes potential outcomes of actions, often involving causality and appeals

[1] https://github.com/irfanrob/fallacy-template/tree/main

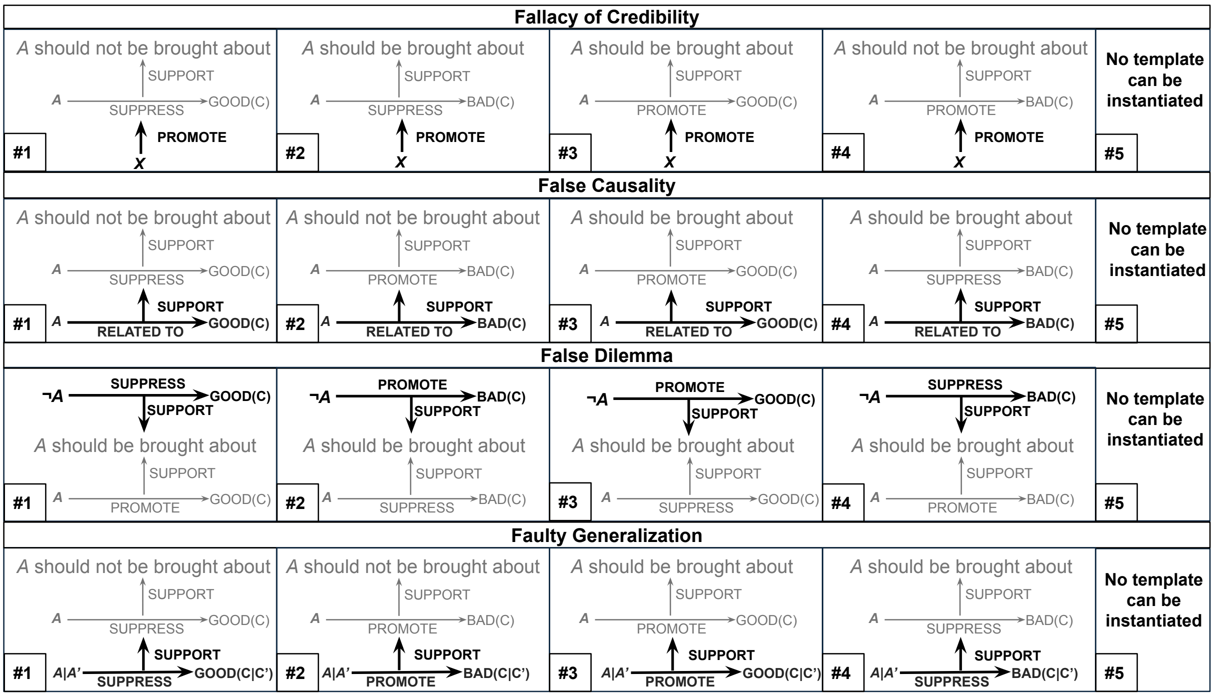

Figure 2: Our templates for annotating fallacious argument logical structure. We extend upon existing work (Walton et al., 2008; Reisert et al., 2018), consisting of a conclusion (i.e., *A should (not) be brought about*) and supporting premise, by adding an additional supporting premise in bold which represents the committed fallacy logical structure.

to consequences. Evaluating it requires considering 1) the connection between action and consequence, 2) the quality of supporting evidence, and 3) whether opposing consequences have been addressed. Building on prior work on argument structure, particularly the Argument from Consequences scheme (a frequently used scheme by Walton), this research addresses a gap by using argument templates, inspired by Reisert et al. (2018) to capture the structure of fallacies within this scheme. This choice is motivated by the scheme's frequent use and its potential for revealing fallacious arguments. Building on this potential, and inspired by Reisert et al. (2018)'s templates, we address a gap by using templates to capture the structure of fallacies within the Argument from the Consequence scheme. Previous work on Argument from Consequences demonstrates high coverage in annotation efforts, further supporting this approach.

3 Fallacy Logic Structure

3.1 Design Principles

To develop an annotation scheme for fallacy logic structure, we adhere to three key criteria.

First, we require the annotation to be able to explain the underlying structure of fallacy. We extend the existing representation of arguments (Fig. 1 (b')) by an additional premise attached with an explanation as to why it fallaciously supports the original premise (Fig. 1 (c)).

Second, our annotation scheme must cover the majority of instances of common fallacy types. We focus on the fallacies most commonly studied in computational argumentation, such as those in Alhindi et al. (2023) and Helwe et al. (2023), whose statistics on fallacy types guide our template design to match the most frequent occurrences. We develop 20 new templates covering four defective induction fallacy types: Fallacy of Credibility, False Causality, False Dilemma, and Faulty Generalization. An example and more detailed explanation regarding the types can be found in Section A.2.

Third, our annotation scheme must utilise Reisert et al. (2018)'s template selection and slot-filling approach further simplifying annotation while remaining computationally friendly. As inspired by the Argument from Consequences and employing Reisert et al. (2018)'s work as a base scheme, the template design captures both positive and negative consequences within the scheme. This results in two templates for each consequence type, along with a template addressing instances that cannot be directly covered. This approach aims to provide rich information about fallacy structures while simplifying the annotation process.

3.2 Representation of Core Arguments

The underlying structure of arguments has been represented previously with Walton et al. (2008)'s Argumentation Schemes, a set of roughly 60 schemes which provide structure between argumentative components such as a conclusion (i.e., claim) and premise. An example of a common scheme, Argument from Negative Consequences, is as follows[2]:

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

Here, A is a placeholder (i.e., slot-filler) that represents an *action* and P supports conclusion. For the argument in Fig. 1, we represent slot-filler [A]="further advanced courses".

Towards operationalizing Walton et al. (2008)'s Argumentation Schemes into more fine-grained logical representations, Reisert et al. (2018) developed *argument templates*, an inventory of annotation-friendly templates consisting of ingredients such as placeholders. An example of an argument template built on top of Argument from Negative Consequences scheme is as follows:

- **Premise (P):** [A] *SUPPRESS* a *GOOD* [C].
- **Conclusion**: [A] is *BAD*.

Both A and C represent *action* and *consequence* placeholders, respectively. *GOOD* and *BAD* represent the sentiment of each placeholder, and *SUPPRESS* represents the relation between A and C, where *SUPPRESS* refers to preventing the consequence (Hashimoto et al., 2012). Revisiting the argument in Fig. 1, we can instantiate the argument template with A="further advanced courses" and C="GPA". Such argument templates are a simple, efficient way to represent underlying logic.

As shown for Faulty Generalization fallacies in Fig. 2, argument templates were handcrafted to allow for both Argument from Positive Consequence (A *should be brought about*) and Argument from Negative Consequence (A *should not be brought about*) with a supporting P' (grey) consisting of positive (e.g., A PROMOTE GOOD(C)) and negative (e.g., A SUPPRESS GOOD(C)) consequences, respectively, where *PROMOTE* refers to the triggering of the consequence (Hashimoto et al., 2012). We build on top of this for adding logical structure for fallacies.

[2]For readability, we represent placeholders in brackets.

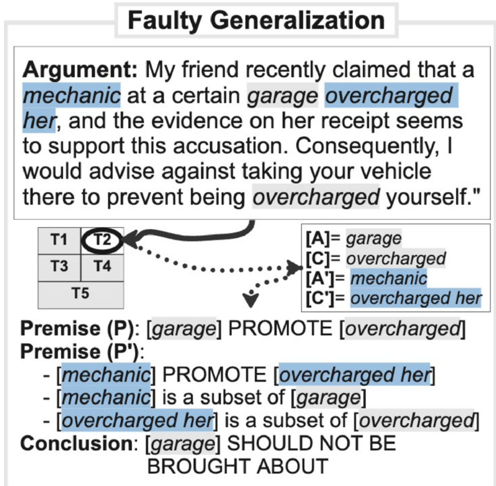

Figure 3: Examples of template and slot-fillers from FtF for Faulty Generalization.

3.3 Our Fallacy Template Inventory

For representing fallacy logical structure, we extend Walton et al. (2008) and Reisert et al. (2018) by introducing a new premise P' which supports premise P. Consider the following representation for Faulty Generalization:

- **Premise (P):** [A] *SUPPRESS* a *GOOD* [C].
- **Premise (P'):** [A'], a subset of A, *SUPPRESS* a *GOOD* [C]
- **Conclusion**: [A] is *BAD*.

Here, on top of the argument template placeholders A and C, P' includes a new placeholder A', where A' is an action and $A' \subseteq A$. The faulty generalization is committed as a result of the argument considering A' to represent A as a whole. Revisiting the argument in Fig. 1, we can instantiate the above with A="further advanced courses", A'="NLP class", and C="GPA".

Fig. 3 shows additional examples of template instantiation with placeholders for each target fallacy type, with our new premise P'. Using this figure, we exemplify a complex Faulty Generalization argument, where two subsets A' and C' are considered. The main point is symbolized by A="garage" and C="overcharged", as the narrative implies that the A is notorious for C. Hence, it is implicated that C is *BAD* and that A *PROMOTE* C. In P',' A'="mechanic" and C'="overcharged her" are identified, where $A' \subseteq A$ and $C' \subseteq C$ and A' PROMOTE C'. Therefore, the relation A' PROMOTES C' supports the relation A PROMOTE C, so template #2 is selected.

4 Flee the Flaw (FtF) Dataset

We discuss the creation of our dataset *Flee the Flaw* (henceforth, *FtF*). First, we use an existing dataset of annotated fallacious arguments for creating our guidelines and building our inventory of fallacy templates. We then conduct a full-fledged annotation on top of 400 arguments.

4.1 Data Collection

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 fallacy types. We sampled 400 arguments (100 per target fallacy type) 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.

4.2 Guideline Construction

We employed two expert annotators for guideline development and annotation: a native English-speaking postdoctoral researcher specializing in argumentation (who led guideline creation), and a graduate student specializing in argumentation.

To create a set of guidelines and test annotation feasibility, we conduct a multi-round pilot study on top of LOGIC-DEV$_{200}$. Aside from the pilot study itself, annotators did not go through any training phase. Given that the LOGIC dataset has limited fallacious arguments, our pilot study consisted of 200 instances (50 per fallacy type) for creating our final guidelines, where the study began with an initial set of guidelines for all fallacy types. For each of the four fallacy types, annotators focused on the 50 instances per each fallacy. For each type, we split up the instances to annotate (e.g., 10 out of 50) using the latest updated set of guidelines, where results were compared and discussed after each round. Discussion consisted of findings and whether annotators agree with each other's annotation. If there was a new finding or disagreement, instances were discussed to reach a consensus and guidelines were updated accordingly. The process was repeated until all 200 instances in LOGIC-DEV$_{200}$ were annotated, and the final annotation guidelines were created.[3]

Reducing Annotation Complexities During guideline construction, annotators found that multiple templates could be instantiated for a single argument. In order to reduce annotation complexity, the following conditions were created: i) *preservation of argument's original, explicit intent*, ii) *paraphrase arguments into Argument from Consequences*, and iii) *preference of entities over events.*

We demonstrate such conditions with the False Dilemma argument: "*We either have to cut taxes or leave a huge debt for our children.*". 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 Consequences, 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.

In addition to the above, it was discovered that the fallacy type provided by LOGIC could be categorized into other, non-target fallacy types (e.g., *Slippery Slope* instead of *Faulty Generalization*). In such instances, annotators were instructed to annotate the instance considering its given type and encouraged to apply template #5 if the template instantiation could not be made.

4.3 Annotation Procedure

Given a fallacious argument, its fallacy type, and our templates, the procedure for fallacious template instantiation is as follows. First, annotators select the appropriate template from the given set of 5 templates. Next, annotators fill in the necessary slot-fillers taken from the input argument. Afterwards, annotators provide their confidence level for instances in which they are not 100% confident. Finally, annotators provide any necessary comments to accompany the annotation. The resulting annotation of our fallacious templates on top of LOGIC-DEV$_{200}$ and LOGIC-TRAIN$_{200}$ resulted in FtF-DEV and FtF-TRAIN, respectively.

[3]The final guidelines are made publicly available: https://github.com/irfanrob/fallacy-template/tree/main

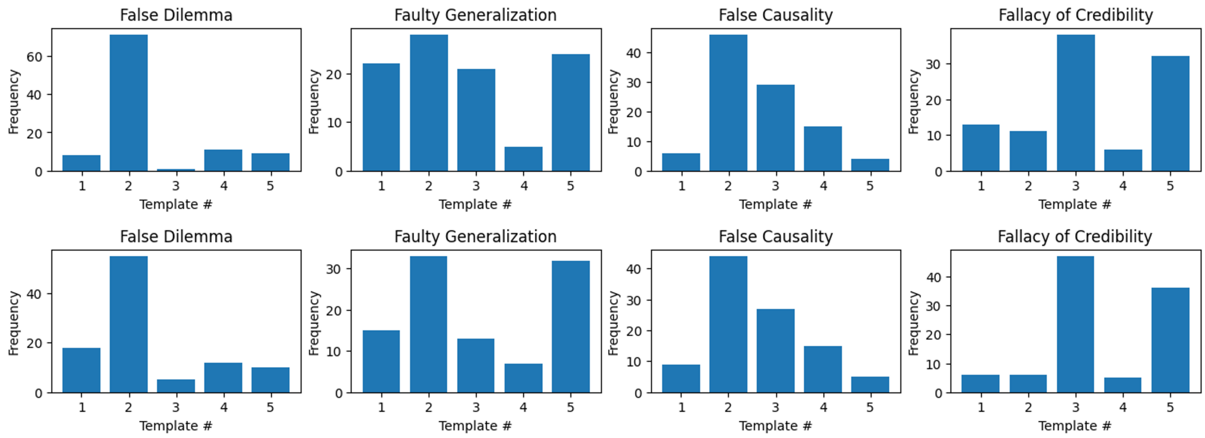

Figure 4: The distribution of fallacy templates in our FtF between one annotator (top row) and the other (bottom row) for all 400 instances in our train and dev set, where each fallacy type consists of 100 instances. The x-axis refers to the selected template, and y-axis refers to the frequency.

Fallacy Type	GWET AC1	Krippendorff's α
False Dilemma	0.63	0.44
Faulty Generalization	0.40	0.36
False Causality	0.71	0.65
Fallacy of Credibility	0.58	0.49
Average	0.57	0.54

Table 1: Template selection Inter-Annotator Agreement.

Fallacy Type	Annotator 1	Annotator 2
False Dilemma	90%	91%
Faulty Generalization	68%	76%
False Causality	95%	96%
Fallacy of Credibility	64%	83%
Average	80%	83%

Table 2: Coverage of fallacy templates for both annotators.

4.4 Statistics and Analysis

Inter-Annotator Agreement (IAA) Table 1 shows our IAA scores for template selection. Our GWET AC1 (Gwet, 2008) scores range from 0.40 to 0.71, indicating moderate to the substantial agreement. We also calculate Krippendorff's alpha (Hayes and Krippendorff, 2007) and achieve a score of 0.54, indicating a high agreement.

Given that *Faulty Generalization* had the lowest agreement, we conduct an additional analysis on all disagreements for *Faulty Generalization* arguments. We discover that 60% of disagreements were caused when one annotator labeled '#5' and the other instantiated a template, where reasons annotators labeled '#5' were 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*.

Distribution of Templates Fig. 4 shows the distribution of the fallacy templates for both annotators. We immediately observe that annotators rarely selected certain templates, such as template #3 for False Dilemma. Regardless of this skewed distribution, as reported, we still achieved a high IAA and coverage for template selection.

Coverage Table 2 provides a comparison of annotation coverage for annotators, namely the percentage of instances where a non-template #5 is annotated. Overall, our templates achieve a high coverage for both annotators, with scores of 80% and 83%. We observe that fallacy types such as *False Dilemma* and *False Causality* achieve high coverage due to their straightforward reasoning.

5 Experiments

To what extent is the automatic identification of fallacy logic structure challenging for machines? We evaluate current state-of-the-art LLMs for FtF.

5.1 Experiment Setting

The fallacy logic identification task comprises two sub-tasks: (i) *template selection* and (ii) *slot-filling*. As shown in Table 3, the prompt includes this fallacy-type information, allowing LLM to focus on two key actions. In template selection, the model chooses the template that best reflects the fallacious structure. For slot-filling, the model fills in the slots of the selected template.

 # Task Identify the underlying structure of an argument of {fallacy_type}. Given a list of fallacy templates, your task is to choose a template that best describes the underlying fallacy structure... **# List of Templates** Template No.1:\n {template_1} ... Template No.5:\nThere is either no consequence in the argument. **# Output Format** Template No.=[No.]\n{slot_fillers} **# Example** {examples}\n# **# Query** {}

Table 3: Generalized prompt used for our 0, 1, and 5-shot LLM experiments. {fallacy_type} is either Fallacy of Credibility, False Causality, Faulty Generalization, or False Dilemma. Depending on the fallacy type, the appropriate templates and slot-filler choices are provided to the prompt, and for 1 and 5-shot settings, {examples} are provided. For spacing purposes, we replace newlines with \n in this prompt and omit templates 2-4.

It is commonly known that dataset creation in argumentation requires significant resources (human, time, financial), making it difficult to acquire highly reliable large-scale annotations. Therefore, we employ LLMs with in-context learning and fine-tuning to model both sub-tasks jointly. We experiment with three distinct prompts: (i) NL_1, a pure natural language prompt, (ii) NL_2, simplified version of NL_1, and (iii) PL, a semi-structured prompt with propositional logic and mathematical notation. Table 3 summarizes a general form of these prompts; see Appendix A.5 for an example of the zero shot prompt for False Dilemma.[4]

5.2 Setup

Models We employed four state-of-the-art LLMs: GPT-3.5-turbo (Abdullah et al., 2022), GPT-4o (Achiam et al., 2023), Llama-3-8b(Meta, 2024), and Mistral-7b(Jiang et al., 2023). We use a temperature of 0, max tokens of 0.6, top_p of 1.0, and both frequency and presence penalties of 0. Experiments were conducted using zero-shot, one-shot, and five-shot prompt settings for GPT-3.5-turbo and GPT-4o. Few-shot examples were sampled from FtF-TRAIN, with the number of shots reflecting the number of examples provided in the prompt.

For the fine-tuned model, we split FtF-TRAIN into 150 instances as training data and 50 instances as validation data. We set the learning rate into 2e-4 and optimizer adamw8bit. All models used FtF-DEV for testing and evaluating the results.

[4] Detailed prompts used in our experiments are publicly available at https://github.com/irfanrob/fallacy-template/tree/main

Model	Acc. (TS)	Acc. (SF)	Acc. (Joint)
GPT4-0-shot	0.36	0.06	0.02
GPT4-1-shot	0.42	0.10	0.04
GPT4-5-shot	0.38	**0.24**	0.09
GPT3.5-0-shot	0.21	0.13	0.02
GPT3.5-1-shot	0.31	0.14	0.04
GPT3.5-5-shot	0.37	0.19	0.06
Llama3-8b	0.34	0.16	0.05
Mistral-7b	**0.47**	0.23	**0.11**

Table 4: Model accuracy for template selection (TS) and exact-match accuracy for slot-filling (SF).

Evaluation Metrics We use accuracy for the template section. For the slot-filling, we target only instances where the template is correctly identified by the model. Formally, we define *exact-match slot-filling accuracy* as follows: $\frac{|X \cap Y|}{|X|}$, where X is a set of test instances where the predicted template is correct, and Y is a set of test instances where *all* predicted slot-fillers must exactly match the gold-standard slot-fillers.[5] In addition, we use *partial-match slot-filling accuracy*, where Y is a set of test instances and *all* predicted slot-fillers are required to have over 50% word overlap with the gold standard.

For evaluating overall performance, we define a *joint accuracy* to be a multiplication of template selection accuracy and slot-filling accuracy.

5.3 Results and Analysis

Table 4 demonstrates low accuracy across all models. Regarding template selection, the Mistral-7b model generally outperforms every model. Conversely, in slot-filling, the results show that the GPT4 model with 5-shot prompting outperforms every model. Overall, the low joint accuracy highlights a significant limitation of state-of-the-art language models in identifying the logical fallacy structure that best captures the underlying fallacious structure within FtF. Improving LLMs' ability to handle slot-filling tasks remains a challenge.

5.4 Error Analysis

We conducted an error analysis on 40 instances, aiming to improve the template and prompt. We

[5] We lowercase all tokens for word matching.

	Fallacy Type	Example	Correct Answer	Predicted Answer
1	False Dilemma	"America: Love it or leave it. This is an example of which kind of logical fallacy?"	Template No.=2 $[A]$=Love it $[C]$=leave it	Template No.=5 $[A]$= $[C]$=
2	False Dilemma	We either ban hairspray or the world will end.	Template No.=4 $[A]$=hairspray $[C]$=the world will end	Template No.=2 $[A]$=ban hairspray $[C]$=the world will end
3	False Causality	Iâ€™ve never had the flu because I take my vitamins everyday.	Template No.=4 $[A]$=vitamins $[C]$=flu	Template No.=3 $[A]$=vitamins $[C]$=flu
4	Faulty Generalization	This new test seemed so promising, but the 3 studies that supported its validity turned out to have critical methodological flaws, so the test is probably not valid.	Template No.=2 $[A]$=test $[C]$=critical methodolical flaws $[A']$=3 studies that supported its validity turned out to have critical methodological flaws $[C']$=	Template No.=5 $[A]$= $[C]$= $[A']$= $[C']$=
5	Fallacy of Credibility	Albert Einstein was extremely impressed with this theory.	Template No.=5 $[A]$= $[C]$= $[X]$=	Template No.=2 $[A]$=this theory $[C]$=Albert Einstein $[X]$=extremely impressed

Table 5: False prediction generated by Mistral-7b model

focused on the Mistral-7b generated results due to their highest joint accuracy. We discovered the following errors: (i) The model predicted template 5 despite the argument being instantiatable (32.5%), (ii) The model predicted a different template due to different slot-fillers (32.5%), (iii) The model predicted a different template despite having similar slot-fillers as the gold label (17.5%), and (iv) The model instantiated the template despite no Argument from Consequences (17.5%). The false prediction examples generated by Mistral-7b model are available in Table 5.

We found that template 5 was sometimes predicted due to noise in the input argument. Among all the instances that fell into category (i), four instances predicted template 5 because of this noise.[6]

Although the prompts were built off our guidelines, we found that the model occasionally selected different templates due to many possible terms for slot-filling (category (ii)). Example 2 shows an instance where the model selected a different template due to the difference in slot-filler A. Upon further analysis, the model's predicted answer was also correct, as "ban hairspray" suppress "the world will end" possesses the same semantic meaning as "hairspray" promotes "the world will end".

It still remains a question as to why errors that fall into categories (iii) and (iv) occurred. Example 3 is an instance that falls into category (iii), where the model correctly predicts the slot-filler but chooses different templates. The template conclusion "Vitamins should be brought about" is correct, but the model incorrectly assigns a good sentiment to "flu" and creates the premise "Vitamins promote flu" which does not align with the argument's intention.

Example 5 highlights a case where the argument cannot be instantiated as it is not an Argument from Consequences (category (iv)). However, the model instantiates this argument into template 2 with the premise "This theory suppresses Albert Einstein" and got promoted by "extremely impressed" which has a completely different meaning from the input.

After analyzing Examples 3 and 5, it remains unclear whether the model struggles to define the sentiment of the slot-filler or whether it truly understands Argument from Consequences.

5.5 Template Generalizability

To demonstrate the generalization of our template to other domains outside the LOGIC dataset, we conduct a preliminary annotation on top of existing, pre-labeled fallacy datasets. In total, we identify four datasets: Climate (Alhindi et al., 2023), Argotario (Habernal et al., 2017), MAFALDA[7] (Helwe et al., 2023), and Covid (Bonial et al., 2022).

[6] See Example 1 in Table 5 for an instance of the input question noise leading to template 5.

[7] We found that MAFALDA was built upon the corpus by Sahai et al (Sahai et al., 2021), so we utilize only MAFALDA.

Dataset	Samples	Cov #1	Cov #2	IAA
Argotario	20	60%	60%	0.57
MAFALDA	20	81%	90%	0.44
Climate	10	60%	50%	0.08
Covid	10	40%	30%	0.44
Overall	60	60%	57%	0.45

Table 6: Coverage scores from both annotators (**Cov #1** represent the score from the first annotator and **Cov #2** represent the score from the second annotator) and IAA (Krippendorff's α) score of template selection task.

Among our target fallacy types, we observe that each dataset has *Faulty Generalization* and *Fallacy of Credibility*. Therefore, we focus on these specific types. For this annotation, we employ the same two annotators from our main annotation.

We randomly sample 60 arguments from the four datasets, resulting in coverage scores of 60% and 57% between both annotators. We observe that Covid achieves the lowest coverage results (40% and 30%). We determine the following reasons for low coverage in Covid. In the case of *Fallacy of Credibility* instances, we observe that the credible source information is occasionally absent (e.g., "The COVID-19 pandemic is not a real medical pandemic"), which we cannot instantiate with our templates (i.e., no X provided). We discover that some labeled fallacy types in Covid are mislabeled (e.g., "No, because if you start with same sex marriage, what is next? Marriage with animals?", a *Slippery Slope* argument labeled as *Hasty Generalization)*. Finally, we observe that some instances do not contain enough information to warrant the labeled fallacy type, making it difficult to instantiate a template (e.g., "Covid vaccines contain aborted babies." as "Hasty Generalization", but missing insufficient evidence).

Our IAA Krippendorff's α score achieved moderate agreement at 0.45, with Argotario reaching the highest IAA score (0.57). Disagreements frequently occurred in the Climate dataset, leading to a poor IAA score (0.076). Such disagreements were due to misinterpretations of the input arguments and non-compliance with the guideline rules. In response to this, a further round was conducted, where each annotator was shown the other annotator's annotation for all disagreeing instances and asked whether they agree or disagree with the annotation. For disagreements, annotators discussed until an agreement could be met. The resulting annotations were implemented as the gold standard label for additional LLM experiments.

Model	Acc. (TS)	Acc. (EM)	Acc. (PM)
GPT4-1-shot	0.40	0.00	0.21
Mistral-7b	0.37	0.00	0.50

Table 7: Model accuracy for template selection (TS), and exact-match (EM) accuracy and partial-match (PM) slot-filling accuracy on all 60 preliminary instances. EM and PM include correct prediction without template #5.

We selected the two best models in the experiment using the FtF dataset based on the template selection task (GPT4-1-shot and Mistral-7b). Table 7 shows that both of the best models that we selected struggle, especially for the slot-filling exact match tasks where both models are not able to generate a similar slot-filler with the gold label. The model also has a similar problem with the experiment using the FtF dataset where the model generated different slot-fillers but possessed similar semantic meanings.

6 Conclusion and Future Work

In this work, we conduct the first study to address logical fallacy structure by creating an inventory of fallacy templates. In total, we created 20 novel templates spanned across 4 fallacy types (Fallacy of Credibility, False Causality, False Dilemma, and Faulty Generalization). We created Flee the Flaw, a new dataset consisting of 400 arguments from LOGIC (Jin et al., 2022) annotated with fallacy logic structure, and publicly released both the corpus and guidelines. Our dataset achieved a high inter-annotator agreement (Krippendorf's α of 0.54) and coverage (83%). We experiment on top of our new dataset by conducting In-Context Learning and fine-tuning for fallacy logic structure identification and discover that it is still a significant challenge for state-of-the-art language models. On top of LOGIC, we also test our templates on 60 randomly sampled instances across four additional datasets which results in inter-annotator agreement (Krippendorf's α of 0.45) and coverage (57%).

Our next step involves studying the underlying patterns and reasoning errors in arguments by analyzing the logical structure of fallacies. Simultaneously, we plan to conduct large-scale annotation on top of lengthier, more natural arguments. Finally, we plan to explore non-consequential topics and consider more Argumentation Schemes.

Limitations

In this research, we mainly focus on the proposed explainable fallacy template for only 4 fallacy types which are all mainly informal fallacies. We do not address the fallacy of logic which is the extension from the informal fallacy to formal fallacy. To keep annotation simple, our templates do not cover every possible combination of ingredients (e.g. relations such as *NOT PROMOTE*, *NOT SUPPRESS*) which limits the amount of total instantiations we can acquire. Regardless, we still achieved a coverage score of roughly 80%. Furthermore, we extend on argument templates (Reisert et al., 2018) which were inspired by Walton (2008)'s Argument from Consequences scheme which is a common scheme for every day arguments, but may limit the full range of fallacy instantiations that 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. 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. 4, 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.

Acknowledgements

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

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*.

Tariq Alhindi, Tuhin Chakrabarty, Elena Musi, and Smaranda Muresan. 2023. Multitask instruction-based prompting for fallacy recognition. *arXiv preprint arXiv:2301.09992*.

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

Claire Bonial, Austin Blodgett, Taylor Hudson, Stephanie M. Lukin, Jeffrey Micher, Douglas Summers-Stay, Peter Sutor, and Clare Voss. 2022. The search for agreement on logical fallacy annotation of an infodemic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4430–4438, Marseille, France. European Language Resources Association.

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.

Pierpaolo Goffredo, Mariana Espinoza, Serena Villata, and Elena Cabrio. 2023. Argument-based detection and classification of fallacies in political debates. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11101–11112. 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.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Chadi Helwe, Tom Calamai, Pierre-Henri Paris, Chloé Clavel, and Fabian Suchanek. 2023. Mafalda: A benchmark and comprehensive study of fallacy detection and classification. *arXiv preprint arXiv:2311.09761*.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

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.

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

Saumya Sahai, Oana Balalau, and Roxana Horincar. 2021. Breaking down the invisible wall of informal fallacies in online discussions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 644–657, Online. Association for Computational Linguistics.

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

Zhivar Sourati, Vishnu Priya Prasanna Venkatesh, Darshan Deshpande, Himanshu Rawlani, Filip Ilievski, Hông-Ân Sandlin, and Alain Mermoud. 2023b. Robust and explainable identification of logical fallacies in natural language arguments. *Knowledge-Based Systems*, 266:110418.

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

Stephen E Toulmin. 2003. *The uses of argument*. Cambridge university press.

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. 2013. *Argumentation schemes for presumptive reasoning*. Routledge.

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

A Appendix

A.1 Template Examples

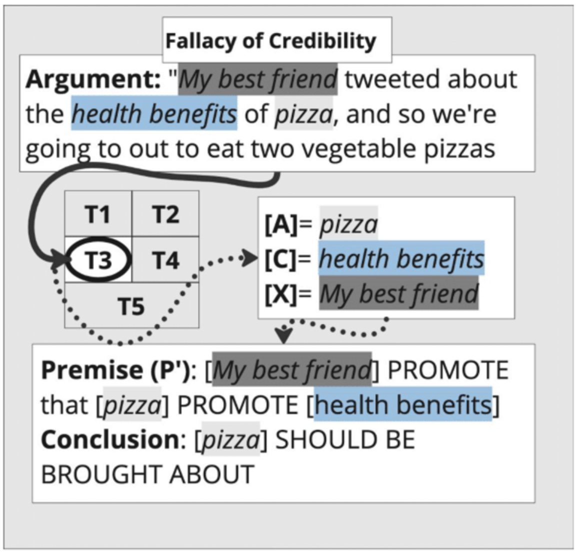

Figure 5: Examples of template and slot-fillers from FtF for Fallacy of Credibility.

Shown in Fig. 5 is the example of Fallacy of Credibility. For the Fallacy of Credibility argument, the fallacy is committed as the X="best friend" is promoting that A="pizza" has C="health benefits", resulting in P': X="best friend" promote that A="pizza" promote C="health benefits", thus Conclusion is A="pizza" should be brought about. However, the friend is not an expert in the field of nutrition.

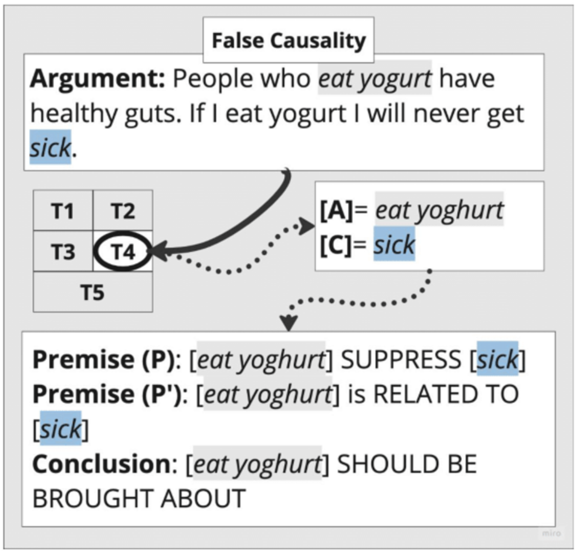

Figure 6: Examples of template and slot-fillers from FtF for False Causality.

For the False Causality argument shown in Fig. 6, the argument is stating that A="eat yoghurt" has a correlation with people with healthy guts, and thus the P: A="eat yoghurt" suppressing C="sick". The False Causality is linked, as it's implying that A="eating yoghurt" will definitely suppress C="sick". In conclusion, A="eating yoghurt" should be brought about.

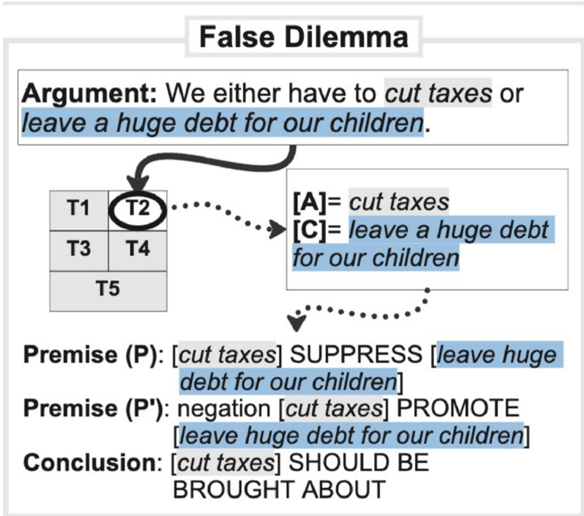

Figure 7: Examples of template and slot-fillers from FtF for False Dilemma.

The example of argument shown in Fig. 7 is considered as False Dilemma fallacy. The argument limited the option to A="cut taxes" and negation of A="cut taxes" for determine the consequence of C="leave a huge debt for our children". It conclude A="cut taxes" should be brought about without considering any possible action except P: A="cut taxes" suppress C="leave a huge debt for our children", and P': negation of A="cut taxes" promote C="leave a huge debt for our children".

A.2 Fallacy Types

False Dilemma occurs due to the restriction of the choices and ignoring additional potential options. *Faulty Generalization* occurs when a belief is applied to a large population without a sufficient and unbiased sample. *False Causality* assumes a cause-and-effect relationship between two events. Finally, *Fallacy of Credibility* involves an appeal to ethics, authority, or credibility that is not directly relevant to the argument. Table 8 provides a definition, example, and further explanation of the example for False Dilemma, Faulty Generalization, False Causality, and Fallacy of Credibility.

Fallacy Type	Definition	Example	Explanation
False Dilemma	This fallacy is when incorrect limitations are made on the possible options in a scenario when there could be other options.	We either have to cut taxes or leave a huge debt for our children	This argument only limits the options into “cut taxes” or “not cut taxes” for dealing with a “debt” without considering other potential options.
Faulty Generalization	This fallacy occurs when an argument applies a belief to a large population without having a large enough sample to do so.	I took an NLP class, an advanced course in Stanford. I suggest not taking further advanced courses because they will hurt your GPA.	This argument generalizes “further advanced courses” should not be taken due to hurting the person’s "GPA" only because the took one of the advanced courses “NLP class.”
False Causality	This fallacy occurs when an argument assumes that since two events are correlated,they must also have a cause and effect relationship.	People who eat yoghurt have healthy guts. If I eat yoghurt I will never get sick	This argument has a belief that “eat yoghurt” has a strong relation with “never get sick” because of having “healthy guts”. Thus, it believes that by eating yoghurt will “never get sick.”
Fallacy of Credibility	This fallacy is when an appeal is made to some form of ethics, authority, or credibility.	My Best friend tweeted about the health benefits of pizza, and so we’re going to out to eat two vegetable pizzas	The argument has been promoted by the person’s best friend by “tweet about the health benefits of pizza”, but the person best friend is not an expert in the field of nutrition which makes the argument is not credible.

Table 8: Definition and example explanation of four defective induction fallacy types

A.3 Prompt Type for Template Selection and Exact Match Performance

We report the result of template selection accuracy and the average accuracy of slot-filling for exact match for every three prompt types using GPT-4 in table 9 and GPT3.5 in table 10.

Template selection performs better for 1-shot prompting for every prompt type in the GPT4 model. However, not for the slot-filling task, 5-shot prompting outperforms 1-shot prompting for every prompt type despite not having the highest accuracy in the template selection task. Different from GPT3.5 where every task is dominated by 5-shot prompting for every prompt type.

Overall, model performance shows minimal variation based on prompt type, suggesting that prompt variation has no significant impact on performance.

Pr	n	**Acc. (TS)**	**Acc. (SF)**	**Acc. (Joint)**
NL_1	0	0.31	0.10	0.03
NL_1	1	**0.36**	0.12	0.04
NL_1	5	0.32	**0.22**	**0.07**
NL_2	0	0.36	0.06	0.02
NL_2	1	**0.42**	0.10	0.04
NL_2	5	0.38	**0.24**	**0.09**
PL	0	0.32	0.10	0.03
PL	1	**0.38**	0.10	0.04
PL	5	0.31	**0.18**	**0.06**

Table 9: GPT-4 accuracy for template selection (TS) and exact-match accuracy for slot-filling (SF). n denotes the number of few-shot examples, and **Pr** denotes a prompt type.

Pr	n	**Acc. (TS)**	**Acc. (SF)**	**Acc. (Joint)**
NL_1	0	0.21	0.12	0.02
NL_1	1	0.31	0.14	0.04
NL_1	5	**0.37**	**0.19**	**0.07**
NL_2	0	0.21	0.06	0.01
NL_2	1	0.30	0.14	0.04
NL_2	5	**0.35**	**0.17**	**0.06**
PL	0	0.21	0.13	0.03
PL	1	0.29	0.04	0.01
PL	5	**0.37**	**0.17**	**0.06**

Table 10: GPT-3.5 accuracy for template selection (TS) and exact-match accuracy for slot-filling (SF).

A.4 Prompt Type for Template Selection and Partial Match Performance

We report the average accuracy of slot-filling for partial match. The results are shown in table 11 for GPT-4 and table 12 GPT3.5. Despite NL_2 zero-shot prompt on GPT4 model performance of only 0.06 accuracy for an exact match slot-filling task in table 9, it performs the best with 0.49 accuracy in the partial match slot-filling task.

Pr	n	**Acc. (TS)**	**Acc. (SF)**	**Acc. (Joint)**
NL_1	0	0.31	0.24	0.07
NL_1	1	**0.36**	0.43	**0.16**
NL_1	5	0.32	0.32	0.10
NL_2	0	0.36	**0.49**	**0.17**
NL_2	1	**0.42**	0.35	0.15
NL_2	5	0.38	0.42	0.16
PL	0	0.32	0.32	**0.10**
PL	1	**0.38**	0.21	0.08
PL	5	0.31	**0.33**	**0.10**

Table 11: GPT-4 accuracy for template selection (TS) and partial-match accuracy for slot-filling

Pr	n	**Acc. (TS)**	**Acc. (SF)**	**Acc. (Joint)**
NL_1	0	0.21	0.12	0.02
NL_1	1	0.31	0.33	0.10
NL_1	5	**0.37**	**0.37**	**0.14**
NL_2	0	0.21	0.19	0.04
NL_2	1	0.29	**0.36**	0.11
NL_2	5	**0.37**	**0.36**	**0.12**
PL	0	0.21	0.20	0.04
PL	1	0.30	**0.43**	0.12
PL	5	**0.35**	0.38	**0.14**

Table 12: GPT-3.5 accuracy for template selection (TS) and partial-match accuracy for slot-filling

A.5 Prompt for LLM Experiments

Table 13, Table 14, Table 15 provides an example of the zero shot prompt for False Dilemma used during our LLM experiments. Instances used for non-zero-shot settings are randomly selected from FtF-TRAIN$_{200}$.

Task
Identify the underlying structure of an argument of False Dilemma.
Given a list of fallacy templates, your task is to choose a template that best describes the underlying fallacy structure, choosing the template's placeholders, $[A]$ and $[C]$, directly from the input text. Additionally, the text must be a consecutive sequence of one or more terms without any conjugation.
Please follow the output format.
Definitions
Entity: a noun phrase in the input.
Event: a verb phrase in the input.
Placeholder: A fill-in-the-blank choice within a template. Each placeholder may either be an entity or an event.
Please note! Placeholders can ONLY be either an entity (i.e., noun phrase) or an event (i.e., verb phrase) and may not be any other type of phrase (e.g., prepositional phrase).
List of Templates
Template No.1:
Premise 1: An entity/action [A] promotes a good entity/action [C].
Premise 2: The absence of an entity/action [A] suppresses a good entity/action [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should be brought about.
Template No.2:
Premise 1: An entity/action [A] suppresses a bad entity/action [C]
Premise 2: The absence of an entity/action [A] promotes a bad entity/action [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should be brought about.
Template No.3:
Premise 1: An entity/action [A] suppresses a good entity/action [C]
Premise 2: The absence of an entity/action [A] promotes a good entity/action [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should not be brought about.
Template No.4:
Premise 1: An entity/action [A] promotes a bad entity/action [C].
Premise 2: The absence of an entity/action [A] suppresses a bad entity/action [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should not 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.]
$[A]$=
$[C]$=
Important Criteria: Prioritizing entities over events for placeholder.
For choosing placeholder, please prioritize entities over events in the case that an entity itself captures the underlying intent of the argument opposed to the event. However, if the event makes more sense, please choose an event for the placeholder.
Correct Example
Input: To get better schools, we have to raise taxes. If we don't, we can't have better schools.
Output:
Template No.=1
$[A]$=raise taxes
$[C]$=schools
Explanation:
Here, there are 2 possible options for [C] which are "schools" (i.e., entity) and "can't have better schools" (i.e., event). Since the entity is the top priority and the second option does not work with template 1 because it is a suppressed relation, "schools" is cchosen for [C].
Also, [A] and [C] are taken directly from the input text. For example, "raising taxes" as [A] also sounds correct, but the term "raising" is not mentioned in the input text. That is why "raise taxes" is chosen for [A]. Because the argument believes that "raise taxes" promote "schools" while not "raise taxes" suppress "school". he conclusion is implicit that "Premise 1 supports that raise taxes should be brought about." Thus, Template No.=1 is selected.
Wrong Example
Input: To get better schools, we have to raise taxes. If we don't, we can't have better schools.
Output:
Template No.=1
$[A]$=raising taxes
$[C]$=can't have better schools
Explanation:
Here, there are 2 possible options for [C] which are "schools" (i.e., entity) and "can't have better schools" (i.e., event). However, "can't have better schools" as [C] is incorrect because it is an event instead of the entity of "schools" which already makes sense.
Also, "raising taxes" as [A] is incorrect because the placeholder is not taken directly from the text. Here "raising taxes" is chosen as [A] but the word "raising" does not appear in the input text. Therefore the correct choice for [A] is "raise taxes".
Again, please only select the placeholders directly from the text!
Query
{}

Table 13: Natural Language (NL_1): Zero shot False Dilemma prompt for LLM experiment

Task
Identify the underlying structure of an argument of False Dilemma.
Given a list of fallacy templates, your task is to choose a template that best describes the underlying fallacy structure, choosing the template's placeholders, $[A]$ and $[C]$, directly from the input text. Additionally, the text must be a consecutive sequence of one or more terms without any conjugation.
Please follow the output format.
Definitions
Entity: a noun phrase in the input.
Event: a verb phrase in the input.
Placeholder: A fill-in-the-blank choice within a template. Each placeholder may either be an entity or an event.
Please note! Placeholders can ONLY be either an entity (i.e., noun phrase) or an event (i.e., verb phrase) and may not be any other type of phrase (e.g., prepositional phrase).
List of Templates
Template No.1:
Premise 1: An entity/event [A] promotes a good entity/event [C].
Premise 2: An entity/event [¬A] suppresses a good entity/event [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should be brought about.
Template No.2:
Premise 1: An entity/event [A] suppresses a bad entity/event [C]
Premise 2: An entity/event [¬A] promotes a bad entity/event [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should be brought about.
Template No.3:
Premise 1: An entity/event [A] suppresses a good entity/event [C]
Premise 2: An entity/event [¬A] promotes a good entity/event [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should not be brought about.
Template No.4:
Premise 1: An entity/event [A] promotes a bad entity/event [C]
Premise 2: An entity/event [¬A] suppresses a bad entity/event [C].
Conclusion: Therefore, both Premise 1 and Premise 2 support that [A] should not 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.]
$[A]$=
$[C]$=
Important Criteria: Prioritizing entities over events for placeholder.
For choosing placeholder, please prioritize entities over events in the case that an entity itself captures the underlying intent of the argument opposed to the event. However, if the event makes more sense, please choose an event for the placeholder.
Correct Example
Input: To get better schools, we have to raise taxes. If we don't, we can't have better schools.
Output:
Template No.=1
$[A]$=raise taxes
$[C]$=schools
Explanation:
Here, there are 2 possible options for [C] which are "schools" (i.e., entity) and "can't have better schools" (i.e., event). Since the entity is the top priority and the second option does not work with template 1 because it is a suppressed relation, "schools" is cchosen for [C].
Also, [A] and [C] are taken directly from the input text. For example, "raising taxes" as [A] also sounds correct, but the term "raising" is not mentioned in the input text. That is why "raise taxes" is chosen for [A]. Because the argument believes that "raise taxes" promote "schools" while not "raise taxes" suppress "school". he conclusion is implicit that "Premise 1 supports that raise taxes should be brought about." Thus, Template No.=1 is selected.
Wrong Example
Input: To get better schools, we have to raise taxes. If we don't, we can't have better schools.
Output:
Template No.=1
$[A]$=raising taxes
$[C]$=can't have better schools
Explanation:
Here, there are 2 possible options for [C] which are "schools" (i.e., entity) and "can't have better schools" (i.e., event). However, "can't have better schools" as [C] is incorrect because it is an event instead of the entity of "schools" which already makes sense.
Also, "raising taxes" as [A] is incorrect because the placeholder is not taken directly from the text. Here "raising taxes" is chosen as [A] but the word "raising" does not appear in the input text. Therefore the correct choice for [A] is "raise taxes".
Again, please only select the placeholders directly from the text!
Query
{}

Table 14: Propositional Logic (PL): Zero shot False Dilemma prompt for LLM experiments.

Task
Identify the underlying structure of an argument of False Dilemma.
Given a list of fallacy templates, your task is to choose a template that best describes the underlying fallacy structure, choosing the template's placeholders, $[A]$ and $[C]$, directly from the input text. Additionally, the text must be a consecutive sequence of one or more terms without any conjugation.
Please follow the output format.
Definitions
Entity: a noun phrase in the input.
Event: a verb phrase in the input.
Placeholder: A fill-in-the-blank choice within a template. Each placeholder may either be an entity or an event.
Please note! Placeholders can ONLY be either an entity (i.e., noun phrase) or an event (i.e., verb phrase) and may not be any other type of phrase (e.g., prepositional phrase).
List of Templates
Template No.1:
Premise 1: An entity/event [A] promotes a good entity/event [C].
Premise 2: The absence of an entity/event [A] suppresses a good entity/event [C].
Conclusion: Therefore, [A] should be brought about.
Template No.2:
Premise 1: An entity/event [A] suppresses a bad entity/event [C]
Premise 2: The absence of an entity/event [A] promotes a bad entity/event [C].
Conclusion: Therefore, [A] should be brought about.
Template No.3:
Premise 1: An entity/event [A] suppresses a good entity/event [C]
Premise 2: The absence of an entity/event [A] promotes a good entity/event [C].
Conclusion: Therefore, [A] should not be brought about.
Template No.4:
Premise 1: An entity/event [A] promotes a bad entity/event [C]
Premise 2: The absence of an entity/event [A] suppresses a bad entity/event [C].
Conclusion: Therefore, [A] should not 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.]
$[A]$=
$[C]$=
Important Criteria: Prioritizing entities over events for placeholder.
For choosing placeholder, please prioritize entities over events in the case that an entity itself captures the underlying intent of the argument opposed to the event. However, if the event makes more sense, please choose an event for the placeholder.
Correct Example
Input: To get better schools, we have to raise taxes. If we don't, we can't have better schools.
Output:
Template No.=1
$[A]$=raise taxes
$[C]$=schools
Explanation:
Here, there are 2 possible options for [C] which are "schools" (i.e., entity) and "can't have better schools" (i.e., event). Since the entity is the top priority and the second option does not work with template 1 because it is a suppressed relation, "schools" is cchosen for [C].
Also, [A] and [C] are taken directly from the input text. For example, "raising taxes" as [A] also sounds correct, but the term "raising" is not mentioned in the input text. That is why "raise taxes" is chosen for [A]. Because the argument believes that "raise taxes" promote "schools" while not "raise taxes" suppress "school". he conclusion is implicit that "Premise 1 supports that raise taxes should be brought about." Thus, Template No.=1 is selected.
Wrong Example
Input: To get better schools, we have to raise taxes. If we don't, we can't have better schools.
Output:
Template No.=1
$[A]$=raising taxes
$[C]$=can't have better schools
Explanation:
Here, there are 2 possible options for [C] which are "schools" (i.e., entity) and "can't have better schools" (i.e., event). However, "can't have better schools" as [C] is incorrect because it is an event instead of the entity of "schools" which already makes sense.
Also, "raising taxes" as [A] is incorrect because the placeholder is not taken directly from the text. Here "raising taxes" is chosen as [A] but the word "raising" does not appear in the input text. Therefore the correct choice for [A] is "raise taxes".
Again, please only select the placeholders directly from the text!
Query
{}

Table 15: Natural Language$_2$ (NL$_2$): Zero shot False Dilemma prompt for LLM experiments.