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 (0.83). 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 (0.18 accuracy). To facilitate research on fallacies, we make our dataset and guidelines 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 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 pro-

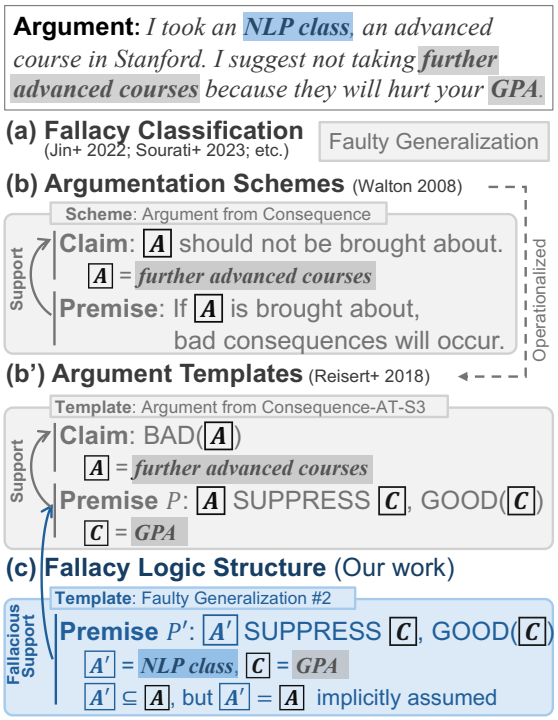

Figure 1: Overview of our proposed fallacy logic structure. We extend (b') existing argumentative representation (Reisert et al., 2018) consisting of Claim and Premise P by adding (c) Premise P', which explains what makes the argument fallacious. The example annotation shows: (i) the claim "*further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

vides 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 Consequence* (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 both the dataset and guidelines[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, namely GPT-3.5 and GPT-4 (§5).

2 Related Work

Fallacies Annotation Study Several studies address creating benchmarks for fallacy identification, including (Habernal et al., 2017) for game facilitation and (Ruiz-Dolz and Lawrence, 2023) for validating argumentation corpora. Particularly, Jin et al. (2022) focus 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 benchmark establishment, this research proposes benchmark resources aimed at capturing fallacy structure rather than solely identifying fallacies. This 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) establishes methods for parsing argumentation structure in persuasive essays by identifying and classifying argument components and their relationships. (Toulmin, 2003) provides a framework for analyzing arguments by breaking them down into components like Claim, Grounds, Warrant, and Rebuttal. (Walton, 2013) focuses 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 Consequence (Walton

[1] https://github.com/itsanonnymous/fallacytemplate

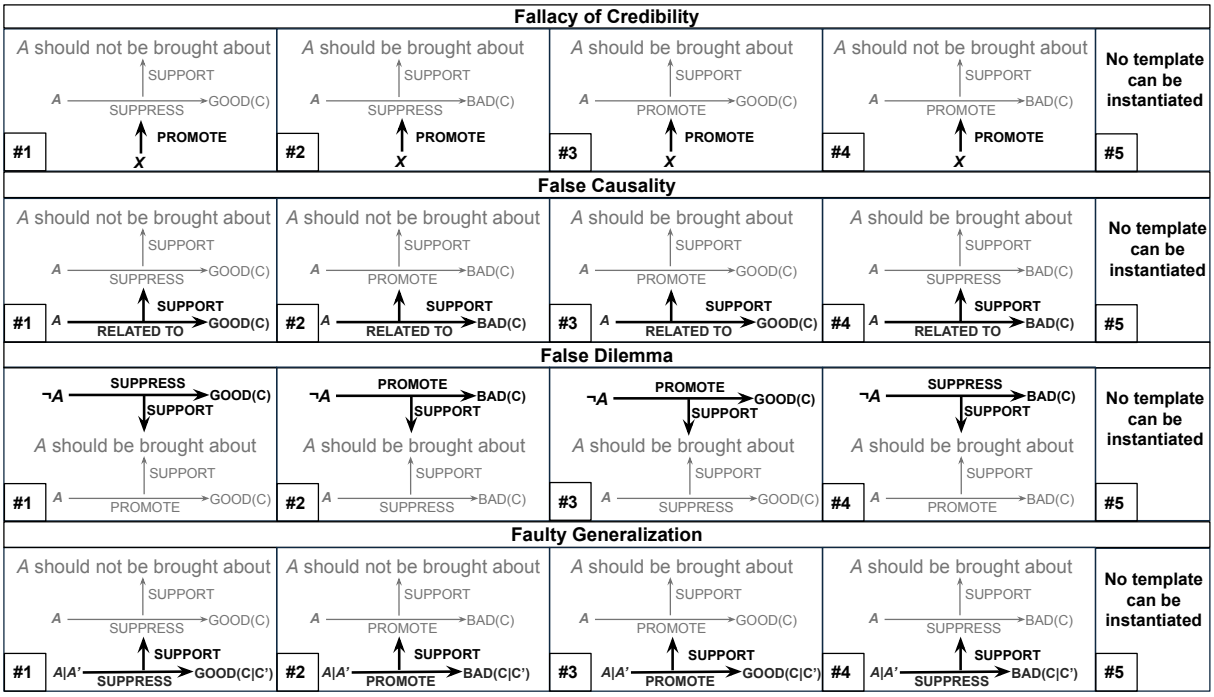

Figure 2: Our templates for annotating fallacious argument logical structure. We extend upon existing work (Walton et al., 2008; Reisert et al., 2018), consisting of a conclusion (i.e., *A should (not) be brought about*) and supporting premise, by adding an additional supporting premise in bold which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

et al., 2008) emphasizes potential outcomes of actions, often involving causality and appeals 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 Consequence 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) on 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 Consequence 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 a majority of fallacy types. We focus on the fallacies most commonly experimented with in the field of computational argumentation, including (Alhindi et al., 2023) and (Helwe et al., 2023). These studies provide statistics on the types of fallacies encountered, guiding us to design our templates to match the most frequently occurring types. We develop 20 new templates covering four defective induction fallacy types–Fallacy of Credibility, False Causality, False Dilemma, and Faulty Generalization. *False Dilemma* occurs due to restrictions on the available choices, preventing consideration of additional potential options. *Faulty Generalization* occurs when a belief is applied to a large population without a sufficient and unbiased sample. *False Causality* assumes that when two events occur together, they must have a cause-and-effect relationship. Finally, *Fallacy of Credibility* involves an appeal to ethics, authority, or credibility that is not directly relevant to the argument.

Third, our annotation scheme must utilise Reis-

ert et al. (2018) template selection and slot-filling approach further simplifying annotation while remaining computationally friendly. As inspired by the Argument from Consequence 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) represents an *action* and P supports conclusion. For the argument in Fig. 1, we represent Argument from Negative Consequence with [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 Figure 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.

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".

Our template inventory is shown in Fig. 2, with the new premise P' in bold. From this figure, we can generalize P' for each fallacy type as follows:

- **Fallacy of Credibility P':** [x] *PROMOTE* P
- **False Dilemma P':** [$\neg A|A'$] *PROMOTE* or *SUPPRESS* a *GOOD* or *BAD* [$C|C'$]
- **False Causality P':** [A] *RELATED TO* a *GOOD* or *BAD* [C]
- **Faulty Generalization P':** [$A|A'$] *PROMOTE* or *SUPPRESS* a *GOOD* or *BAD* [$C|C'$]

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

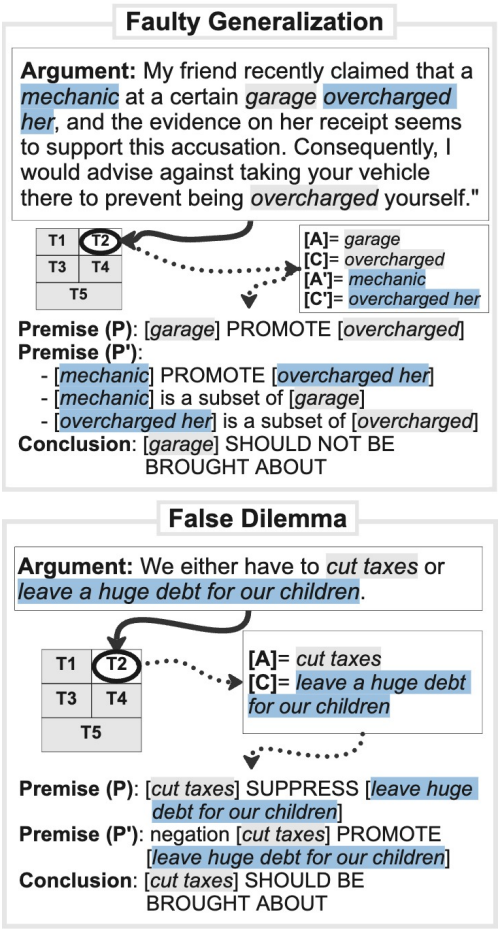

Figure 3: Examples of template and slot-fillers from FtF for Faulty Generalization and False Dilemma fallacies.

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 template 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 Annotator Background

We employed two expert annotators for guideline development and annotation: a native English-speaking postdoctoral researcher specializing in argument mining (who led guideline creation), and a non-native English-speaking graduate student specializing in argumentation (IELTS score 6.5).

4.3 Guideline Construction

In order 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 phrase. 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]

[3]The final guidelines are made publicly available: https://github.com/itsanonnymous/fallacytemplate

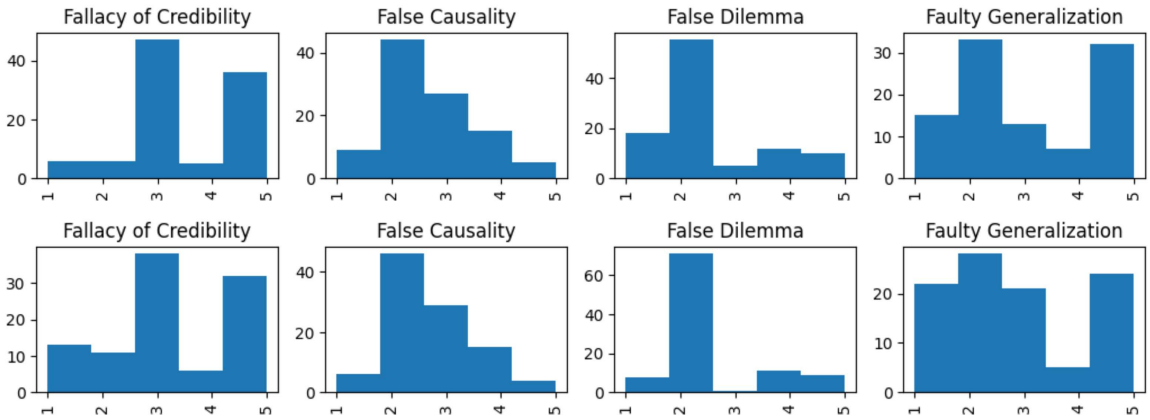

Figure 4: The distribution of fallacy templates in our FtF between one annotator (top row) and the other (bottom row) for all 400 instances in our train and dev set, where each fallacy type consists of 100 instances. The x-axis refers to the selected template, and y-axis refers to the frequency.

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.*

As shown in Fig. 3, 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 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.

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.

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.

4.4 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 write 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.

4.5 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

Fallacy Type	Annotator 1	Annotator 2
False Dilemma	0.90	0.91
Faulty Generalization	0.68	0.76
False Causality	0.95	0.96
Fallacy of Credibility	0.64	0.83
Average	0.80	0.83

Table 2: Coverage of fallacy templates for both annotators.

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 certain templates were rarely selected by annotators for LOGIC, such as template #3 for False Dilemma. In the case of *False Dilemma* fallacies, the structure of the argument generally follows the pattern "Either A or C", where A is GOOD and C is BAD, and A SUPPRESS C, which is categorized as template #2. An example is as follows: "Either you love me or you hate me.", where A="love me" and C="hate me". 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 between two annotators, namely the percentage of instances where a non-template #5 is annotated, on top of FtF. Overall, our templates achieve a high annotation coverage for both annotators, with coverage scores of 80% and 83%, respectively. 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 Methodology

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

 # 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,
filling the template's placeholders.
Please follow the Output Format!!!
 # List of Templates
Template No.1:\n {template_1}
...
Template No.5:\nThere is either no consequence in the argument, or the argument cannot be instantiated with one of the templates above.
 # 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. Please see the Appendix for an example of the 5-shot prompt used when {fallacy_type}="False Causality".

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.

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 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 Table 7, 9, and 8 in Appendix A.3 for an example of the 5-shot prompt for False Dilemma.[4]

5.2 Setup

Models We employed two state-of-the-art LLMs: GPT-3.5-turbo (Abdullah et al., 2022) and GPT-4-1106-preview (Achiam et al., 2023). We use a temperature of 1.0, max tokens of 0.6, top_p of

[4]Detailed prompts used in our experiments are publicly available at https://github.com/itsanonnymous/fallacytemplate/tree/main/ftf_prompts

1.0, and both frequency and presence penalties of 0. Experiments were conducted using zero-shot, one-shot, and five-shot prompt settings. Few-shot examples were sampled from FtF-TRAIN, with the number of shots reflecting the number of examples provided in the prompt.

Evaluation Metrics We use accuracy for the template section. For the slot-filling, we will 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 where *all* predicted slotfillers are required to have over 50% word overlap with the gold standard.

Finally, for evaluating overall performance, we define a *joint accuracy* to be a multiplication of template selection accuracy and slot filling accuracy. To mitigate the probabilistic nature of LLMs, we performed each experiment five times and report the averaged results and the standard deviation.

5.3 Results and Analysis

Tables 4 and 5 demonstrate low accuracies across prompt variations. Regarding template selection, GPT-4 models generally outperform GPT3.5 models. Conversely, in slot filling, the results show that the GPT-3.5 models with zero-shot prompting outperform the GPT4 model across all prompt types. Additionally, model performance shows minimal variation based on prompt type, suggesting that prompt variation has no significant impact on performance. Overall, the low joint accuracy highlights a significant limitation of GPT models in identifying the logical fallacy structure that best captures the underlying fallacious structure within FtF. Improving GPT models' ability to handle slot-filling tasks remains a significant challenge.

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 and released Flee the Flaw, a new, novel dataset consisting of 400 arguments from LOGIC (Jin et al., 2022) annotated with fallacy logic structure and publicly release both the corpus and guidelines.[6] Our dataset achieved a high inter-annotator agreement (Krippendorf's α of 0.54) and coverage (0.83%). We experiment on top of our new dataset by conducting In-Context Learning for fallacy logic structure identification and discover that logical fallacy structure identification is a significant challenge for state-of-the-art language models such as GPT-3.5 and GPT-4.

Our immediate next step involves studying the underlying patterns and reasoning errors in arguments by analyzing the logical structure of fallacies. Simultaneously, we plan to utilize our templates for conducting a large-scale annotation on top of lengthier, more natural arguments. Finally, we plan to explore more non-consequential topics, allowing for more Argumentation Schemes to be considered.

Pr	n	Acc. (TS)	Acc. (SF)	Acc. (Joint)
NL_1	0	0.28±0.02	0.45±0.04	0.12
NL_1	1	0.30±0.01	0.45±0.02	0.14
NL_1	5	0.37±0.01	0.34±0.03	0.13
NL_2	0	0.31±0.01	0.49±0.04	**0.15**
NL_2	1	0.36±0.01	0.41±0.03	**0.15**
NL_2	5	0.37±0.02	0.34±0.05	0.12
PL	0	0.29±0.02	0.41±0.05	0.12
PL	1	0.31±0.02	0.40±0.02	0.13
PL	5	0.41±0.02	0.35±0.02	0.14

Table 4: 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	0.19±0.01	0.83±0.15	0.16
NL_1	1	0.30±0.02	0.52±0.02	0.15
NL_1	5	0.30±0.02	0.56±0.02	0.17
NL_2	0	0.18±0.01	0.94±0.01	0.17
NL_2	1	0.25±0.02	0.52±0.04	0.13
NL_2	5	0.29±0.02	0.58±0.05	0.16
PL	0	0.19±0.01	0.87±0.05	0.16
PL	1	0.29±0.01	0.61±0.05	**0.18**
PL	5	0.29±0.02	0.54±0.04	0.15

Table 5: GPT-3.5 accuracy for template selection (TS) and exact-match accuracy for slot filling (SF).

[5] We lowercased all tokens for word matching.

[6] https://github.com/itsanonnymous/fallacytemplate

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 fallacy 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 Consequence 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.

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. 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.

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.

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.

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.

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.

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.

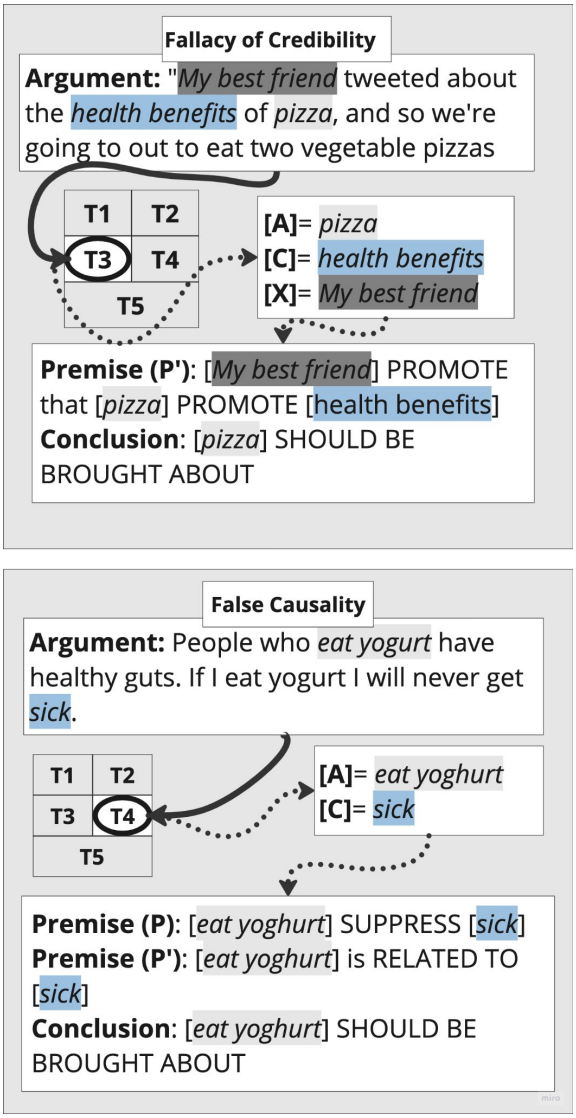

Figure 5: Examples of template and slot-fillers from FtF for Fallacy of Credibility and False Causality.

A Appendix

A.1 Template Examples

Shown in Fig. 5 are examples for both Fallacy of Credibility and False Causality arguments. For the Fallacy of Credibility argument, the fallacy is committed as the "best friend" is promoting that "pizza" has "health benefits", but the friend is not an expert in the field of nutrition. For the False Causality argument, the argument is stating that "eat yoghurt" has a correlation with people with healthy guts, and thus suppressing "sick". The False Causality is linked, as it's implying that "eating yoghurt" will definitely suppress "sick".

Pr	n	Acc. (TS)	Acc. (SF)	Acc. (Joint)
NL_1	0	0.28±0.02	0.45±0.04	0.12
NL_1	1	0.30±0.01	0.45±0.02	0.14
NL_1	5	0.37±0.01	0.34±0.03	0.13
NL_2	0	0.31±0.01	0.49±0.04	0.15
NL_2	1	0.36±0.01	0.41±0.03	0.15
NL_2	5	0.37±0.02	0.34±0.05	0.12
PL	0	0.29±0.02	0.41±0.05	0.12
PL	1	0.31±0.02	0.40±0.02	0.13
PL	5	0.41±0.02	0.35±0.02	0.14

Table 6: Performance of Template Selection and Partial Match for Slot Filling (GPT-4).

A.2 Partial Match Slot Filling

We report the average accuracy of slot-filling for partial match. The results are shown in Table 6 (GPT-4).

A.3 Prompt for LLM Experiments

Table 7, Table 8, Table 9 provides an example of the 5-shot prompt for False Dilemma used during our LLM experiments. Instances used for non-zero-shot settings are randomly selected from LOGIC-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, filling the template's placeholders.
Please follow the Output Format!!!
 # 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.]
$[A]$=
$[C]$=
Example1
If you can't prove that Ken had an affair with the nanny, then he's been faithful to his wife.
Template No.=1
$[A]$=prove that Ken had an affair with the nanny
$[C]$=he's been faithful to his wife
Example2
People either like coffee or hate it.
Template No.=2
$[A]$=like coffee
$[C]$=hate
Example4
We cannot support immigrants because we have too many homeless and poor Americans.
Template No.=4
$[A]$=support immigrants
$[C]$=homeless and poor Americans
Example5
The speaker insinuates that there are only two options despite this not being true.
Template No.=5
$[A]$=
$[C]$=
 # Query
{}

Table 7: Natural Language (NL_1): 5-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, filling the template's placeholders.
Please follow the Output Format!!!
List of Templates
Template No.1:
Premise 1: An entity/action [A] promotes a good entity/action [C].
Premise 2: 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: 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: 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: 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.]
$[A]$=
$[C]$=
Example1
If you can't prove that Ken had an affair with the nanny, then he's been faithful to his wife.
Template No.=1
$[A]$=prove that Ken had an affair with the nanny
$[C]$=he's been faithful to his wife
Example2
People either like coffee or hate it.
Template No.=2
$[A]$=like coffee
$[C]$=hate
Example4
We cannot support immigrants because we have too many homeless and poor Americans.
Template No.=4
$[A]$=support immigrants
$[C]$=homeless and poor Americans
Example5
The speaker insinuates that there are only two options despite this not being true.
Template No.=5
$[A]$=
$[C]$=
Query
{}

Table 8: Propositional Logic (PL): 5-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, filling the template's placeholders.
Please follow the Output Format!!!
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, [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, [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, [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, [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.]
$[A]$=
$[C]$=
#Example1
If you can't prove that Ken had an affair with the nanny, then he's been faithful to his wife.
Template No.=1
$[A]$=prove that Ken had an affair with the nanny
$[C]$=he's been faithful to his wife
Example2
People either like coffee or hate it.
Template No.=2
$[A]$=like coffee
$[C]$=hate
Example4
We cannot support immigrants because we have too many homeless and poor Americans.
Template No.=4
$[A]$=support immigrants
$[C]$=homeless and poor Americans
Example5
The speaker insinuates that there are only two options despite this not being true.
Template No.=5
$[A]$=
$[C]$=
Query
{}

Table 9: Natural Language$_2$ (NL$_2$): 5-shot False Dilemma prompt for LLM experiments.