Beyond Logical Forms: LLM-Extracted Patterns for Fallacy Classification

Eleni Papadopulos[1,2], **Firoj Alam**[3] **Giovanni Da San Martino**[2]

[1]Politecnico di Torino, Italy, [2]Università di Padova, Italy

[3]Qatar Computing Research Institute, Qatar

eleni.papadopulos@polito.it, giovanni.dasanmartino@unipd.it

fialam@hbku.edu.qa

Abstract

In today's fast-paced information era, logical fallacies, defined as defective patterns of reasoning, inevitably contribute to the growth of information disorder. However, often fallacies appear in nuanced forms that complicate automated classification. In this study, we investigate whether merging abstract logical structures with context-level linguistic cues proves beneficial for fallacy classification, developing a framework that inductively extracts such patterns from fallacious examples and their explanations using Large Language Models (LLMs). We evaluate the impact of these patterns across different LLMs and experimental zero- and one-shot configurations, showing statistically significant improvements over zero-shot baselines and outperforming competing approaches. Cross-dataset experiments validate generalization, establishing data-driven pattern extraction as an effective method for generating logical representations.

1 Introduction

A logical fallacy is a common thinking error, especially one apt to mislead (Gensler, 2010). These arguments often appear rational and logically coherent on the surface, but deeper analysis reveals they are not (Copi et al., 1953). Fallacies are traditionally classified into formal and informal types: ***formal fallacies*** violate the rules of logical structure regardless of content, while ***informal fallacies*** are patterns of mistakes that are made in the everyday uses of language and are related to contextual meaning (Hamblin, 1970; Bacon et al., 1999).

To evaluate the quality of an argument, it is helpful to reconstruct it into what is known as logical form, the structure that emerges when the specific content of a statement is replaced by variables (Johnson and Blair, 1977). For example, the argument *If it rains, then the ground will be wet. It is raining. Therefore, the ground is wet* has the logical form *If P, then Q. P. Therefore, Q*. Building on this formalization framework, Jin et al. (2022) developed a structure-aware model for fallacy detection on the LOGIC dataset that compares arguments' and fallacies' logical forms. However, in their approach a single logical form is assigned to each fallacy, which might fail to capture the full spectrum of ways a fallacy can manifest in natural discourse. Another challenge is related to informal fallacies, where reasoning is often more nuanced and context-dependent than abstract representations suggest.

These limitations motivate the need to go beyond purely abstract representations, incorporating linguistic elements, such as lexical markers or rhetorical devices, to provide a more comprehensive characterization of how fallacies manifest in natural language. We argue that LLMs can inductively extract such representations from fallacious examples, capturing both the logical structure and linguistic cues that reveal the underlying mechanisms of deception. Unlike prior work that formalized fallacy logic through hand-crafted templates (Robbani et al., 2024), our approach is data-driven and not restricted to a limited number of fallacies. Hereafter, we refer to these extracted structures collectively as *structural patterns*. Our goal is to investigate whether context-aware structural information is valuable for automated fallacy detection.

While existing supervised approaches require heavy computational resources for fine-tuning (Vijayaraghavan and Vosoughi, 2022; Lei and Huang, 2024; Sourati et al., 2023a,b; Alhindi et al., 2024), to our knowledge, no prior work has explored fallacy classification from a structural perspective without any additional fine-tuning. Although we use labeled data for pattern extraction (resulting in a weakly supervised approach), our framework avoids fine-tuning costs and produces generalizable patterns that allow classification through prompting alone, enabling comparison with unsu-

Proceedings of the 13th Workshop on Argument Mining and Reasoning, pages 12–25
July 3, 2026 ©2026 Association for Computational Linguistics

pervised methods.

We evaluate multiple prompting configurations to determine which components enhance performance and examine the impact of demonstrations on detection capabilities. Our approach, incorporating generated patterns, achieves noteworthy results among unsupervised methods on the dataset LOGIC. Finally, to validate the robustness and transferability of our patterns, we assess their performance across two further datasets spanning diverse domains and argumentative styles.

In summary, our contributions are threefold:

- We leverage LLMs to automatically extract patterns from fallacious examples and their explanations, which are then employed in inference-only classification.
- We evaluate different LLMs with various prompt designs outperforming competing approaches on the LOGIC dataset.
- We validate generalizability by testing our patterns on two different datasets with different domains and structures.

2 Related Work

Recent advances in fallacy detection have increasingly turned to LLMs, though few studies have relied exclusively on prompting-based techniques. Several works have employed fallacy detection to probe LLMs' logical reasoning abilities (Teo et al., 2025; Hong et al., 2024; Li et al., 2024; Xu et al., 2026). Among these, Hong et al. (2024) investigated self-verification capabilities and showed that LLMs face more challenges with structure-based (formal) fallacies with respect to content-based (informal) ones, and that fallacy definitions provide minimal improvements. Xu et al. (2026) has shown that reasoning models have better performances with respect to non-reasoning ones for fallacy classification. Among studies relying exclusively on prompting techniques, Pan et al. (2024) designed single-round and multi-round prompting schemes for zero-shot detection, while Jeong et al. (2025) introduced contextual prompting incorporating counterarguments, explanations, and goals with confidence-based ranking, showing that explanations particularly enhance performance. Lim and Perrault (2024) assessed detection abilities on the LOGIC dataset using few-shot prompting, though their different taxonomy limits direct comparison with our work. Other research has examined the logical structure of argumentation. Most notably, Jin et al. (2022) developed a structure-aware model based on Electra that distills arguments into logical forms and compares them against fallacy patterns sourced from logicallyfallacious.com. Another prominent framework in this field is Walton's theory of argumentation (Walton, 2008), consisting in about 60 templates that capture common argument types, each associated with a set of critical questions to evaluate their validity. These schemes have been adopted in computational approaches for disinformation (Gutiérrez-Mandingorra et al., 2024), misinformation (Ruiz-Dolz and Lawrence, 2025) and fake news (Wang et al., 2025) detection. Regarding fallacies specifically, Ruiz-Dolz and Lawrence (2023) introduced a dataset of sentences grounded in Walton's argumentation schemes, labeling them as fallacies when the associated critical questions could not be successfully answered. However, Walton's schemes do not hold a one-to-one correspondence with fallacies and cover only a limited number of them, limiting their direct applicability to fallacy classification. Of particular relevance is the work of Robbani et al. (2024), who re-designed four of Walton (2008) and Reisert et al. (2018)'s schemes with the goal of expliciting fallacies' implicit logic, introducing formal logical schemas with explicit variables and relationships. While this represents a meaningful step toward structural formalization, their patterns are manually designed and leave a large portion of fallacy types unrepresented. Our patterns, by contrast, are extracted automatically from data, allowing to adapt to intra-class variation.

3 Datasets

The LOGIC dataset is a collection of 2,449 examples across 13 fallacy types (Jin et al., 2022). Instances are sourced from educational platforms about fallacies such as Quizziz and study.com. The dataset consists of brief dialogues and short statements. Given the educational intent behind these examples, sentences tend to have relatively straightforward syntactic structures, making the dataset particularly well-suited for the extraction of the patterns.

Although it contains 13 distinct classes, a thorough analysis revealed that some of the classes actually contain instances of different fallacies, that were grouped together. For instance, the class *Hasty Generalization* contains examples of actual

Class	*Fallacies included*
Intentional Fallacy	Intentional Fallacy Shifting the Burden of Proof Moving the Goalposts No True Scotsman
False Cause	Post Hoc False Cause
Hasty Generalization	Hasty Generalization Slippery Slope

Table 1: Examples of classes in LOGIC dataset containing instances of different fallacy types. While coherent, these groupings comprise fallacies with distinct structural patterns. A detailed breakdown of all classes' subtypes is provided in Appendix D of Jin et al. (2022).

Hasty Generalization as well as *Slippery Slope* (Table 1). While these grouped fallacies share common logical flaws and thus belong to the same conceptual group, they manifest through different structural patterns.

We experiment on two further datasets: REDDIT (Sahai et al., 2021), consisting of fallacious comments extracted from subreddits covering different topics and ELECDEBATE60TO16 (hereafter ELECDEBATE) (Goffredo et al., 2023), a collection of televised debates of the presidential election campaigns in the U.S. from 1960 to 2016. Some fallacy classes contain sub-categories. In Table 2, we report a summary of the dataset, and a description of each taxonomy is provided in Appendix B.

Data	Dataset split	# Classes	Genre	Domain
LOGIC	1807/299/299	13	Dialogue	Education
REDDIT	588/148/105	8‡	Comments	General
ELECDEBATE	1120/200/187	6	Dialogue	Politics

Table 2: Statistics of the three datasets. ‡ indicates that the *No Fallacy* class is included.

4 Pattern Generation

Natural arguments appear in several forms. Such variability manifests itself in LOGIC dataset as well as many others (Habernal et al., 2018; Da San Martino et al., 2019). For this reason, we address our research question by modeling patterns inductively from the training set of LOGIC. The choice of the dataset for pattern extraction is critical. It provides the required combination of structural clarity and fallacy diversity through its multiple sub-types per class. These properties make it especially suited for our purpose. The clean argumentative structure allows to formalize clear logical patterns while capturing intra-class variations.

Our pattern generation procedure features two steps:

Step 1: Explanation Generation Explanations have been shown to be instrumental in identifying and discrediting fallacious reasoning, as they make the logical structure of arguments explicit and open to scrutiny (Storer, 1949). Furthermore, Jeong et al. (2025) has demonstrated that providing explanations constitutes valuable contextual information in zero-shot settings. We expected explanations to facilitate pattern extraction by breaking down the reasoning process and revealing shared reasoning flaws, particularly useful for informal fallacies.

Given a sentence from the training set and its fallacy label, we used `llama-3.3-70B-Instruct` (Dubey et al., 2024) to generate an explanation that justifies why that sentence contains the specified fallacy.

Step 2: Pattern Extraction For each fallacy class, we used OpenAI's reasoning model `o4-mini` (OpenAI, 2025) to extract patterns from all the sentences of that class and their explanations, requiring the model to preserve function words such as prepositions or adverbs and to abstract away from content words by replacing them with placeholders while keeping the original reasoning form. Additionally, summaries were extracted to derive new fallacy definitions.

We opted for `llama-3.3-70B-Instruct` for explanation generation as it provided high-quality explanations while remaining cost-effective for large-scale text generation. For pattern extraction, we employed `o4-mini` given its reasoning capabilities. The prompts used in our experiments are reported in the Github repository.[1]

In the initial phase of our research, we aimed to cover two distinct logical aspects from our arguments and explanations, specific to formal and informal fallacies, respectively:

- arguments' **logical structure** inspired by formal logic theory;
- recurring **lexical schemes** that frequently appear in both sentences and explanations, capturing specific information about the reasoning behind the fallacy as well as frequent syn-

[1] https://github.com/elenipapadopulos/fallacy-patterns

tactic particles, phrases, and examples that convey the fallacious intent.

Our patterns incorporate both of these aspects, as Table 3 shows. These LOGIC-based patterns combine reasoning structure (variables X, Y, Z) with concrete linguistic features (specific phrases, loaded terms, rhetorical devices), occasionally retaining some definitions. The full list of patterns is available in the repository.[1]

The process resulted in approximately 3-6 patterns per fallacy class. Final patterns were obtained after providing different subsets to the model and selecting the best performing one on the validation set, in the attempt to retain only useful information and avoid redundancy. In section 3 we discussed how one class in the datasets could correspond to multiple fallacies. Although in some cases, e.g. a pattern for *Tu quoque* (a fallacy which is part of the class *Ad Hominem* in LOGIC), is correctly generated and selected, sometimes fails to select patterns when multiple fallacies are grouped under the same class label. This is expected because we include the fallacy class name in the prompt, which likely biases the model toward patterns that match its internal knowledge of that particular class name. To ensure a broader coverage of fallacies listed in Table 1, we manually isolated instances of frequent and undetected fallacies (such as *Shifting the Burden of Proof*) and repeated the procedure.

Intentional Fallacy Patterns
1. The argument assumes that because X (e.g., someone's intention, belief, or lack of counter-evidence), therefore Y is true.
2. Asserting P is true because it has not been disproven.
3. Because the creator intended [interpretation], the work should be understood as [interpretation].
4. Questions framed to presuppose guilt or a specific intention (e.g., "Have you stopped X?"), thus assuming what is to be proven.
5. If A does not have trait X, and X is allegedly typical of group G, then A is not a member of G.
Red Herring Patterns
1. Instead of addressing [original issue], the argument shifts focus to [irrelevant topic], which distracts from the main discussion.
2. The argument attempts to justify, explain, or defend by referencing [irrelevant detail], ignoring the original issue of [main topic].
3. A shift from the initial question or problem to a secondary topic that does not logically follow, e.g., "You asked about X, but I will tell you about Y."

Table 3: Patterns for *Intentional Fallacy* and *Red Herring*. For *Intentional Fallacy*, patterns (#4) and (#5) illustrate lexical schemes and logical forms, respectively, that encode intent and structure.

5 Experiments

This section describes our experiments for fallacy classification, including our patterns extracted by the procedure introduced in Section 4 and several competing prompting strategies. Additional experiments are reported in Appendix D. We used the following LLMs for our experiments: `gpt-4o`, `o4-mini`, `gpt-4.1-mini`, `LLama-3.3-70B`, `deepseek-r1` and `Gemma-3-27B-it` for a total cost of 75 USD. Our intent was to test LLMs from different providers and with different sizes and to compare reasoning and non-reasoning models.

5.1 Prompt Design

Baselines. We compared our approach against several baselines that vary in the type and amount of information provided to the model. The simplest baseline (**ZERO-SHOT**) provides only the list of fallacy names in the dataset as a reference, establishing a minimal information condition. Our second baseline incorporates fallacy definitions to provide more comprehensive background knowledge (**DEF**). These definitions were initially sourced from Lei and Huang (2024) and subsequently refined based on our analysis to ensure clarity and consistency. Finally, we tested a baseline using standard logical forms, following the approach of Jin et al. (2022) and sourcing these forms from logicallyfallacious.com. This final baseline (**LOGICAL FORMS**) allows us to assess the effectiveness of expert-made logical representations compared to our generated pattern-based approach.

LLM-derived Patterns and Definitions. Beyond generating structural patterns, we leveraged the explanations from Section 4 to automatically create new fallacy definitions based on LOGIC training samples. We then replicated experiment **DEF** with these new definitions (**NEW DEF**). We also exploited the patterns extracted by adding them to the prompt (**PATTERNS**) and by implementing a two-step approach where we first ask the LLM to identify the pattern and then to output the corresponding fallacy (**PATTERN MATCHING**).

One-shot Prompting. We further investigated the impact of providing examples to the model through several experimental configurations (Brown et al., 2020), with one-shot prompting prov-

ing most effective. Initially, we tested a static approach where one example per fallacy was randomly selected and shown to all test sentences (**ONE-SHOT**), establishing a baseline for example-based learning. To enhance this approach, we augmented the same examples with manually crafted explanations following our previously established definitions as guidelines (**ONE-SHOT + EXP**). We sampled 5 different example sets and performance across all configurations was assessed over 5 runs to ensure statistical reliability.

More sophisticated was our dynamic one-shot prompting approach (**DYNAMIC ONE-SHOT**), which computes embeddings for both training and test sentences to retrieve, for each test sentence, the most similar example per class in the training set. We used `sentence-transformers/all-MiniLM-L6-v2`[2] model and `cross-encoder/stsb-roberta-base`[3] cross-encoder from SentenceTransformers (Reimers and Gurevych, 2019) to compute embeddings and employed cosine similarity to evaluate similarity. We included the previously generated explanations of examples in the prompt as well (**DYNAMIC + EXP**).

Furthermore, we explored structure-focused similarity. Since Jin et al. (2022) released a version of LOGIC with masked arguments (with content words replaced by placeholders), we conducted the same similarity-based procedure using these masked sentences (see an example in Table 4) in the attempt to force the embedding model to focus on structural rather than lexical similarities. For this configuration (**SYNTAX-BASED DYNAMIC ONE-SHOT**), we used `sentence-transformers/all-MiniLM-L6-v2` from SentenceTransformers alongside a syntax-augmented version of RoBERTa-large extracted from Sachan et al. (2021) (see Appendix E).

Finally, we incorporated the generated patterns into our dynamically retrieved examples and their explanations (**DYNAMIC + EXP + PATTERNS**).

Multi-step Classification. An alternative approach decomposes the classification task into three sequential steps within a single model call (**MULTISTEP**) using chain-of-thought prompting (Wei et al., 2022). In the first step, the model is required to generate a structural pattern from the argument according to predefined structural rules. Subsequently, the model should match it to one of the patterns and, as a result, classify the argument.

Original argument	Every time I wear this necklace, I pass my exams. Therefore, wearing this necklace causes me to pass my exams.
Masked argument	Every time MSK<0> MSK<2>, MSK<0> MSK<4>. Therefore, MSK<2> causes MSK<0> to MSK<4>.

Table 4: Example of a masked argument in LOGIC. The distillation algorithm is explained in Jin et al. (2022). The masked version of the dataset was publicly released by the authors and was not created by us.

5.2 Results and discussion

Table 5 summarizes all experimental configurations and results on LOGIC. It reveals a consistent improvement when the model leverages information about the underlying logic extracted through the LLMs, especially with reasoning models and `gpt-4o`. When using reasoning models, the model-generated definitions yield a 4.65% accuracy improvement over our manually corrected definitions. In the same way, including our generated patterns causes a 8.2% increase with respect to the logical forms extracted by the website logicallyfallacious.com and used in Jin et al. (2022). McNemar's test proved statistical significance for all models using **PATTERNS** against **ZERO-SHOT** and for all except `llama` and `deepseek` against **LOGICAL FORMS** method. When it comes to non-reasoning models, the new definitions do not really affect the performance, whereas using our patterns improves the accuracy by 5.8% on average. For comparison, we test our method against Robbani et al. (2024)'s templates: our patterns outperform said schemes by an average 10.7% across all models.

A notable result is the performance increase achieved through dynamic one-shot prompting. In particular, **DYNAMIC ONE-SHOT** approach (using `all-MiniLM-L6-v2`) yields an average 8.87% increase in accuracy compared to **ONE-SHOT**, despite relying on semantic similarity for example selection. On the other hand, the syntax-oriented example retrieval strategy (**SYNTAX-BASED DYNAMIC ONE-SHOT**) does not outperform the semantic selection. This may be partially due to inaccuracies in the sentence masking process,

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[3] https://huggingface.co/cross-encoder/stsb-roberta-base

Method	o4-mini		gpt-4o		deepseek-r1		gpt-4.1-mini		llama-3.3-70B		gemma-3-27b-it	
	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Baselines												
ZERO-SHOT	61.7	55.3	62.7	57.0	62.7	57.3	**57.8**	**51.0**	55.8	47.7	60.5	51.3
DEF	62.1	58.7	65.0	58.7	62.2	56.5	57.5	50.6	59.1	51.5	**63.5**	**55.2**
LOGICAL FORMS	**63.2**	**57.4**	**65.4**	**59.4**	**63.1**	**55.4**	57.8	49.4	**60.2**	**51.3**	62.8	53.9
LLM-derived Patterns and Definitions												
NEW DEF	66.8	67.3	66.8	59.9	66.8	60.0	57.5	52.5	58.8	53.3	64.8	57.7
PATTERNS	**72.2**	**66.4**	73.2	64.9	70.5	66.2	63.5	55.7	64.5	53.3	**68.5**	**61.9**
PATTERN MATCHING	70.1	65.9	**73.5**	**66.5**	**71.5**	**66.5**	**65.2**	**57.9**	**66.2**	**59.6**	67.2	59.9
One-shot prompting												
ONE-SHOT	63.6	59.6	64.1	58.7	58.5	55.9	56.2	48.1	56.1	46.2	60.0	49.7
ONE-SHOT + EXP	65.2	59.5	63.5	59.0	45.7	48.6	56.8	50.0	56.3	47.9	59.2	49.7
DYNAMIC ONE-SHOT												
all-MiniLM-L6-v2	70.2	67.6	71.3	66.4	70.4	66.2	65.8	61.7	65.5	59.7	68.5	63.3
roberta-base	69.5	64.3	69.5	64.6	72.5	67.8	65.5	61.3	64.8	58.9	66.5	60.6
SYNTAX-BASED DYNAMIC ONE-SHOT												
all-MiniLM-L6-v2	68.2	63.6	71.2	66.1	68.5	64.2	63.2	58.3	62.8	55.7	64.5	57.3
syntax-augmented roberta-large	65.5	65.5	71.2	66.3	68.5	64.2	64.5	58.6	64.2	56.5	63.5	56.0
DYNAMIC + EXP	71.2	68.9	69.5	65.0	72.7	67.9	**67.8**	**61.0**	**67.5**	**62.2**	68.2	63.2
DYNAMIC + EXP + PATTERNS	**74.2**	**68.9**	**73.1**	**67.2**	**73.2**	**67.9**	66.8	62.3	67.2	55.1	**70.5**	**65.9**
Multi-step classification												
MULTISTEP	65.4	64.9	70.9	62.7	62.2	57.1	65.8	55.8	62.5	55.1	66.8	60.2

Table 5: Fallacy classification performance on LOGIC. **Bold**: best approach in section per model by accuracy, **Bold**: best approach overall per model by accuracy. F_1 score denotes Macro F_1 score, which accounts for the class imbalance in the dataset.

which can negatively impact the retrieval of similar examples and the classification, consequently. The MULTISTEP approach shows weaker performance than PATTERN MATCHING, especially for `deepseek-r1` (DeepSeek-AI et al., 2024), implying that generating logical forms without explicit guidance constitutes the main challenge for the model in the request.

In summary, including context-aware logical patterns proves consistently beneficial for fallacy classification: PATTERNS with `gpt-4o` reaches 73.5% accuracy, outperforming prior unsupervised methods (Table 6), while DYNAMIC+EXP+PATTERNS with `o4-mini` achieves 74.2% when augmented with examples and patterns.

Method	Acc	F_1
Jeong et al. (2025)	49.0	37.0
Pan et al. (2024)	-	50.5
PATTERNS (gpt-4o)	73.5	66.5
DYNAMIC+EXP+PAT. (o4-mini)	74.2	68.9

Table 6: Comparison of our best results against the unsupervised baselines provided by Jeong et al. (2025) and Pan et al. (2024) (described in Appendix C) for LOGIC.

5.3 Error analysis

Pattern matching Requesting the model to identify the closest pattern for each argument provides insight into the association process between sentences and patterns. For our analysis, we have split our fallacies into two groups in Table 7: *(i)* group 1, consisting of fallacies whose patterns include logical forms while still including additional contextual cues; *(ii)* group 2, consisting of fallacies that lack highly structured patterns and rely more on contextual and semantic features of the sentence.

Figure 1 shows consistently superior accuracy

Group 1	Group 2
• Ad Hominem • Ad Populum • Circular Reasoning • Irrelevant Authority • False Cause • Hasty Generalization • Deductive Fallacy • Black-and-White Fallacy	• Red Herring • Equivocation • Emotional Language • Extension Fallacy • Intentional Fallacy

Table 7: Grouped fallacy classes based on pattern features for analytical purposes.

for Group 1, whose classes maintain relatively high performance across all experimental settings. The class *Circular Reasoning* emerges as the most accurately predicted class across all models. For what concerns Group 2, the overall accuracy is, on average, 22% lower with respect to Group 1. The classes *Emotional Language*, *Red Herring* and *Extension Fallacy* achieve moderate prediction accuracy, whereas only *Evading the Burden of Proof*'s patterns within the *Intentional Fallacy* category are correctly classified, and *Equivocation* remains entirely undetected by `gpt-4.1-mini` (OpenAI, 2023). In summary, the models achieve better performance on logical fallacies that exhibit clearer structural characteristics but face difficulties with fallacies requiring more nuanced semantic understanding and contextual analysis.

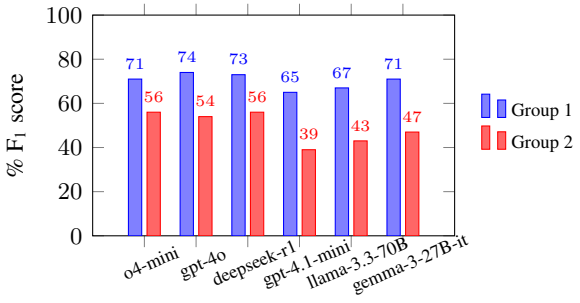

Figure 1: Group-wise F_1 score for each model, relative to the **PATTERN MATCHING** prompt setting.

Furthermore, matching patterns allows us to see that some instances can be deemed as fitting from a structural point of view, thus partially explaining the inherent difficulty of the classification task. While providing guidance through syntactic and logical structure proves beneficial for fallacy detection, this approach does not eliminate all sources of ambiguity, as some sentences may conform to multiple structural patterns. The critical point lies in context-aware pattern application. Models must not only identify logical forms but also evaluate their contextual validity in each sentence.

To quantify the degree of ambiguity inherent in pattern matching, we instructed the best-performing model `o4-mini` to return the five most similar patterns for each argument. This multi-candidate approach enables us to analyze whether lower-ranked patterns might also represent valid interpretations of the same argument. By examining the distribution of pattern similarities and evaluating classification accuracy when considering alternative matches, we can better understand the boundaries of pattern-based classification and identify instances where structural ambiguity genuinely complicates fallacy detection.

Table 9 shows that, when the model is prompted to return multiple matching patterns rather than a single best match, its confidence in the initial prediction decreases, resulting in a 3.4% drop in accuracy (see Table 5).

Acc@1	Acc@2	Acc@3	Acc@4	Acc@5
66.7	75.1	81.8	86.5	88.5

Table 9: Performance analysis in **PATTERN MATCHING** with expanded solution pool: classification results including top 5 predictions as correct.

However, this apparent degradation is misleading when viewed in isolation. By incorporating the second-ranked pattern choice into our evaluation, performance recovers to 75.1%, and continues to improve as we expand our candidate pool to include progressively lower-ranked options. Table 8 illustrates a representative case where the model successfully identifies the correct pattern as its second choice, while its first-ranked selection remains structurally plausible. The model likely assigns one of the *Ad Populum* patterns because it closely matches the argument's logic, while the *Irrelevant Authority* pattern does not fit the sentence since it requires discussion of an unrelated topic, which is not present in the sentence. These subtle distinctions likely make pattern matching more challenging than direct classification because it requires strict structural alignment as well as capturing broader content-related features.

Multistep classification. The **MULTISTEP** approach fails to produce significant results. We conduct this experiment in a single passage to force the model to reason using both semantic and syntactic information. However, classification performance depends critically on the quality of the extracted logical forms, which proves inconsistent and model-dependent. For instance, `o4-mini` embeds classification-relevant contextual information directly into its generated logical forms (Table 10). Furthermore, models demonstrate substantially weaker performance on Group 2 sentences compared to Group 1, showing an average decrease of 21.5% in F_1 score. Additionally, models frequently bypass the pattern matching phase entirely, arbitrarily assigning patterns despite clear misalign-

I have no intention of stopping the use of somatostatin on patients suffering from acute pancreatitis. I consider it to be a very reasonable choice. After all, it has been standard practice in our department for many years and we've been quite satisfied with the results we've had. ***Irrelevant Authority***	
Top 1	Because many people [do/believe/support] X, X must be true/good/right/best/valid. (*Ad Populum*)
Top 2	Using [personal trait, experience, past action] as implicit proof of authority on a distinct or unrelated subject. (*Irrelevant Authority*)

Table 8: Sentence accurately classified by `o4-mini` with the 2^{nd} ranked pattern. Although the model fails initially, the matching can still be deemed reasonable.

Teaching assistant: the homework assignment was much harder than we thought, so I think we should give a few extra points to students who completed it. Professor: that's a terrible idea. If we give everyone a perfect score for no reason, students won't bother working hard in the future. ***Extension Fallacy***	
`o4-mini`	• **Generated Form:** A proposes: "Because B, we should do X_1." Opponent replies: "If we do X_2 (an exaggerated version of X_1) for no reason, then C will happen; therefore, we must not do X_1." • **Matched Pattern:** X proposes Y; response exaggerates Y to an extreme version Y. • **Fallacy:** Extension Fallacy
`gpt-4.1-mini`	• **Generated Form:** A says B was harder than expected, so A suggests giving extra points to those who completed it. B replies: if we give everyone a perfect score for no reason, then C won't work hard; therefore, we should not give extra points. • **Matched Pattern:** Arguing that taking a minor action will cause a chain of escalating events leading to a disastrous outcome, even though no strong evidence supports the inevitability of that chain. • **Fallacy:** Hasty Generalization
`llama-3.3-70B`	• **Generated Form:** If we do A (give everyone a perfect score for no reason), then B (students won't bother working hard in the future). • **Matched Pattern:** Presenting a coincidental or correlated event as proof of causation, often ignoring other plausible causal factors or explanations. • **Fallacy:** False Cause
`gemma-3-27b-it`	• **Generated Form:** If we do A (give extra points), then B (students won't work hard). Therefore, we should not do A. • **Matched Pattern:** If P then Q; Q is true; therefore, P is true. • **Fallacy:** Deductive Fallacy

Table 10: Comparison of outputs from four models evaluated in the **MULTISTEP** configuration on LOGIC.

ment with the extracted logical form. For example, given the argument *People nowadays only vote with their emotions instead of their brains* (an instance of *Hasty Generalization*), the model `o4-mini` first extracts the logical form *All A only do B instead of C*. The model then matches this form to the pattern *Generalizing from a small sample or single event to an entire group or population*, which correctly belongs to *Hasty Generalization*. While this produces an accurate classification, the assigned pattern does not precisely correspond to the extracted logical form. In summary, while humans naturally decompose pattern matching into multiple cognitive steps, this multi-stage process proves to be challenging for current LLMs. Models struggle to bridge the gap between abstract logical patterns and their content-dependent manifestations, often failing to identify the implicit premises and unstated logical connections that underlie the reasoning chain.

6 Experiments on Further Datasets

In order to further assess the quality of LOGIC-derived patterns, we conducted a subset of the experiments on REDDIT and ELECDEBATE using the best performing model, `o4-mini`. We tested pat-

	REDDIT		ELECDEBATE	
	Acc.	**M-F1$_1$**	**Acc.**	**M- F1$_1$**
ZERO-SHOT	82.8	82.8	67.3	50.8
DEF	82.6	82.5	65.9	54.7
LOGICAL FORMS	84.7	84.3	70.7	59.5
PATTERNS	**84.7**	**84.5**	65.5	56.3
PATTERN MATCHING	80.9	80.8	65.5	56.7
DYNAMIC ONE-SHOT	81.9	81.6	**81.7**	**70.4**
DYNAMIC + EXP	83.8	83.6	79.1	71.3
DYNAMIC + EXP + PATTERNS	79.0	78.8	78.8	72.3
SAME-DATASET PATTERNS	83.8	83.4	74.1	64.9
SAME-DATASET PATTERNS MATCHING	84.7	84.3	74.3	63.7

Table 11: Fallacy classification performance using `o4-mini` on REDDIT and ELECDEBATE. **PATTERNS** method involves using patterns generated on LOGIC while **SAME-DATASET PATTERNS** approach includes patterns generated on the datasets REDDIT and ELECDEBATE themselves. Acc.: Accuracy; M-F1: Macro-F1.

terns extracted from LOGIC, restricted to the two datasets' classes (first eight rows in Table 11) and patterns extracted from the datasets themselves (latest two rows in Table 11). REDDIT patterns show a prevalence of linguistic markers over logical forms while ELECDEBATE ones emphasize stronger logical formalization, incorporating symbolic formalism.

Consistent with previous findings, logical pattern incorporation outperforms competing approaches on REDDIT. Moreover, LOGIC-based and REDDIT-based patterns yield comparable results. While taxonomy alignment prevents direct comparison, results from supervised and unsupervised methods (Sahai et al., 2021; Lei and Huang, 2024; Pan et al., 2024; Yeh et al., 2024) are consistent with our findings (see Appendix C). Only a comparison with Lei and Huang (2024) (Macro F_1=81.3%) is possible: **PATTERNS** and **SAME-DATASET PATTERNS** outperform their results.

Regarding ELECDEBATE, Table 11 shows that **DYNAMIC ONE-SHOT** yields the best performance, possibly due to the predominant presence of the class *Emotional Language* (62.5% of test set) whose detection may particularly benefit from similar worded examples. Indeed, **SAME-DATASET PATTERNS** achieve competitive results with respect to Goffredo et al. (2023); Pan et al. (2024) (see Appendix C). These experiments showed a fair generalization of LOGIC-derived patterns on other datasets, with the additional advantage of not requiring labeled data to re-extract the patterns.

In order to prove the broader applicability of our approach beyond LOGIC-specific patterns, we tested patterns generated from the other two datasets on LOGIC.

Table 12 demonstrates solid results, validating findings on LOGIC and proving the robustness and transferability of our pattern-based methodology. Notice that accuracy is not directly comparable with values in Table 5 since REDDIT and ELECDEBATE have a subset of the classes of LOGIC.

	LOGIC$_{\text{REDDIT}}$	LOGIC$_{\text{ELECDEBATE}}$
	Acc.	**Acc.**
SAME-DATASET PATTERNS	90.8	87.1
SAME-DATASET PATTERNS MATCHING	88.3	87.5
LOGIC PATTERNS	89.1	83.7
LOGIC PATTERNS MATCHING	90.0	87.1

Table 12: Fallacy classification performance using `o4-mini` on LOGIC. LOGIC$_X$ refers to LOGIC restricted to the classes from dataset X. **SAME-DATASET PATTERNS** approach includes patterns generated on non-LOGIC dataset X while **LOGIC PATTERNS** involves using LOGIC-derived patterns restricted to the classes of dataset X.

7 Conclusions

Fallacy detection is a challenging yet critical task to solve. Since fallacies often manifest in nuanced and context-dependent forms, purely abstract representations are insufficient to characterize the full spectrum of ways a fallacy can appear in natural language, thus motivating the need to combine logical structure with context-level linguistic cues. We present an experimental framework that inductively extracts context-aware structural patterns from fallacious arguments and their explanations, demonstrating that incorporating such patterns significantly enhances fallacy classification performance. Specifically, pattern-based classification achieves 73.5% accuracy on LOGIC, significantly outperforming prior unsupervised approaches, and 74.2% including one-shot examples. Being data-driven, these patterns are not bound to a fixed set of fallacies and can flexibly capture the diverse nuances through which each fallacy type manifests. Notably, reasoning models demonstrate consistently superior performance across all experimental configurations. Moreover, experiments on additional datasets confirm that the extracted patterns generalize effectively across domains, establishing data-driven pattern extraction as an effective method to generate valid and generalizable logical representations.

8 Limitations

While this work demonstrates the efficacy of large language models in detecting logical fallacies by exploiting the underlying logical structure of sentences, it has several limitations. First, we intentionally generated patterns exclusively from the LOGIC dataset due to the quality and straightforward structure of its sentences. We are aware, however, that it does not fully cover the complex and multi-faceted spectrum of fallacies. Furthermore, our work is based on a small sample of LLMs. Nevertheless, we selected a diverse and representative subset, including models from different providers, with varying sizes and reasoning capabilities.

9 Ethics Statement

Logical fallacies can reinforce societal bias and facilitate the spread of misinformation, leading to harmful consequences for society. This work focuses on leveraging LLMs for detecting logical fallacies in argumentation and should not be employed to manipulate discourse by exploiting identified reasoning patterns. Furthermore, this approach risks amplifying existing LLM biases, potentially causing unfair detection. We acknowledge these limitations and encourage future bias mitigation research. We are aware of the environmental impact of large-scale LLMs usage. However, this study exclusively employs inference-only methods, significantly reducing computational requirements compared to training approaches. All datasets are used in accordance with their license and they have been checked for personally identifying and offensive content.

Acknowledgements

This publication is part of the project PNRR-NGEU, which has received funding from the MUR - DM 629/2024. We would like to thank the Qatar National Research Fund, part of Qatar Research Development and Innovation Council (QRDI), for also funding this work by grant NPRP14C0916-210015.

References

Tariq Alhindi, Smaranda Muresan, and Preslav Nakov. 2024. Large language models are few-shot training example generators: A case study in fallacy recognition.

John B. Bacon, Michael Detlefsen, and David Charles McCarty. 1999. *Logic from A to Z: The Routledge Encyclopedia of Philosophy Glossary of Logical and Mathematical Terms*. Routledge.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners.

Irving Marmer Copi, Carl Cohen, and Kenneth McMahon. 1953. *Introduction to Logic*. Macmillan, New York, NY, USA.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 179 others. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

H.J. Gensler. 2010. *The to Z of Logic*. Number v. 169 in G - Reference,Information and Interdisciplinary Subjects Series. Bloomsbury Academic.

Pierpaolo Goffredo, Mariana Chaves, 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, Singapore. Association for Computational Linguistics.

Ana Gutiérrez-Mandingorra, Stella Heras, and Javier Palanca. 2024. Detecting disinformation through computational argumentation techniques and large language models. In *Proceedings of the 24th Workshop on Computational Models of Natural Argument (CMNA 2024)*, volume 3769 of *CEUR Workshop Proceedings*, pages 46–51.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 386–396, New Orleans, Louisiana. Association for Computational Linguistics.

C.L. Hamblin. 1970. *Fallacies*. University paperbacks. Methuen.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications.

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2024. A closer look at the self-verification abilities of large language models in logical reasoning.

Jiwon Jeong, Hyeju Jang, and Hogun Park. 2025. Large language models are better logical fallacy reasoners with counterargument, explanation, and goal-aware prompt formulation.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. 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.

Ralph Henry Johnson and J. Anthony Blair. 1977. *Logical Self-Defense*. Toronto, Canada.

Yuanyuan Lei and Ruihong Huang. 2024. Boosting logical fallacy reasoning in LLMs via logical structure tree.

Yanda Li, Dixuan Wang, Jiaqing Liang, Guochao Jiang, Qianyu He, Yanghua Xiao, and Deqing Yang. 2024. Reason from fallacy: Enhancing large language models' logical reasoning through logical fallacy understanding.

Gionnieve Lim and Simon T. Perrault. 2024. Evaluation of an llm in identifying logical fallacies: A call for rigor when adopting llms in hci research.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2025. Openai o3 and o4-mini system card.

Fengjun Pan, Xiaobao Wu, Zongrui Li, and Anh Tuan Luu. 2024. Are LLMs good zero-shot fallacy classifiers?

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

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

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*, pages 1–10, Singapore. Association for Computational Linguistics.

Ramon Ruiz-Dolz and John Lawrence. 2025. An explainable framework for misinformation identification via critical question answering. *Preprint*, arXiv:2503.14626.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information?

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.

Thomas Storer. 1949. Carl g. hempel and paul oppenheim. studies in the logic of explanation. philosophy of science, vol. 15 (1948), pp. 135–175. *Journal of Symbolic Logic*, 14(2):133–133.

Nicole Teo, Donghao Huang, Erik Cambria, and Zhaoxia Wang. 2025. Large language models for logical fallacy detection. In *Trends and Applications in Knowledge Discovery and Data Mining*, pages 387–398, Singapore. Springer Nature Singapore.

Prashanth Vijayaraghavan and Soroush Vosoughi. 2022. TWEETSPIN: Fine-grained propaganda detection in social media using multi-view representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3433–3448, Seattle, United States. Association for Computational Linguistics.

Douglas N. Walton. 2008. *Informal logic : a pragmatic approach*, second edition. edition. Cambridge University Press, Cambridge.

Xiaoou Wang, Elena Cabrio, and Serena Villata. 2025. When automated fact-checking meets argumentation: Unveiling fake news through argumentative evidence. *Argument & Computation*, 16(3):405–424.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models.

Zihao Xu, Junchen Ding, Yiling Lou, Kun Zhang, Dong Gong, and Yuekang Li. 2026. Socrates or smartypants: Testing logic reasoning capabilities of large language models with logic programming-based test oracles.

Min-Hsuan Yeh, Ruyuan Wan, and Ting-Hao Kenneth Huang. 2024. CoCoLoFa: A dataset of news comments with common logical fallacies written by LLM-assisted crowds.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training.

A Implementation Details

In experiments where the task consisted of returning only the fallacy label, we set the temperature to 0, with the exception of `o4-mini`, `gpt-4o` and `deepseek-r1`. In all other experiments, the standard configuration was kept. Multiple prompt configurations were evaluated for each approach.

B Fallacy Datasets

B.1 Logic

The dataset LOGIC (Jin et al., 2022) contains the following 13 fallacy classes: *Faulty Generalization (Hasty Generalization)*, *Ad Hominem*, *Ad Populum*, *Circular Claim (Circular Reasoning)*, *False Cause (False Causality)*, *Appeal to Emotion (Emotional Language)*, *Fallacy of Relevance (Red Herring)*, *Deductive Fallacy*, *Intentional Fallacy*, *Fallacy of Extension (Extension Fallacy)*, *False Dilemma (Black-and-White Fallacy)*, *Fallacy of Credibility (Irrelevant Authority)* and *Equivocation*. The names in the parentheses are the actual names used in our experiments.

REDDIT	ELECDEBATE
• Ad Populum	• Ad Hominem
• Irrelevant Authority	• Irrelevant Authority
• Hasty Generalization	• Emotional Language
• Slippery Slope	• Slippery Slope
• Black-and-White Fallacy	• False Cause

Table 13: Fallacy classes in REDDIT and ELECDEBATE used in our experiments.

B.2 Reddit

The dataset REDDIT (Sahai et al., 2021) contains 8 fallacy classes: *Appeal to Authority (Irrelevant Authority)*, *Appeal to Majority (Ad Populum)*, *Appeal to Nature*, *Appeal to Tradition*, *Appeal to Worse Problems*, *Black-and-White fallacy*, *Hasty Generalization* and *Slippery Slope*. It contains the class *No Fallacy* as well. The names in parentheses are the actual labels used. In our experiments, only the classes included in LOGIC are retained (Table 13). We can keep the class *Slippery Slope* because two generated patterns for *Hasty Generalization* correspond to it.

B.3 ElecDebate

The dataset ELECDEBATE (Goffredo et al., 2023) contains the following 6 fallacy classes: *Ad Hominem*, *Appeal to Emotion (Emotional Language)*, *Appeal to Authority (Irrelevant Authority)*, *Slippery Slope*, *False Cause* and *Slogan*. The names in parentheses are the actual labels used. In our experiments, only the classes included in LOGIC are retained (Table 13).

C Baselines

We consider only the classes of REDDIT and ELECDEBATE in common to LOGIC. For this reason, direct comparison with prior work is generally not possible. However, for REDDIT, Lei and Huang (2024) provide classwise F_1 scores, allowing us to compute Macro F_1 and compare our results. Tables 14 and 15 present the comparison with prior work for both datasets.

D Additional Experiments

We are going to report some other experimental setups that have been explored, including some basic baselines that we have not included in Section 5.

Method	Macro F_1
Supervised	
Sahai et al. (2021)	58.4
Lei and Huang (2024)†	81.3
Pan et al. (2024)	83.2
Unsupervised	
Pan et al. (2024)	81.1
Yeh et al. (2024)	81.0
Ours	
PATTERNS	**84.5**
SAME-DATASET PATTERN MATCHING	**84.3**

Table 14: Performance comparison on REDDIT.† indicates that Macro F_1 is computed on the exact same classes as LOGIC.

Method	Macro F_1
Supervised	
Goffredo et al. (2023)	**73.9**
Pan et al. (2024)	62.3
Unsupervised	
Pan et al. (2024)	44.5
Ours	
SAME-DATASET PATTERN	64.9
DYNAMIC ONE-SHOT	70.4

Table 15: Performance comparison on ELECDEBATE.

D.1 Prompt design

- **EXP**: to investigate whether explicit reasoning improves performance, we implemented a baseline that not only provides fallacy names but also requests the model to generate a two-sentence explanation for its classification decision, testing whether forcing the model to articulate its reasoning leads to better outcomes. The two-sentence constraint was intentionally designed to keep explanations concise and manageable for manual inspection of explanations.

- **GUIDELINES**: to leverage the model's classification errors for improvement, we develop guidelines derived from observed mistakes. We conduct pattern matching evaluation on the validation set and collect misclassified instances. For each class, we provide the model with incorrectly classified examples and prompt it to generate comprehensive detection guidelines (as can be seen from table 16), given our generated pattern as a reference. These guidelines are then adopted to evaluate the test set. Notably, only guidelines produced by `o4-mini` and partially by `gpt-4.1-mini` incorporate a little structural and logical information such as common connectors or logical forms while the majority of guidelines content across models focuses primarily on semantic characteristics rather than structural patterns.

Fallacy	Irrelevant Authority
Core definition	A fallacy that treats an individual's status, title, or popularity as proof of a claim when their expertise or relevance to the topic is absent or insufficient.
Key indicators	Argument rests on "X says so" without independent support. Authority cited has no recognized expertise in the claim's domain. No substantive evidence beyond the authority's endorsement.
Typical confusion patterns	Ad Populum: group popularity vs. single authority endorsement. Appeal to Tradition: "has always been done by experts" vs. citing irrelevant experts. Equivocation: shifting word senses vs. relying on irrelevant credentials.

Table 16: Guidelines relative to the *Irrelevant Authority* fallacy generated by `o4-mini`.

Model	EXP		GUIDELINES	
	Acc.	F_1	Acc.	F_1
`o4-mini`	61.3	61.5	65.5	65.7
`gpt-4o`	60.4	53.3	62.6	56.6
`deepseek-r1`	61.8	54.8	63.5	57.9
`gpt-4.1-mini`	57.5	57.9	60.5	60.6
`llama-3.3-70B`	56.1	56.5	52.8	53.3
`gemma-3-27b-it`	59.1	60.8	58.8	59.4

Table 17: Logical fallacy classification performance on additional experiments. F_1 denotes Macro-F_1.

D.2 Results

EXP's (Table 17) results show that requesting the model to articulate the reasoning does not really cause any improvement. Specifically, certain classes such as *Intentional Fallacy* and *Extension Fallacy* exhibit extremely low F_1 scores under the non-reasoning models (0.027 and 0.13 respectively on average), indicating performance

Text	Explanation	Gold
The Bible is true because God exists, and God exists because the Bible says so.	The argument uses its conclusion as a premise, creating a logical loop without independent evidence. ***Circular Reasoning***	***Circular Reasoning***
My friend said that if you sneeze more than three times, you have the corona virus.	The argument assumes sneezing three times indicates the virus, generalizing a symptom without considering other causes. ***Hasty Generalization***	***Irrelevant Authority***

Table 18: Examples from `GPT-4.1-mini` in the **EXP** setting: the first is correctly classified; the second is misclassified because the explanation, while coherent, fails to capture the underlying fallacy.

deterioration compared to the **ZERO-SHOT** baseline. This proves that models process surface-level semantic patterns without being able to access the multi-layered intentional structures behind reasoning (Table 18).

Including **GUIDELINES** yields only modest results. While these guidelines are designed to provide comprehensive fallacy knowledge, they appear to lack the appropriate type of information from which models can benefit. Indeed, providing explicit information about the underlying logical structure proves significantly more beneficial for model performance.

E Syntax-augmented roBERTa

Sachan et al. (2021) introduces a syntax-augmented model that incorporates dependency tree information into pre-trained BERT-based (Devlin et al., 2019) transformers through specialized Graph Neural Networks (GNNs) (Hamilton et al., 2017) that process dependency trees. The authors introduce two distinct fusion strategies to integrate syntactic structure into BERT representation. We adopted specifically roBERTa-large (Zhuang et al., 2021) in the attempt to perform a syntax-driven examples selection. Further details about the implementation are available in Sachan et al. (2021).