Boosting Logical Fallacy Reasoning in LLMs via Logical Structure Tree

Yuanyuan Lei Ruihong Huang
Department of Computer Science and Engineering
Texas A&M University, College Station, TX
{yuanyuan, huangrh}@tamu.edu

Abstract

Logical fallacy uses invalid or faulty reasoning in the construction of a statement. Despite the prevalence and harmfulness of logical fallacies, detecting and classifying logical fallacies still remains a challenging task. We observe that logical fallacies often use connective words to indicate an intended logical relation between two arguments, while the argument semantics does not actually support the logical relation. Inspired by this observation, we propose to build a logical structure tree to explicitly represent and track the hierarchical logic flow among relation connectives and their arguments in a statement. Specifically, this logical structure tree is constructed in an unsupervised manner guided by the constituency tree and a taxonomy of connectives for ten common logical relations, with relation connectives as non-terminal nodes and textual arguments as terminal nodes, and the latter are mostly elementary discourse units. We further develop two strategies to incorporate the logical structure tree into LLMs for fallacy reasoning. Firstly, we transform the tree into natural language descriptions and feed the textualized tree into LLMs as a part of the hard text prompt. Secondly, we derive a relation-aware tree embedding and insert the tree embedding into LLMs as a soft prompt. Experiments on benchmark datasets demonstrate that our approach based on logical structure tree significantly improves precision and recall for both fallacy detection and fallacy classification.

1 Introduction

Logical fallacy refers to the use of invalid or flawed reasoning in an argumentation [**?**, **?**, **?**]. Logical fallacy can occur as unintentional mistakes or deliberate persuasions in a variety of human communications, such as news media [**?**], educational essay [**?**], political debates [**?**, **?**], or online discussions [**?**]. Logical fallacies can lead to harmful consequences for society, such as spreading misinformation [**?**, **?**], raising public health risks [**?**], manipulating public opinions [**?**, **?**, **?**], introducing societal bias and polarization [**?**]. Despite their prevalence and harmfulness, understanding logical fallacies still remains a challenging task, which requires both semantics understanding and logical reasoning [**?**, **?**]. In this paper, we focus on fallacy detection and classification, and aim to develop an approach that generalizes across different domains and genres.

The key observation is that logical fallacies heavily rely on connective phrases to indicate an intended logical relation between two textual arguments, while the semantics of the arguments do not actually support the claimed logical relation. Figure 1 shows two examples where the connective phrases were bolded. The first example uses the connective words *therefore* and *cause* to suggest a causal relation between vaccinations and increasing flu cases, however, the temporal relation between the two events as stated in the first half of the statement does not necessarily entail a causal relation between them, and indeed, their semantics do not actually support the suggested causal relation. Recognizing this discrepancy undermines the credibility of the whole statement. Similarly in the second example, the connective word *likewise* is commonly used to indicate an analogy relation, however, the second argument is clearly a specific case of the general condition stated in the first argument and therefore there is no analogy relation between them, and recognizing this mismatch between the suggested logical relation and the real relation enables us to detect this fallacy.

Therefore, we propose to construct a logical structure tree that organizes all connective phrases in a statement and their textual arguments into a hierarchical structure. We expect the logical structure tree to effectively capture the juxtaposition of connective phrase suggested logical relations and the real logical relations between textual arguments, and therefore guide LLMs in fallacy detection and classification. Specifically, a logical structure tree consists of relation connectives as non-terminal nodes and textual arguments as terminal nodes, and the latter mostly corresponds to elementary discourse units (EDU) considered in discourse parsing. Figure 1 shows the logical structure trees constructed for the two example texts.

As the logical relation indicated by a connective phrase may not be supported by semantics of its arguments in the context, we identify the purposefully indicated logical relations in a context-free unsupervised manner by matching a connective phrase with a taxonomy of connectives compiled for ten common logical relations (conjunction, alternative, restatement, instantiation, contrast, concession, analogy, temporal, condition, causal). To construct a logical structure tree, we first construct a constituency tree for a statement and then search in the constituency tree for connective phrases in the top-down left to right order, and the first found connective phrase will be the root node of the logical structure tree. Next, we identify the text spans of its two arguments using rules and recursively build the left and right sub-trees by applying the same procedure to constituency tree segments corresponding to the two arguments.

The logical structure tree is integrated into LLMs for fallacy reasoning using two strategies. The first considers textualized tree, where we convert the tree into natural language descriptions, making the tree readable by LLMs. Particularly, we describe the relations and arguments in a bottom-up manner, providing the LLMs with insight into logical relations from a local to global perspective. We then concatenate the textualized tree with the instruction prompt, and input them into LLMs as a hard prompt. The second considers tree-based soft prompt, where we derive a relation-aware tree embedding. Specifically, we design relation-specific encoders to process each type of relation and incrementally derive the tree embedding from bottom up to the root node. We then insert the tree embedding into LLMs as a soft prompt for further tuning. Experiments on benchmark datasets across various domains and genres validate that our approach based on logical structure tree effectively improve precision and recall for both fallacy detection and fallacy classification tasks. Our main contributions are summarized as follows:

We propose to construct a logical structure tree to capture the juxtaposition of connective phrase suggested logical relations and the real logical relations between textual arguments, and use it to serve as additional guidance for fallacy detection and classification. We effectively improve the F1 score for fallacy detection by up to 3.45% and fallacy classification by up to 6.75% across various datasets.

2 Related Work

2.1 Logical Fallacy

Logical Fallacy is erroneous patterns of reasoning [?, ?]. Initial work explored the taxonomy of fallacies [?, ?, ?]. Recent works have focused on the automatic detection and classification of fallacies. Habernal *et al.* (2017) developed a software that deals with fallacies in question-answering. Sheng *et al.* (2021) investigated ad hominem fallacy in dialogue responses. Habernal *et al.* (2018) explored the ad hominem fallacy from web argumentations. Stab and Gurevych (2017) recognized insufficient arguments in argumentation essays. Goffredo *et al.* (2022) categorized fallacies in political debates. Nakpih and Santini (2020) focused on fallacies in legal argumentations. Musi *et al.* (2022) researched fallacies about pandemics on social medias. Alhindi *et al.* (2022) proposed a multi-task prompting approach to learn the fallacies from multiple datasets jointly. Jin *et al.* (2022) proposed a structure-aware method to classify fallacies. Different from Jin *et al.* (2022) that masked out content words to form a sequence-based pattern, our paper proposes a tree-based hierarchical logical structure to unify both relation connectives and content arguments together.

2.2 Logical Reasoning of Large Language Models

Logical Reasoning abilities of large language models are gaining increasing research attention [?, ?, ?, ?, ?, ?, ?, ?]. Olausson *et al.* (2023) combined large language models with first-order logic. Pan *et al.* (2023); Zhang *et al.* (2023) empowered large language models with symbolic solvers. Pi *et al.* (2022) presented an adversarial pre-training framework to improve logical reasoning. Zhao *et al.* (2023) incorporated multistep explicit planning into the inference procedure. Jiao *et al.* (2022) proposed a contrastive learning approach to improve logical question-answering. Different from these previous work, we particularly focus on logical fallacy reasoning, aiming to detect and classify fallacies.

2.3 Misinformation

Misinformation refers to the unverified or false information [?, ?, ?, ?]. Misinformation detection was studied for years, such as fake news [?, ?, ?], rumor [?, ?], satire [?], political bias [?, ?, ?, ?], propaganda [?, ?, ?]. Logical fallacies are often employed within misinformation to present invalid claim as credible, facilitating the spread of misinformation [?, ?, ?]. Developing automatic models to detect logical fallacies can also benefit the identification and mitigation of misinformation.

3 Logical Structure Tree

The logical structure tree consists of relation connectives as non-terminal nodes, and textual arguments as terminal nodes. The relation connectives serve as parent nodes, and the two corresponding arguments are linked as left and right children nodes. Figure 1 illustrates examples of the logical structure tree. The logical structure tree is constructed in an unsupervised manner, guided by the constituency tree and a taxonomy of connectives complied for ten common logical relations.

3.1 Relation Connectives

The logical fallacies usually rely on relation connectives to indicate a logical relation. Inspired by the discourse relations proposed by Prasad *et al.* (2008), we define a taxonomy of ten logical relations which are commonly seen: conjunction, alternative, restatement, instantiation, contrast, concession, analogy, temporal, condition, and causal relations. Moreover, we build a set of connective words and phrases that correspond to each type of logical relation, as shown in Table 1.

Table 1: The ten types of logical relations and their relation connectives.

Logical Relations	Relation Connectives
conjunction	and, as well as, as well, also, separately
alternative	or, either, instead, alternatively, else, nor, neither
restatement	specifically, particularly, in particular, besides, additionally, in addition, moreover, furthermore, plus, not only, indeed,
instantiation	for example, for instance, such as, including, as an example, as an instance, for one thing
contrast	but, however, yet, while, unlike, rather, rather than, in comparison, by comparison, on the other hand, on the contrary,
concession	although, though, despite, despite of, in spite of, regardless, regardless of, nevertheless, nonetheless, even if, even thou
analogy	likewise, similarly, as if, as though, just as, just like, namely
temporal	during, before, after, when, as soon as, then, next, until, till, meanwhile, in turn, meantime, afterwards, simultaneously
condition	if, as long as, unless, otherwise, except, whenever, whichever, once, only if, only when, depend on
causal	because, cause, as a result, result in, due to, therefore, hence, thus, thereby, since, now that, consequently, in conseque

This set of connectives includes the explicit discourse connectives from the PDTB discourse relation dataset [**?**], and is further expanded by manually adding relevant connectives from the development set of the logic fallacy dataset [**?**].

We further conduct a statistical analysis on the distribution of ten logical relations and compare distributions between fallacy and no fallacy classes as well as across different fallacy classes, with the detailed results shown in Appendix A. The statistical analysis shows that both the fallacy and no fallacy classes contain many connective phrases and their distributions of the ten logical relations are also very similar. But as expected, different fallacy types tend to employ varying logical patterns, for example, False Dilemma uses more alternative relation, while Deductive Fallacy uses more analogy relation.

3.2 Tree Construction Algorithm

To construct a logical structure tree T_{logic}, we first construct a constituency tree T_{con} for a statement. We use the stanza library to get the constituency tree [**?**]. At the beginning, T_{logic} is initialized as an empty tree. Then we traverse the constituency tree T_{con} from top to bottom and from left to right, and match relation connectives within each subtree of T_{con}. If there is a subtree $S_{con(w)}$ whose text equals to a relation connective w, we use the algorithm in section 3.3 to extract the two textual arguments α, β associated with w. Then a new logical subtree $S_{logic(w)}$ is created, with the matched relation connective w as a parent node, and the two arguments α, β as its left and right children. This new logical subtree $S_{logic(w)}$ is added into the logical structure tree T_{logic}. If the textual arguments α, β still contain other relation connectives, then we recursively match relation connectives in the arguments and replace the original argument node in the T_{logic} with the newly created logical subtree. The termination condition is that all the relation connectives in the given text have been matched.

3.3 Textual Arguments Extraction

The textual arguments are the two content components linked by a relation connective. Given a matched relation connective w, its corresponding subtree in the T_{con} is $S_{con(w)}$. To extract the arguments of w, we find the parent tree of $S_{con(w)}$ in the T_{con}, denoted as $P(S_{con(w)})$. The text enclosed by $P(S_{con(w)})$ is the concatenation of all its leaf node texts. If the text enclosed by parent tree $P(S_{con(w)})$ contains content before and after the relation connective w, i.e., has the form of $\alpha + w + \beta$, then the left argument of w is α, and the right argument is β. If the text enclosed by parent tree $P(S_{con(w)})$ only contains content after the relation connective w, i.e., has the form of $w + \beta$, then the right argument of w is β, and the left argument α is the text enclosed by grandparent tree $P(P(S_{con(w)}))$ subtracted by the text enclosed by $P(S_{con(w)})$.

4 Logical Fallacy Reasoning

We further design a framework to incorporate the logical structure tree into LLMs for fallacy detection and classification. This framework consists of two main components. The first is textualized tree, where we convert the logical structure tree into natural language descriptions, and feed it into LLMs as a hard text prompt. The second is tree-based soft prompt, where we derive a relation-aware tree embedding, and insert it into LLMs as a soft prompt for additional tuning. The hard and soft prompts are complementary: the hard prompt enriches the instruction with logical structure information, while the soft prompt facilitates direct tuning on tree embeddings. Figure 2 shows an illustration.

4.1 Textualized Tree

The textualized tree aims to transform the logical structure tree into the textual form, which can be interpretable by LLMs. As shown by the upper path of Figure 2, the textualized tree is represented as a table which consists of three columns: left argument, relation connective, right argument. Each row in the table represents a triplet (left argument, relation connective, right argument) corresponding to each logical relation in the tree. In particular, we organize the triplets into the table in a bottom-up order, to provide the LLMs with insight into logical relations from a micro to macro perspective. The textualized tree is then input into the LLMs as a part of the hard text prompt:

$$h_t = \text{TextEmbedder}\left(\text{textualize}(T_{logic})\right) \quad (1)$$

where textualize $(\cdot)$ denotes the textualization operation, TextEmbedder refers to the text embedding layer of LLMs, h_t is the mapped embedding of the textualized tree.

4.2 Tree-based Soft Prompt

The tree-based soft prompt is a tree embedding which is projected into LLMs as a soft prompt for further tuning. As

Table 2: The number of samples in train/dev/test set, the number of fallacy and no fallacy (benign) samples, and the number of fallacy types in each dataset.

Dataset	Train	Dev	Test	Fallacy
Benign	Types			
ArgoTario	863	201	267	909
422	5			
Reddit	2313	668	335	1691
1625	8			
Climate	436	114	133	477
206	9			
Logic	1849	300	300	249
-	13			

shown by the lower path of Figure 2, this process includes a tree encoder to derive the tree embedding, as well as a projection layer to transform the tree embedding into the same representation space of LLMs.

During the tree encoder stage, we aim to derive a relation-aware tree embedding. To integrate relation information into tree embedding, we design relation-specific encoders to process each type of logical relation. For a simple tree whose children nodes are leaf nodes without hierarchical layers, its embedding is computed as:

$$e_s = W^r(e_l \oplus e_c \oplus e_r) + b^r \quad (2)$$

where e_s is the embedding of this simple tree, e_l e_c e_r are the embeddings of left argument, relation connective, and right argument, which are initialized as the average of word embeddings derived from RoBERTa language model [?], $\oplus$ denotes feature concatenation, W^r, b^r are the trainable parameters of the encoder that corresponds to the relation type r, where $W^r \in R^{3d \times d}$ $b^r \in R^d$, and $d = 768$ is the dimension of embedding space in RoBERTa. The relation type r is one of the ten logical relations associated with the relation connective.

For the tree with hierarchical structure, we derive the tree embedding incrementally, starting from the bottom simple tree and up towards the root node:

$$e_t = W^r(\hat{e}_l \oplus e_c \oplus \hat{e}_r) + b^r \quad (3)$$

where e_t is the tree embedding, $\hat{e}_l$ is the embedding of the left subtree, $\hat{e}_r$ is the embedding of the right subtree, e_c is the connective embedding.

During the projection stage, we transform the tree embedding e_t into the same representation space of LLMs through a projection layer, which includes two layers of neural networks:

$$\hat{e}_t = W_2\left(W_1 e_t + b_1\right) + b_2 \quad (4)$$

where W_1, W_2, b_1, b_2 are the trainable parameters of the projection layer, $W_1 \in R^{d \times d'}$ $W_2 \in R^{d' \times d'}$ $b_1, b_2 \in R^{d'}$ d is dimension of hidden states in RoBERTa, d' is the dimension of embedding space of the target LLM. $\hat{e}_t$ is the resulting tree-based soft prompt, which is then inserted into LLMs as a token representation within the input sequence.

4.3 Fallacy Training

The LLMs take the instruction prompt, textualized tree h_t, and tree-based soft prompt $\hat{e}_t$ as input, and generate fallacy label as output. The loss is calculated between the generated text and golden label. The text embedding layer and self attention layers of LLMs are frozen. The tree-based soft prompt $\hat{e}_t$ receives gradients and enables back propagation.

5 Experiments

5.1 Datasets

We experiment with four datasets from various domains and genres. Table 3 shows their statistics.

Argotario [?] collects fallacies from the general domain question-answering pairs. The dataset includes the following fallacy labels: Ad Hominem, Appeal to Emotion, Hasty Generalization, Irrelevant Authority, Red Herring, and No Fallacy. We use this dataset for both fallacy detection and classification experiments, and follow the dataset splitting method in Alhindi *et al.* (2022).

Reddit [?] collects user generated posts from Reddit, and annotates logical fallacies into: Slippery Slope, Irrelevant Authority, Hasty Generalization, Black-and-White Fallacy, Ad Populum, Tradition Fallacy, Naturalistic Fallacy, Worse Problem Fallacy, and No Fallacy. This dataset is used for both fallacy detection and classification.

Climate [?] collects statements from articles in the climate change domain, and annotated the following fallacies: Evading the Burden of Proof, Cherry Picking, Red Herring, Strawman, Irrelevant Authority, Hasty Generalization, False Cause, False Analogy, Vagueness, and No Fallacy.

Logic [?] annotates logical fallacies in the educational materials into 13 types including Ad Hominem, Ad Populum, False Dilemma, False Cause, Circular Reasoning, Deductive Fallacy, Appeal to Emotion, Equivocation, Fallacy of Extension, Faulty Generalization, Intentional Fallacy, Fallacy of Credibility, Fallacy of Relevance. This dataset does not include No Fallacy class and is only used for fallacy classification.

5.2 Experimental Settings

To validate our approach, we experiment on two types of language models: a decoder-only model and an encoder-decoder model. For the decoder-only model, we choose the open-source large language model Llama-2 (llama-2-7b-chat-hf) [?]. For the encoder-decoder model, we choose the Flan-T5-large model [?]. Both the models are trained in a generative setting, where they take the instruction and given text as input, and generate a fallacy label as output. The fallacy detection task generates "Yes" or "No" label as output, while the fallacy classification task generates the name of each fallacy type. We follow Alhindi *et al.* (2022) to unify the different names of the same fallacy across datasets, such as False Dilemma is converted into Black-and-White Fallacy since they are the same fallacy. We also follow Alhindi *et al.* (2022) to feed the definitions of each fallacy type into the instruction prompt. The details of instruction prompt are explained in Appendix B. The maximum input length is set to be 1024, number of epochs is 10, weight decay is 1e-2, the gradient accumulation step is 4, learning rate for Llama-2 is 3e-4, and learning rate for Flan-T5 is 3e-5. The Llama-2 model is trained with LoRA [**?**], with rank 8, alpha 16, dropout 0.05, and trainable modules include q_proj and v_proj.

5.3 Baselines

We compare our models with the baselines listed below. Besides the existing baselines, we also implement several additional baselines based on the GPT and RoBERTa [**?**] models:

Sahai *et al.* (2021): a multi-granularity network is designed that trains sentence-level representation and the token-level representations jointly.

Jin *et al.* (2022): a structure-aware framework is developed that forms a sequence-based logical pattern for each text by masking out the content words.

Sourati *et al.* (2023b): a prototype-based reasoning method that injects background knowledge and explainable mechanisms into the language model.

Sourati *et al.* (2023a): a case-based reasoning that retrieves similar cases from external sources based on goals, counterarguments, and explanation etc.

Alhindi *et al.* (2022): a multi-task instruction tuning framework that learns the logical fallacies from multiple datasets collaboratively.

GPT-3.5: we prompt the gpt-3.5-turbo model to automatically choose one of the fallacy labels for each text, and the prompt is listed in Appendix C.

GPT-3.5 + T_{logic}: guide the gpt-3.5-turbo model to firstly reason the logical structure of each text, and then choose one of the fallacy labels through a chain-of-thought process [**?**].

RoBERTa: the RoBERTa model is used to encode the text and the average of word embedding is used as the text embedding. A classification head is built on top of the text embedding to classify labels.

RoBERTa + T_{logic}: we concatenate the text embedding with the logical structure tree embedding, and build classification head on top of the combined embedding to predict labels. The tree embedding is derived based on the method in Section 4.2.

5.4 Fallacy Detection

The fallacy detection task identifies whether a given text contains logical fallacy or not, which is a binary classification task. The precision, recall, and F1 score of the fallacy class, as well as the micro F1 score (i.e., accuracy) are used as evaluation metrics. Table 2 presents the performance on the ArgoTario, Reddit, and Climate datasets.

The results demonstrate that incorporating the logical structure tree effectively improves both precision and recall for logical fallacy detection. This observation is consistent for both types of Llama-2 and Flan-T5 models across all the three datasets, which span various domains and genres. Compared to the baselines that lack logical structure information, our approach based on the logical structure tree noticeably enhances the precision and recall, leading to the F1 score increased by up to 3.45%. This indicates that the logical structure tree is effective in capturing the difference in logical flows between fallacious and benign texts.

Moreover, informing the large language model GPT-3.5-turbo of logical structure information significantly improves fallacy detection under the zero-shot setting, resulting in a substantial improvement in the F1 score. This underscores the importance of integrating the logical structure information into LLMs for fallacy detection. Also, concatenating the logical structure tree embedding with the text embedding in the RoBERTa model also enhances the performance, which proves the usefulness of this logical structure tree embedding. Overall, incorporating the logical structure tree helps improve fallacy detection for various types of models.

5.5 Fallacy Classification

The fallacy classification task classifies the fallacy types for the fallacious text, which is a multi-class classification task excluding the No Fallacy class. The macro precision, recall, and F1 score, as well as the micro F1 score (i.e., accuracy) are used as evaluation metrics. Table 4 shows the results on the ArgoTario, Reddit, and Logic datasets.

The results demonstrate that integrating the logical structure tree into Llama-2 and Flan-T5 models notably enhances the performance of fallacy classification, with both precision and recall increased. This conclusion is valid across the three datasets from different domains and genres. Compared to the baselines without logical structure tree, our proposed approach significantly improves precision and recall, leading to an increase of up to 6.75% in the F1 score. This suggests that the logical structure tree effectively distinguishes the different logical patterns used in each fallacy type, and is applicable across various domains and genres.

In addition, our approach based on the logical structure tree outperforms the previous methods that may lack logical relations information. This highlights the necessity to infuse the logical relations into LLMs for fallacy classification. Besides, our approach achieves higher performance than the baselines that overlook content words. This indicates that analyzing content words also plays an essential role in fallacy reasoning. The logical structure tree connects the logical relations and

Table 3: The results of logical fallacy detection on three datasets. The precision, recall, F1 score of fallacy class, and accuracy are reported. The rows "+ T_{logic} " represent incorporating the logical structure tree into the model.

	ArgoTario				Reddit				Climate			
	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc
Baselines												
Sahai *et al.* (2021)	–	–	–	–	69.57	69.27	69.20	–	–	–	–	–
GPT-3.5	92.86	14.61	25.24	41.67	54.17	15.38	23.96	50.00	70.00	7.61	13.72	33.83
GPT-3.5 + T_{logic}	74.72	75.55	75.14	66.16	58.26	82.94	68.45	60.61	72.45	77.17	74.74	63.91
RoBERTa	81.18	83.42	82.29	75.65	65.00	76.02	70.08	66.86	67.77	89.13	76.99	63.16
RoBERTa + T_{logic}	83.87	86.19	85.01	79.40	67.31	81.87	73.88	70.45	68.22	95.65	79.64	66.16
Flan-T5	81.91	85.08	83.47	77.15	67.86	77.78	72.48	69.85	68.50	94.56	79.45	66.16
Flan-T5 + T_{logic}	84.37	89.50	86.86	81.65	69.31	81.87	75.07	72.24	69.17	100.00	81.78	69.17
Llama-2	83.52	83.98	83.75	77.90	68.53	79.41	73.57	70.96	68.80	93.48	79.26	66.16
Llama-2 + T_{logic}	86.02	88.40	87.19	82.40	70.05	84.80	76.72	73.73	69.17	100.00	81.78	69.17

Table 4: The results of logical fallacy classification on three datasets. The macro precision, recall, F1 score, and accuracy are reported. The rows "+ T_{logic} " represent incorporating the logical structure tree into the model.

	ArgoTario				Reddit				Logic			
	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc
Baselines												
Jin *et al.* (2022)	–	–	–	–	–	–	–	–	55.25	63.67	58.77	47.67
Sourati *et al.* (2023b)	–	–	–	–	–	–	–	–	63.8	63.1	62.7	63.1
Sourati *et al.* (2023a)	–	–	–	–	–	–	–	–	66.3	66.4	65.7	–
Alhindi *et al.* (2022)	–	59	59	–	–	–	–	–	62	68	62	–
Sahai *et al.* (2021)	–	–	–	–	62.72	55.91	58.41	–	–	–	–	–
GPT-3.5	41.65	31.32	32.48	37.02	60.35	49.22	49.81	55.62	38.14	32.58	31.30	42.28
GPT-3.5 + T_{logic}	49.77	38.98	40.26	48.07	63.22	57.90	57.96	65.29	36.93	40.59	35.97	47.99
RoBERTa	57.97	55.98	55.92	57.46	71.99	70.37	70.42	70.76	62.50	59.66	60.03	64.88
RoBERTa + T_{logic}	59.51	58.45	58.48	59.67	75.41	74.66	74.65	74.85	67.85	63.97	64.30	67.56
Flan-T5	60.91	57.40	58.46	58.01	76.37	76.10	76.01	76.47	65.24	63.60	63.60	69.23
Flan-T5 + T_{logic}	65.23	62.12	62.95	62.78	81.98	81.34	81.25	81.29	70.90	69.14	69.37	73.49
Llama-2	60.79	58.71	59.20	59.67	77.87	77.16	77.21	77.19	65.52	63.38	63.05	69.36
Llama-2 + T_{logic}	65.63	63.29	63.92	64.09	84.84	83.68	83.95	83.63	70.70	70.03	69.55	74.16

content arguments together to form a cohesive logical structure, representing the hierarchical logical flow and thereby improving fallacy classification.

5.6 Ablation Study

The ablation study of the two designed strategies to incorporate the logical structure tree into LLMs is shown in Table 5, where we take Llama-2 model as an example. The upper rows show the results of fallacy detection on the three datasets, and the lower rows show the results of fallacy classification.

The results demonstrate that both the textualized tree and tree-based soft prompt brings improvement for fallacy detection and classification across multiple datasets. This proves that the textualized tree and tree-based soft prompt are complementary with each other: the textualized tree enriches the instruction prompt with logical structure information, and the

Table 5: The results of ablation study. The precision, recall, F1 score of fallacy class are reported for fallacy detection (upper rows). The macro precision, recall, F1 score are reported for fallacy classification (lower rows).

	Argotario				Reddit				Climate			
	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	A
Fallacy Detection												
Llama-2	83.52	83.98	83.75	77.90	68.53	79.41	73.57	70.96	68.80	93.48	79.26	66
+ textualized tree	85.25	86.19	85.71	80.52	69.54	80.12	74.46	71.94	68.70	97.83	80.72	67
+ tree-based soft prompt	85.11	88.40	86.72	81.65	69.42	83.63	75.86	72.84	68.94	98.91	81.25	68
+ both (full model)	86.02	88.40	87.19	82.40	70.05	84.80	76.72	73.73	69.17	100.00	81.78	69
	Argotario				Reddit				Logic			
	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	A
Fallacy Classification												
Llama-2	60.79	58.71	59.20	59.67	77.87	77.16	77.21	77.19	65.52	63.38	63.05	69
+ textualized tree	62.63	61.32	61.86	61.67	80.98	80.71	80.45	80.59	68.71	66.09	66.38	71
+ tree-based soft prompt	64.34	61.89	62.30	62.98	82.87	82.57	82.30	82.35	68.75	68.72	67.52	72
+ both (full model)	65.63	63.29	63.92	64.09	84.84	83.68	83.95	83.63	70.70	70.03	69.55	74

Table 8: The F1 score change across each fallacy type of fallacy classification on Logic dataset. The fallacy types include Ad Hominem, Ad Populum, False Dilemma (Black-and-White Fallacy), False Cause, Circular Reasoning, Deductive Fallacy, Appeal to Emotion (Emotional Language), Equivocation, Fallacy of Extension, Faulty Generalization (Hasty Generalization), Intentional Fallacy, Fallacy of Credibility (Irrelevant Authority), Fallacy of Relevance (Red Herring).

	Ad Hominem	Ad Populum	False Dilemma	False Cause	Circular	Deductive	Emotional
Llama-2	82.35	72.41	78.57	68.42	61.90	62.07	66.67
Llama-2 + T_{logic}	80.46	87.50	78.57	66.67	75.68	66.67	65.22
	Equivocation	Extension	Generalization	Intentional	Authority	Relevance	Macro F1
Llama-2	25.00	60.00	78.13	34.48	64.71	65.00	63.05
Llama-2 + T_{logic}	44.44	72.22	81.03	38.71	68.97	78.05	69.55

Table 6: The F1 score change across each fallacy type of fallacy classification on Argotario dataset. The fallacy types include Ad Hominem, Emotional, Generalization, Authority, Red Herring.

	Ad Hominem	Emotional	Generalization	Authority	Red Herring	Macro F1
Llama-2	60.79	67.33	55.38	63.16	49.35	59.20
Llama-2 + T_{logic}	63.16	72.16	61.29	67.80	55.17	63.92

Table 7: The F1 score change across each fallacy type of fallacy classification on Reddit dataset. The fallacy types include Slippery Slope, Irrelevant Authority, Hasty Generalization, Black-and-White Fallacy, Ad Populum, Tradition Fallacy, Naturalistic Fallacy, and Worse Problem Fallacy.

	Slippery	Authority	Generalization	Black-White	Ad Populum	Tradition	Naturalistic	Worse Problem
Llama-2	86.96	82.05	69.57	63.41	68.29	81.82	90.00	75.56
Llama-2 + T_{logic}	88.89	92.31	77.27	65.22	82.93	87.18	95.25	82.61

tree-based soft prompt enables direct learning from the tree embedding. Comparing across these two strategies, the soft prompt usually achieves better performance than the hard text prompt, and exhibits higher recall. Combining the two strategies together leads to the best performance, achieving the highest precision and recall.

5.7 Effect on Different Fallacy Types

We further analyze the F1 score change across each fallacy type in the fallacy classification task. The Llama-2 model is used as an example to show the performance change before and after incorporating the logical structure tree. Table 6 presents the F1 score change across each fallacy type on Argotario dataset. The performance change across each fallacy type on the Reddit and Logic dataset are shown in the Table 7 and Table 8. We observe that the logical structure tree brings bigger improvements for the fallacy types such as Red Herring, Hasty Generalization, Irrelevant Authority, Ad Populum, Extension Fallacy, Equivocation, Circular Reasoning etc. One possible explanation is that these fallacy types usually employ certain logical relations or logical patterns to persuade the readers. However, the performance increase is less noticeable for the fallacy types such as Appeal to Emotion and Ad Hominem. It may due to the reason that these fallacies rely more on the emotional or sentimental language instead of logical relations.

6 Limitations

We have compiled a set of connective words and phrases for the ten logical relations, as detailed in Table 1. While we have included the common connectives in this set, it may not contain all the possible connectives. The logical structure tree that is constructed based on this connective words set demonstrates its usefulness in fallacy reasoning. Future work can be expanding this connectives set and investigating the effects of various connectives.

7 Conclusion

This paper detects and classifies fallacies. We propose a logical structure tree to explicitly represent and track the hierarchical logic flow among relation connectives and their arguments. We also design two strategies to incorporate this logical structure tree into LLMs for fallacy reasoning. Extensive experiments demonstrate the effectiveness of our approach based on the logical structure tree.

Ethical Considerations

This paper aims to detect and classify logical fallacies. Logical fallacy is the error or flaws in the reasoning, and can occur in various human communications. Logical fallacies can lead to harmful consequences for society, such as spreading misinformation or introducing societal bias. The goal of this research is to understand logical fallacies, so that we can better identify and mitigate them. The release of code, datasets, and model should be used for mitigating logical fallacies, instead of expanding or disseminating the misinformation.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable feedback and input. We gratefully acknowledge support from National Science Foundation via the award

IIS2127746. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High-Performance Research Computing.

figures/fig1.pdf

Figure 1: Examples of logical fallacy sentences and their logical structure trees. The logical structure tree features logical relation connectives as non-terminal nodes, and textual arguments as terminal nodes.

figures/fig2.pdf

Figure 2: An illustration of logical fallacy classification informed by logical structure tree.