ThinkStruct: RST-Aware Attention for Logical Reasoning in Machine Reading Comprehension

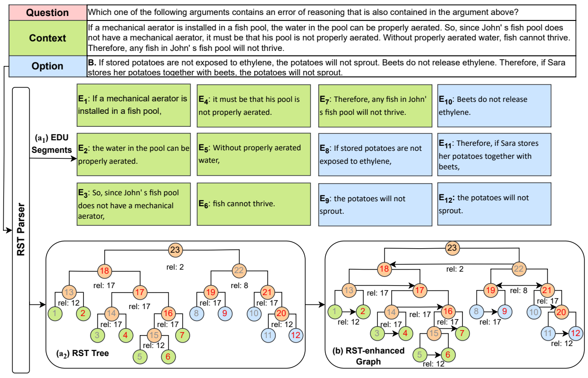
Nhi Thao Tran[1,2] and **Tien Le**[1,2] and **Toan Pham**[1,2] and **Quoc Hoang Vu**[1,2]
[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam,
[2]Vietnam National University, Ho Chi Minh City, Vietnam,
`tttnhi@fit.hcmus.edu.vn, {lpttien21,pktoan21}@clc.fitus.edu.vn,`
`vqhoang@fit.hcmus.edu.vn`
Correspondence: vqhoang@fit.hcmus.edu.vn

Abstract

Logical Reasoning is a novel approach to deal with challenging Machine Reading Comprehension tasks by utilizing the ability to construct logical structures in natural language. However, previous promising studies struggle with the accuracy of logical unit division and the consistency of model prediction on equivalent semantics. In this paper, we propose ThinkStruct, a new method that leverages a transformer network enhanced with the information of Rhetorical Structure (RS) relations for logical reasoning. Specifically, our method uses Rhetorical Structure Theory (RST) to split natural language text into Elementary Discourse Units (EDUs) and identify the relationship among these units. Node information is then fed into the fully connected transformer network, which is enhanced with logical relationships among the extracted units via adjacency matrix. Subsequently, the features of the transformer network are integrated before being passed into the answer prediction module. In addition, we employ a contrastive learning module for improving its understanding of the relationship between Elementary Discourse Units. Our experiments on the LogiQA and Reclor datasets demonstrate that our results outperform other state-of-the-art models.

1 Introduction

Machine Reading Comprehension (MRC), which facilitates machines in understanding natural language text, is a major focus (Huang et al., 2021; Li et al., 2022; Ouyang et al., 2021; Wang et al., 2021; Jiao et al., 2022) in Natural Language Processing (NLP). However, traditional MRC models have unsatisfactory performance in many datasets that require a wider and deeper range of logical methods, such as LogiQA (Liu et al., 2020) and Reclor (Yu et al., 2020) due to the lack of robust logical reasoning ability. To address this limitation, logical reasoning in MRC emerged as a new approach.


Question Which one of the following arguments contains an error of reasoning that is also contained in the argument above?
Context If a mechanical aerator is installed in a fish pool, the water in the pool can be properly aerated. So, since John' s fish pool does not have a mechanical aerator, it must be that his pool is not properly aerated. Without properly aerated water, fish cannot thrive. Therefore, any fish in John' s fish pool will not thrive.
Option **B.** If stored potatoes are not exposed to ethylene, the potatoes will not sprout. Beets do not release ethylene. Therefore, if Sara stores her potatoes together with beets, the potatoes will not sprout.

E_1: If a mechanical aerator is installed in a fish pool,
E_2: the water in the pool can be properly aerated.
E_3: So, since John' s fish pool does not have a mechanical aerator,
E_4: it must be that his pool is not properly aerated.
E_5: Without properly aerated water,
E_6: fish cannot thrive.
E_7: Therefore, any fish in John' s fish pool will not thrive.
E_8: If stored potatoes are not exposed to ethylene,
E_9: the potatoes will not sprout.
E_{10}: Beets do not release ethylene.
E_{11}: Therefore, if Sara stores her potatoes together with beets,
E_{12}: the potatoes will not sprout.

Figure 1: An example of input to the RST parser and the subsequent graph construction process for Logical Reasoning. The green boxes represent EDUs segmented from the **context**, while the blue boxes represent EDUs from the **option**. The RS tree (left) is constructed from both context and option using an RST parser, where nodes with red numbers are **nucleus** and nodes with gray numbers are **satellite**. The right figure shows the RS-enhanced graph, in which additional edges are added for better information flow.

This approach integrates comprehension ability by analyzing and evaluating the relationships among facts in natural language passage (Ouyang et al., 2023). The LogiQA example shown in Fig 1, like previous MRC models, requires the context, question, and options to generate a predicted score for the option. A unique characteristic of these texts is the presence of both implicit and explicit logical structures across semantic meaning. In Fig 1 (a_1), sentences are divided into smaller units (e.g., $E_1 - E_{12}$) by some explicit connective words or punctuation. Other information, which is presented in Fig 1 (b), includes logical relationships among Elementary Discourse Units (EDUs). These relationships are encoded numerically, for instance: 12 represents Condition relation, 17 indicates Cause-Effect relation.

While recent generative Large Language Models (LLMs) such as the GPT (Ouyang et al., 2022;

Proceedings of the 30th Conference on Computational Natural Language Learning, pages 471–480
July 3-4, 2026 ©2026 Association for Computational Linguistics

OpenAI, 2023) and LLaMA (Meta, 2024; Touvron et al., 2023) series have shown impressive capabilities, integrating complex, bidirectional discourse structures directly into their autoregressive (decoder-only) attention mechanisms remains architecturally challenging and computationally expensive. Standard token-level generation often struggles to explicitly represent global logical constraints necessary for multi-hop reasoning. Therefore, employing an encoder-only architecture (e.g., RoBERTa) serves as an optimal testbed. It naturally provides the bidirectional contextualization required to capture full-graph relationships, allowing us to explicitly merge tokens into logical units that contain independent, complete semantics and are easier to assemble into reasoning structures. Some models like DAGN (Huang et al., 2021) or AdaLoGN (Li et al., 2022) utilize a given list of words to split sentences into EDUs and use graphs to propagate the information between these EDUs. Another method is to implement learning models based on the Transformers (Vaswani et al., 2017) idea for EDUs. Furthermore, models such as LogiDRS (Pham et al., 2026) use discourse relations to split text into EDUs and enhance the relationships among all logical units.

Previous models exhibit two main limitations in their performance. The first drawback is that the EDU division phase is based on a specific set of words. This approach is not fully correct for splitting logical units due to the ambiguous semantics of those words. These errors account for a large portion of the MRC system's total error, leading to difficulty in enhancing the architecture of the prediction kernel. A second disadvantage is that their model predictions are inconsistent across equivalent semantics (Yao et al., 2024). Changes in the expression of the original text alter token-level features while the unchanged relationships among logical units are not captured.

The core motivation behind our proposed model, **ThinkStruct**, stems from the insight that human logical reasoning over text inherently relies on hierarchical discourse structures rather than just flat, sequential token interactions. While previous models attempt to build graphs from surface-level connectives, they fail to capture the global, tree-like logical dependencies of a passage. By utilizing Rhetorical Structure Theory (RST) (Thompson and Mann, 1987; Bakshi and Sharma, 2021; Liu et al., 2025; Lambropoulos and Ishihara, 2024), ThinkStruct explicitly models these global hierarchical constraints. RST provides a principled framework to segment text into logically cohesive EDUs and maps out nucleus-satellite relationships, which naturally align with logical entailment (e.g., premises often act as satellites to a core claim nucleus).

Specifically, our approach first segments EDUs using a semantic parser, which makes the formation of these logical units more rigorous than predefined word lists. By relying on deep Rhetorical Structure (RS) relations, our model ensures logical consistency in its output even when dealing with different word-level expressions that share equivalent semantic structures. Furthermore, the architecture of the model is enhanced with this hierarchical knowledge by integrating an RS-enhanced graph into the transformer attention mechanism. Finally, a targeted contrastive learning module is employed to strictly enforce the model's understanding of asymmetric logical directions between EDUs.

2 Related work

To improve logical reasoning in MRC, existing studies follow two main directions: (1) graph-based methods (Huang et al., 2021; Li et al., 2022; Ouyang et al., 2021) that explicitly model structural relationships, and (2) data augmentation approaches (Wang et al., 2021; Jiao et al., 2022) that enrich inputs. Graph-based methods like DAGN (Huang et al., 2021) use surface signals (e.g., punctuation, connectives from PDTB (Prasad et al., 2008)) to segment texts into EDUs and apply Graph Convolutional Networks (GCNs). While effective, they often create simple linear chains of EDUs that miss complex global structures, and suffer from over-smoothing during message passing. Conversely, data augmentation models like LReasoner (Wang et al., 2021) generate logical formulas for context enrichment, and MERIt (Jiao et al., 2022) leverages contrastive learning on external Wikipedia corpora. However, these methods lack an effective architecture to fully exploit explicit structural, hierarchical relationships among logical units.

3 Approach

Motivated by the limitations of graph-based and data-augmentation methods, we propose a novel approach that leverages the strength of Transformer architecture while integrating discourse structure to guide attention. Specifically, we construct an

RST graph from the input, then use rhetorical relations and nuclearity information to inform attention weights. We also incorporate contrastive learning to encourage the model to learn highly differentiated logical representations by sharpening the distinction between positive and negative samples.

3.1 Task definition

An MRC task is represented as a triplet $< \mathcal{C}_i, q_i, O_i >$, where $\mathcal{C}_i$ is the context, q_i is the question, and O_i is a set of candidate options. The goal is to select the correct answer $\hat{o}_i$ from M options $o_{i,j} \in O_i$ by maximizing the conditional probability:

$$\hat{o}_i = \arg \max_{o_{i,j} \in O_i} p(o_{i,j} | \mathcal{C}_i, q_i, O_i; \theta) \tag{1}$$

where θ represents the trainable model parameters.

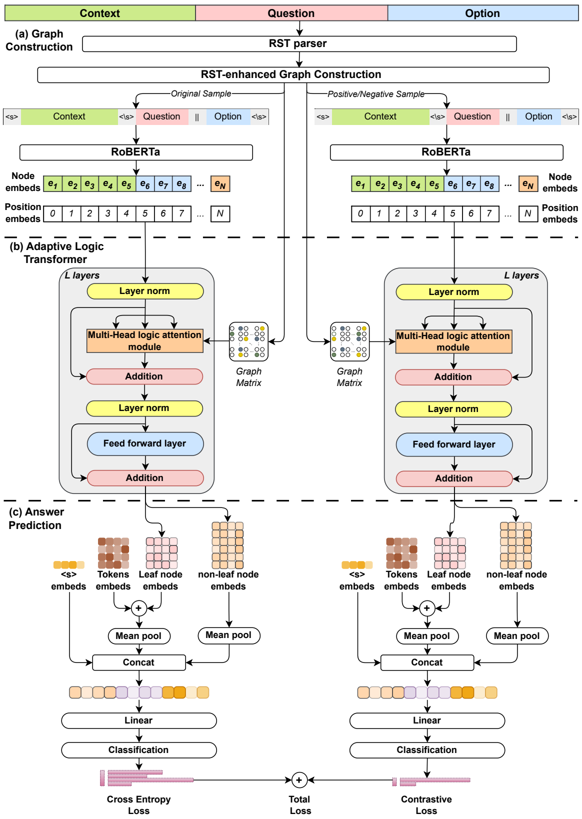


Figure 2: Overview of our ThinkStruct Model

The proposed ThinkStruct approach is illustrated in Fig 2. Our approach employs a Transformer network enhanced with RST relation information for logical reasoning.The first part of the model illustrates how the RST parser is used to segment natural language text into EDUs and construct an RST-enhanced graph based on the relations among EDUs. These enhanced graphs are split into two different branches: the original sample (left) and the augmented positive/negative samples (right), which are used to capture and learn from reversed logical relationships in the contrastive learning module. Each branch passes through the same central component to process the sample. Subsequently, the losses of both branches are combined. This central component consists of two main parts: the first part is the Transformer network enhanced with a logical graph matrix, which is used to update the node features of all EDUs. The second part illustrates the combination of all relevant features in the Answer Prediction Module to generate a predicted score for each answer.

3.2 Graph Construction

To capture and analyze the underlying logical relationships within the text, we construct an RS-enhanced graph. This graph is designed to represent both the rhetorical and logical structures inherent in the input text. The process of building this logical graph involves three distinct stages: (1) utilizing an RST parser (Chistova, 2024) to generate an RS tree directly from the input text, (2) transforming this RS tree into an RS-enhanced graph, specifically tailored to facilitate logical reasoning; and (3) generating node embeddings for each unit in the graph. As a result, we obtain a graph representation and corresponding node embeddings that serve as inputs for the reasoning module in our model.

Rhetorical Structure Theory (Thompson and Mann, 1987) (RST) Tree Construction Previous approaches like DAGN construct graphs based on surface-level textual signals, including punctuation or explicit connectives. These methods typically model the text as a sequence of adjacent Elementary Discourse Units (EDUs), where connectives serve as discourse relations linking consecutive segments. Such relations, however, are usually local, capturing only shallow connections between neighboring units, and thus fail to reflect the global discourse structure of the entire passage.

To overcome this, we utilize the IsaNLP-RST-Parser (Chistova, 2024) to parse the context $\mathcal{C}_i$ and each option $o_{i,j}$ into a hierarchical RS tree. Unlike sequential connectives, RST captures global relationships by representing text as a binary tree where leaf nodes are EDUs and non-leaf nodes define rhetorical relations. Each parent-child pair has a defined nuclearity (nucleus for primary information, satellite for supplementary), allowing the model to weigh the centrality of different logical components.

RS-Enhanced Graph Construction To bridge the gap between linguistic trees and multi-hop reasoning, distant premises must interact directly. However, the RS tree's binary nature forces sibling nodes to communicate via their parent. We address this by creating an RS-enhanced graph $G = (V, E)$. We retain original parent-child edges and add direct edges between sibling nodes: directed for nucleus-satellite relations (satellite → nucleus or nucleus ← satellite), and undirected for symmetric nucleus ↔ nucleus pairs. The graph G contains 17 relation types (Appendix 4). We encode G into a structure matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$. To avoid mathematical biases from raw categorical indices, $S[i, j]$ uses the relation's one-hot representation to look up a learnable scalar embedding (bias). This matrix $\mathbf{S}$ then guides the Adaptive Logic Transformer's attention.

To tackle this issue, we enhance the original RS tree by transforming it into an RS-enhanced graph. In this graph, first, edges are added between sibling nodes to allow direct interaction between EDUs. The direction of each edge is determined based on the nuclearity configuration of the nodes: Directed edges are used for nucleus-satellite relationships (either from satellite → nucleus or nucleus ← satellite); and Undirected edges are used when both siblings are nuclei (nucleus ↔ nucleus). Second, the original parent-child edges from the RS tree are retained to maintain the integrity of the hierarchical discourse structure.

As a result, we have a directed graph $G = (V, E)$, where V denotes the set of N nodes, including EDUs at the leaf level and their rhetorical groupings at non-leaf nodes; E represents the set of directed edges that encode rhetorical relations between nodes. In total, the graph contains 17 distinct relation types (see Appendix 4 for the full list). After that, we represent this RS-enhanced graph G using a structure matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$, where each element $S[i, j]$ encodes the rhetorical relation type from node i to node j. Specifically, the relation type is first encoded as an integer index (such as Condition or Cause-effect) and then mapped to a learnable scalar embedding (bias) via a one-hot representation. Each value of $S[i, j]$ is thus a learnable continuous parameter rather than a raw integer index. This matrix $\mathbf{S}$ is then used to guide the attention mechanism within the Transformer blocks, as described in the next section.

Node generation The model's reasoning units include both leaf nodes (EDUs) and non-leaf nodes (composed EDUs). First, we concatenate the context, question, and option $o_{i,j}$:

$$\text{input}(\mathcal{C}_i, q_i, o_{i,j}) = \text{<s>}\mathcal{C}_i\text{</s>}q_i || o_{i,j}\text{</s>} \quad (2)$$

This sequence is then fed into the RoBERTa model to generate contextualized token embeddings $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_T\}$, where T is the length of the encoded input sequence.

Based on the constructed RST tree, we align each node in the graph to its corresponding token span in the RoBERTa output. To compute the embedding $\mathbf{e}_n$ for a node n, we apply mean pooling over the embeddings of all tokens that fall within the associated EDU span U_n:

$$\mathbf{e}_n = \frac{1}{|U_n|} \sum_{\mathbf{t} \in U_n} \mathbf{t} \quad (3)$$

To preserve the relative order of nodes in the input sequence, we incorporate positional embeddings into the node representations:

$$\mathbf{e}_n = \mathbf{e}_n + \text{PosEmbed}(\mathbf{e}_n) \quad (4)$$

The node embeddings $\mathbf{e}_n$, after being enriched with positional information, are assembled into a representation matrix $\mathbf{E}_i = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N\}$, $\mathbf{E}_i \in \mathbb{R}^{N \times d}$, where N denotes the number of nodes in the RST-enhanced graph of the i-th input sample, and d is the dimensionality of each embedding vector. The nodes are ordered such that the leaf nodes come first, followed by the non-leaf nodes, arranged in a bottom-up manner according to the tree structure. This matrix serves as the input for the structure-aware reasoning component described in the following sections.

3.3 Adaptive Logic Transformer

After obtaining the node embeddings $\mathbf{E}_i = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N\}$ from the RS-enhanced graph, we perform structure-aware reasoning using a Transformer block adapted to incorporate the topological structure of the graph.

First, the matrix $\mathbf{E}_i$ is projected into three subspaces to produce the query, key, and value matrices:

$$\mathbf{Q} = \mathbf{E}_i \cdot \mathbf{W}^Q \mathbf{K} = \mathbf{E}_i \cdot \mathbf{W}^K \mathbf{V} = \mathbf{E}_i \cdot \mathbf{W}^V \quad (5)$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d\times d}$ are learnable weight matrices. Then, to compute attention, we rely not only on the similarity between queries and keys but also on the structure of the RS-enhanced graph. Specifically, the adjacency matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ restricts attention flow to pairs of nodes that share a rhetorical or logical connection:

$$\mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}} + \mathbf{S} \tag{6}$$

$$\mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\mathbf{A}) \cdot \mathbf{V} \tag{7}$$

Following the standard Transformer architecture (Vaswani et al., 2017), we apply multi-head attention $\mathrm{MHA}(\cdot)$ with H heads to capture diverse interaction patterns among nodes:

$$\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_H) \cdot \mathbf{W}^O \tag{8}$$

where $\mathbf{W}^O \in \mathbb{R}^{(H*d)\times d}$ is a linear projection matrix; $\mathrm{head}_i = \mathrm{Att}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V})$. Each attention layer is followed by two essential components: a residual connection with layer normalization $\mathrm{LN}(\cdot)$, and a position-wise feed-forward network $\mathrm{FFN}(\cdot)$.

$$\mathbf{H}' = \mathrm{LN}(\mathbf{E}_i^{(l)} + \mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) \tag{9}$$

$$\mathbf{E}_i^{(l+1)} = \mathrm{LN}(\mathbf{H}' + \mathrm{FFN}(\mathbf{H}')) \tag{10}$$

This process is repeated across L layers. Finally, we select the node representation from the last K layers and aggregate them using sum-pooling to obtain the final representation for all nodes in the graph:

$$\mathbf{E}_i^{final} = \sum_{l=L-K+1}^{L} \mathbf{E}_i^{(l)} \tag{11}$$

where $\mathbf{E}_i^{final} \in \mathbb{R}^{N\times d}$ is the final state of the node features, $\mathbf{E}_i^l \in \mathbb{R}^{N\times d}$ denotes the hidden state at layer l^{th} of the node representations. These final embeddings $\mathbf{E}_i^{final}$ are subsequently used as input to the answer prediction module.

3.4 Answer Prediction Head

At this point, all nodes have been enhanced with logical relationships. To fully utilize this information, we first broadcast the enhanced EDU leaf nodes, U_n, to all tokens they contain $\mathbf{E}_{leaf}$, and integrate these with the token sequence generated from RoBERTa-Large, $\mathbf{E}_{seq}$. These two features are assumed to contribute equally to the answer prediction phase. The final token-level feature $\mathbf{E}_{final}$ is then calculated as:

$$\mathbf{E}_{final} = \mathrm{LayerNorm}(\mathbf{E}_{leaf}^{final} + \mathbf{E}_{seq}) \tag{12}$$

For the non-leaf nodes, we also apply mean pooling layer to extract all globalized information from these nodes $\mathbf{E}_{non\_leaf}$:

$$\mathbf{E}_{non\_leaf} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{E}_i^{final} \tag{13}$$

where k is the number of non-leaf nodes.

We also utilize the special token embedding $\mathbf{E}_{<CLS>}$ to leverage global contextual information. Finally, these representations are concatenated and passed through a linear layer to generate prediction scores for each option:

$$\mathrm{score}_{o_{i,j}} = \mathrm{Linear}(\mathbf{E}_{\text{<CLS>}} || \mathbf{E}_{final} || \mathbf{E}_{non\_leaf}) \tag{14}$$

Here, $\mathrm{score}_{o_{i,j}}$ represents the predicted score for each option. In training phase, we employ Cross-Entropy Loss function to update the model's parameters.

3.5 Contrastive Learning

The objective of contrastive learning is to pull the representations of similar samples closer together while the distance between the representations of dissimilar samples is maximized, as follows:

$$s(f(x), f(x^+)) \gg s(f(x), f(x^-))) \tag{15}$$

where x is the original samples, x^+ and x^- denote the positive and negative samples, $f(.)$ is the decoder function, and $s(.)$ is the cosine similarity function.

Logical reasoning is highly sensitive to relation directionality. Arbitrarily reversing a "Cause-Effect" relation fundamentally alters the premise, yielding a strong hard-negative. Thus, our contrastive learning generates positive samples by reversing two EDUs connected by an undirected edge (preserving semantic symmetry) and negative samples by swapping EDUs with directed edges (violating asymmetric logic). We prioritize swapping nuclear nodes for positive samples and nucleus-satellite pairs for negative ones. Valid contrastive triplets are encoded to compute the contrastive loss $\mathcal{L}_C$:

$$\mathcal{L}_C = -\sum \log \frac{\exp(s(+))}{\exp(s(+)) + \exp(s(-))} \tag{16}$$

where $s(+)$ is a shorthand for $s(f(x), f(x^+))$ and $s(-)$ is a shorthand for $s(f(x), f(x^-))$. Finally, the total loss of our model is the sum of the Answer Prediction Head's loss and the contrastive loss $\mathcal{L}_C$.

4 Experiments

4.1 Datasets

We conduct experiments on two challenging datasets widely used for logical reasoning in MRC: ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2020). ReClor contains 6,138 standardized test questions (e.g., GMAT, LSAT), with its test set evaluated blindly via an official leaderboard (divided into EASY and HARD splits based on a BERT-base baseline). LogiQA consists of 8,678 questions translated from the Chinese Civil Service Examination. While LogiQA offers a larger volume, ReClor covers a broader spectrum of reasoning types (Table 1).

Table 1: Breakdown of the ReClor and LogiQA datasets, including their training, validation, and test set sizes, as well as the number of reasoning types.

Dataset	#Train	#Dev	#Test	#Reason Type
ReClor	4,638	500	1,000	17
LogiQA	7,376	651	651	5

4.2 Baselines

To demonstrate the effectiveness of our model ThinkStruct, we compare it against the following baseline models:

- **Random & Human (Liu et al., 2020; Yu et al., 2020):** Random guesses and average student performance.
- **RoBERTa-Large (Liu et al., 2019):** A strong text encoder used as our primary backbone.
- **DAGN (Huang et al., 2021) & AdaLoGN (Li et al., 2022):** Graph-based models leveraging EDU segmentation via predefined keywords.
- **MERIt (Jiao et al., 2022):** A data-augmentation pre-training framework.
- **LogiDRS (Pham et al., 2026):** A model integrating PDTB 2.0 discourse relations into Transformers.
- **Generative LLMs (Wang et al., 2025):** Recent decoder-only models (GPT-3.5, GPT-4, LLaMA-2/3, Mistral).

4.3 Implementation Details

Experiments were conducted on a single A100 GPU using RoBERTa-Large as the encoder. We trained for 20 epochs with a batch size of 4 using Adam (Kingma and Ba, 2017) and a peak learning rate of $5e-6$ (see Table 2).

Table 2: The tuned hyper-parameters with search scopes.

Name of Parameter	Search Scope	Best
training batchsize	{2,4,6,8,10}	4
#epoch	{8,10,12,14,16,18,20}	20
#head in transformer	{1,2,3,4,5}	5
#layer in transformer	{1,2,3,4,5}	5
max sequence length	{128,256,512}	512
learning rate	{4e-6, 5e-6, 6e-6, 7e-5}	5e-6

4.4 Results

Table 3 displays the primary results. ThinkStruct outperforms most baselines, including its direct competitor, LogiDRS, on LogiQA and ReClor Easy. This validates that our RS tree-based global hierarchical relations provide stronger structural reasoning signals than the linear, locally related EDU sequences used in LogiDRS.

When compared to generative LLMs, ThinkStruct (built upon a 355M-parameter encoder with lightweight reasoning layers) exhibits impressive efficiency. It surpasses the performance of GPT-3.5 and LLaMA2-7B on ReClor, despite having significantly fewer parameters. While massive models like GPT-4, LLaMA3-8B, and Mistral-7B achieve higher overall metrics due to their scale, our method demonstrates that explicitly injecting linguistic discourse structure into a smaller encoder model can dramatically close the performance gap without the immense computational cost required by large decoder-only LLMs. Furthermore, explicitly modeling inter-EDU relationships via RS graphs significantly improves upon the base RoBERTa-Large by margins of 4.4% (ReClor test) and 8.76% (LogiQA test).

We next compare ThinkStruct to structure-aware models that use EDU division, specifically DAGN and AdaLoGN. ThinkStruct maintains competitive performance on both benchmarks. On the more challenging LogiQA dataset, the results (ThinkStruct at 44.09%, AdaLoGN at 40.71%, and

Table 3: Performance comparison of models on ReClor and LogiQA datasets. Accuracy is the evaluation metric. The ReClor test set is further divided into two subcategories, Easy and Hard (440 and 560 data points, respectively).

Model	**ReClor**				**LogiQA**	
	Dev	Test	Test-E	Test-H	Dev	Test
Discriminative (Encoder-only) Models						
Random	25.00	25.00	25.00	25.00	25.00	25.00
Human Performance	-	63.00	57.10	67.20	-	86.00
BERT-Large	53.80	49.80	72.00	32.30	34.10	31.03
RoBERTa-Large	62.60	55.60	75.50	40.00	35.02	35.33
DAGN	65.20	58.20	76.14	44.11	35.48	38.71
AdaLoGN	65.20	60.20	79.32	45.18	39.94	40.71
MERIt	66.80	59.60	78.10	45.20	40.00	38.90
LogiDRS	67.40	60.80	78.40	46.96	38.40	40.09
Ours						
ThinkStruct	61.92	60.00	79.09	45.00	39.63	**44.09**
w/o contrastive learning	**69.18**	**61.80**	**79.55**	**47.85**	**40.70**	39.48
Generative LLMs						
LLaMA2-7B	58.00	58.63	66.14	52.74	49.71	49.75
LLaMA3-8B	67.60	70.97	77.27	66.01	60.29	58.78
Mistral-7B	69.73	71.17	78.79	65.18	61.22	60.28
GPT-3.5	56.00	58.20	61.82	55.36	55.07	51.15
GPT-4	87.20	89.30	90.45	88.39	76.32	74.81

DAGN at 38.71%) demonstrate that modeling inter-EDU relationships is key to success in the reasoning phase. Crucially, ThinkStruct outperforms AdaLoGN by 3.38 percentage points, suggesting that our RS tree-based approach provides superior structural information for deep reasoning compared to AdaLoGN's logic graph formation. In the ReClor dataset, although ThinkStruct's dev set result is lower (61.92% vs 65.20% and 65.20%), the test set shows the opposite: ThinkStruct achieves results comparable to AdaLoGN and marginally surpasses DAGN (around 60.00% vs 58.20%). This smaller performance divergence is expected for ReClor because it requires less complex inferences and logical structure analysis, yet its high lexical diversity still presents a challenge.

ThinkStruct and other discourse information integration models, such as LogiDRS, achieve superior results compared to data augmentation methods like MERIt. This is starkly evident on LogiQA, where ThinkStruct (44.09%) and LogiDRS (40.09%) significantly surpass MERIt (38.90%). A similar pattern holds for ReClor, where MERIt (59.60%) trails both LogiDRS (60.80%) and ThinkStruct (60.00%). These outcomes demonstrate the necessity of leveraging intrinsic discourse knowledge for effective logical reasoning. While MERIt generates additional training data from Wikipedia, it lacks an architectural mechanism to exploit the structural relationships within that knowledge effectively. LogiDRS and ThinkStruct, conversely, achieve higher performance by focusing on internal discourse structure without external data augmentation. Notably, ThinkStruct maintains a lead over LogiDRS on LogiQA and the ReClor Easy set, even though it is marginally behind on the full ReClor test set. This result shows that ThinkStruct's RS tree-based representation, reflecting hierarchical and global discourse relations, offers a distinct advantage over the linear, locally related EDU sequence utilized by LogiDRS.

An ablation study was performed to isolate the effect of the contrastive learning module in ThinkStruct. We find a clear trade-off between the two datasets. On LogiQA, the contrastive learning module significantly benefits the model, increasing test performance from 39.48% to 44.09%. This gain occurs because the contrastive learning module both emphasizes existing implicit relations and allows ThinkStruct to infer new implicit relations unobserved during training. This outcome strongly supports the conclusion that logical reasoning tasks in LogiQA necessitate a multi-hop reasoning phase reliant on diverse inference paths.

Conversely, on ReClor, the contrastive learning module proves detrimental. Scores drop from 69.18% (dev) / 61.80% (test) to 61.92% (dev) / 60.00% (test) when the module is included. To explain this dataset-level discrepancy, we analyzed the structural differences between LogiQA and ReClor. First, LogiQA passages typically contain more complex logical structures, yielding significantly more EDUs per sample compared to ReClor. Consequently, LogiQA graphs contain a higher ratio of asymmetric directed edges (e.g., Cause-Effect, Condition), which are the primary targets for generating negative samples. This abundance ensures a high success rate in generating hard-negative contrastive pairs, forcing the model to learn rigorous logical directionality. In contrast, ReClor's texts often result in sparse RST trees with fewer EDUs. This sparsity leads to a higher frequency of failing to generate valid contrastive pairs (where the model must fall back to the original sample), causing the contrastive module to be less effective. Furthermore, since most ReClor samples can be solved using simpler surface cues, forcing the extraction and contrast of implicit relationships introduces structural noise, diverting the model's focus from core semantic evidence.

5 Conclusion

In this study, we propose ThinkStruct — the first approach that exploits Rhetorical Structure Theory for logical reasoning problems in NLP. Instead of relying solely on surface cues like previous graph methods, ThinkStruct uses RST trees to analyze the global discourse structure and construct an RST-enhanced graph, which helps the model better capture the logical relationships between EDUs. In addition, the model also integrates contrastive learning to better distinguish between logical choices. Experimental results on two benchmark sets, ReClor and LogiQA, show that ThinkStruct outperforms existing baseline methods.

In the future, we will study other architectures that can better utilize the RST structure, since RST provides a global view of the text and has shown remarkable effectiveness in logical reasoning problems. Additionally, the contrastive learning mechanism will be optimized with more appropriate data generation and training strategies for each question type, in order to continue to effectively exploit the role of rhetorical relations in reading comprehension systems.

Acknowledgments

This research is funded by University of Science, VNU-HCM under grant number CNTT 2024-17. This research used the GPUs provided by the Intelligent Systems Lab at the Faculty of Information Technology, University of Science, VNU-HCM. Additionally, the datasets PDTB 2.0 used in this study were generously supported by the Linguistic Data Consortium (LDC).

References

Sahil Bakshi and Dipti Misra Sharma. 2021. A transformer based approach towards identification of discourse unit segments and connectives. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 13–21.

Elena Chistova. 2024. Bilingual rhetorical structure parsing with large parallel annotations. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9689–9706, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. DAGN: discourse-aware graph network for logical reasoning. *CoRR*, abs/2103.14349.

Fangkai Jiao, Yangyang Guo, Xuemeng Song, and Liqiang Nie. 2022. MERIt: Meta-Path Guided Contrastive Learning for Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3496–3509, Dublin, Ireland. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Michael Lambropoulos and Shunichi Ishihara. 2024. Towards an implementation of rhetorical structure theory in discourse coherence modelling. In *Proceedings of the 22nd Annual Workshop of the Australasian Language Technology Association*, pages 1–11.

Xiao Li, Gong Cheng, Ziheng Chen, Yawei Sun, and Yuzhong Qu. 2022. Adalogn: Adaptive logic graph network for reasoning-based machine reading comprehension. *Preprint*, arXiv:2203.08992.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *CoRR*, abs/2007.08124.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zhengyuan Liu, Ke Shi, and Nancy F Chen. 2025. Transformer-based document-level discourse processing: Exploiting prior language knowledge and hierarchical parsing. *Computer Speech & Language*, page 101809.

AI @ Meta. 2024. The llama 3 herd of models. *CoRR*, abs/2404.19737.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Fact-driven logical reasoning. *CoRR*, abs/2105.10334.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2023. Fact-driven logical reasoning for machine reading comprehension. *Preprint*, arXiv:2105.10334.

Toan Pham, Tien Le, Hoang Quoc Vu, and Nhi-Thao Tran. 2026. Logidrs: Logical reasoning using discourse relation sense. In *Advances in Computational Collective Intelligence*, pages 364–379, Cham. Springer Nature Switzerland.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC‘08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Sandra A Thompson and William C Mann. 1987. Rhetorical structure theory: A theory of text organization. *ISI/RS Report*, 87(190):2–82.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Chenxu Wang, Ping Jian, and Zhen Yang. 2025. Thought-path contrastive learning via premise-oriented data augmentation for logical reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25345–25352.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. 2021. Logic-driven context extension and data augmentation for logical reasoning of text. *CoRR*, abs/2105.03659.

Bingsheng Yao and 1 others. 2024. Discursive socratic questioning: Evaluating the faithfulness of language models' understanding of discourse relations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. *CoRR*, abs/2002.04326.

A List of Rhetorical Relation Types

Table 4: List of rhetorical relation types used in the RS-enhanced graph.

Index	**Relation Type**
1	Preparation
2	Elaboration
3	Sequence
4	Purpose
5	Attribution
6	Solutionhood
7	Concession
8	Joint
9	Contrast
10	Restatement
11	Interpretation-evaluation
12	Condition
13	Same-unit
14	Comparison
15	Evidence
16	Background
17	Cause-effect

B Response to Reviewers

We sincerely thank the Area Chair and all reviewers for their valuable and constructive feedback. Below is a brief summary of how we addressed the major concerns.

Generative LLM Baselines & Architecture Choice (Reviewer 1, Reviewer 2, Reviewer 3)

We expanded our evaluation to include GPT-3.5, GPT-4, LLaMA-2/3, and Mistral (Table 3). The results show ThinkStruct remains highly competitive despite having significantly fewer parameters.

We also clarified in the Introduction why an encoder-only architecture (RoBERTa) is used: it naturally provides the bidirectional contextualization required to capture full-graph hierarchical relationships without the massive computational overhead of autoregressive LLMs.

Implicit Discourse Trees in Decoder-Only Models (Reviewer 3)

Regarding the insightful question of whether LLMs internally build RST trees: while large models capture some discourse dependencies implicitly, their sequential token-level generation often fails to enforce strict, explicit hierarchical logical constraints, leading to inconsistencies in complex multi-hop reasoning. Explicitly modeling these graphs remains beneficial.

Clarification on Motivation and Insight Analysis (Reviewer 1, Reviewer 4)

We thoroughly revised the Introduction and Approach to explicitly justify our design choices. At a macro level, human logical reasoning is intrinsically hierarchical, making RST an ideal framework. At a micro level, we clarified why we enhanced the RST tree (to enable direct sibling information flow) and designed our contrastive learning based on relation directionality (to explicitly penalize violations of asymmetric logic like Cause-Effect).

Additional Experiments, Ablations & Dataset Choice (Reviewer 1, 2, 4)

We appreciate the suggestions for additional experiments and granular ablations. Due to time constraints during the revision, we cannot add new experimental setups. However, our main results (Table 3) inherently serve as a comparative ablation: ThinkStruct's significant margin over DAGN and AdaLoGN—which share our RoBERTa backbone but rely on keyword-based EDU splits—empirically validates our RST-based segmentation. Regarding dataset selection, we evaluated on LogiQA and ReClor (clarified in Section 3.1) as they are the most rigorous, widely accepted benchmarks specifically designed to test complex, multi-hop logical reasoning in MRC.

Dataset-level Analysis on Contrastive Learning (Reviewer 2)

As suggested, we added a dataset-level analysis in Section 3.4. We explain that LogiQA's denser graphs (more asymmetric edges) provide rich contrastive signals, whereas ReClor's sparse graphs struggle to yield valid contrastive pairs, introducing noise rather than beneficial regularization.

Clarification on Equation 6 and Attention Bias (Reviewer 1, 2)

We apologize for the misleading description of Equation 6 in the original manuscript. As rightly pointed out, directly adding categorical integer indices (1-17) as mathematical biases is flawed. In our actual implementation, the categorical relation index is mapped to a one-hot representation and used to look up a learnable scalar embedding (bias) for each edge type. We have corrected the text in the Approach section to explicitly clarify this mechanism.

Formatting and Citation Adjustments (Reviewer 3, Reviewer 4)

We have fully addressed all formatting concerns: figure captions have been moved below the images, tables and equations have been resized to fit within margins and unified in style, and typos have been corrected. Furthermore, we consistently adopted the abbreviation "RST" for Rhetorical Structure Theory and "RS" for Rhetorical Structure, and we appropriately cited the DiSQ paper to strengthen our discussion on logical consistency.