Logiformer: A Two-Branch Graph Transformer Network for Interpretable Logical Reasoning

Fangzhi Xu
School of Computer Science and Technology, Xi'an Jiaotong University
Xi'an, China
Leo981106@stu.xjtu.edu.cn

Jun Liu*
Shaanxi Province Key Laboratory of Satellite and Terrestrial Network Tech. R&D, National Engineering lab for Big Data Analytics
Xi'an, China
liukeen@xjtu.edu.cn

Qika Lin
School of Computer Science and Technology, Xi'an Jiaotong University
Xi'an, China
qikalin@foxmail.com

Yudai Pan
School of Computer Science and Technology, Xi'an Jiaotong University
Xi'an, China
pyd418@foxmail.com

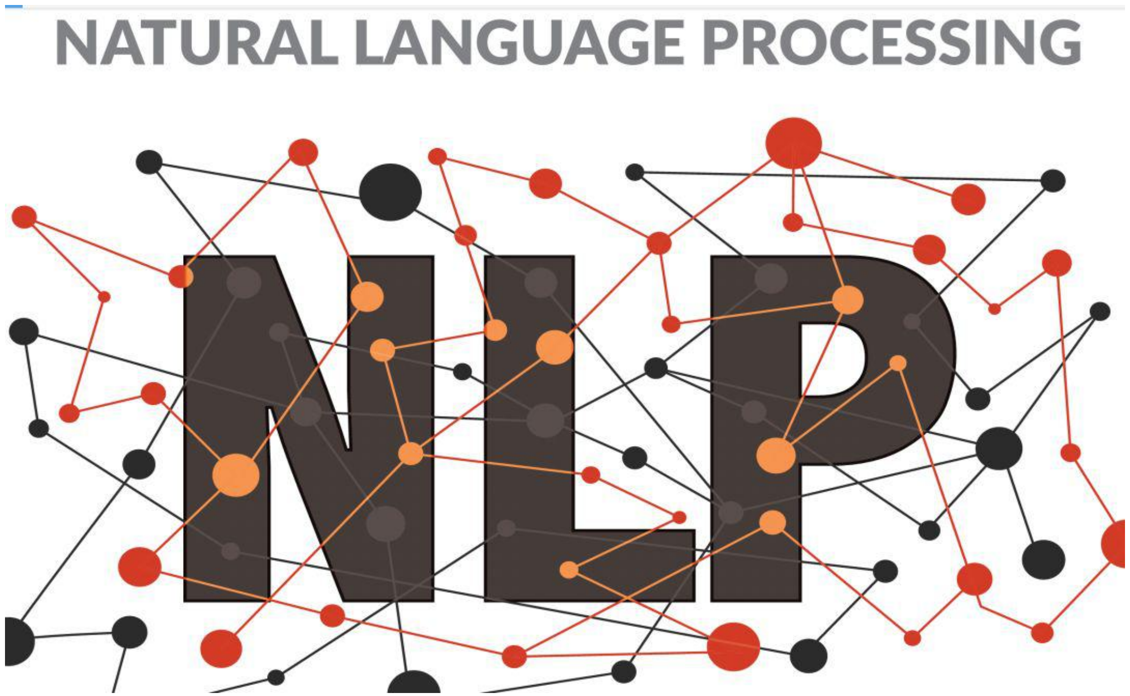
Lingling Zhang
School of Computer Science and Technology, Xi'an Jiaotong University
Xi'an, China
zhanglling@xjtu.edu.cn

2022.09.01 • ChongQing

2022_SIGIR

Reported by Yidan Liu

Leibniz-Institut für Sozialwissenschaften



1. Introduction
2. Method
3. Experiments

Introduction

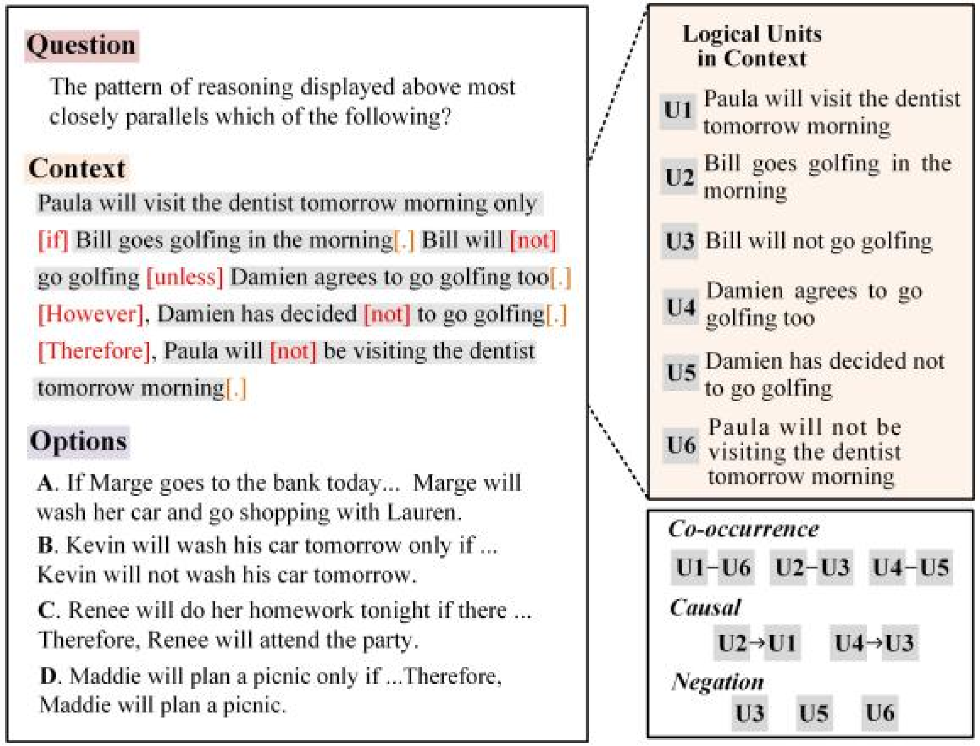


Figure 1: An example of the logical reasoning task and some detailed illustrations.

- There still remains a challenge to model the long distance dependency among the logical units.
- It is demanding to uncover the logical structures of the text and further fuse the discrete logic to the continuous text embedding.

Method

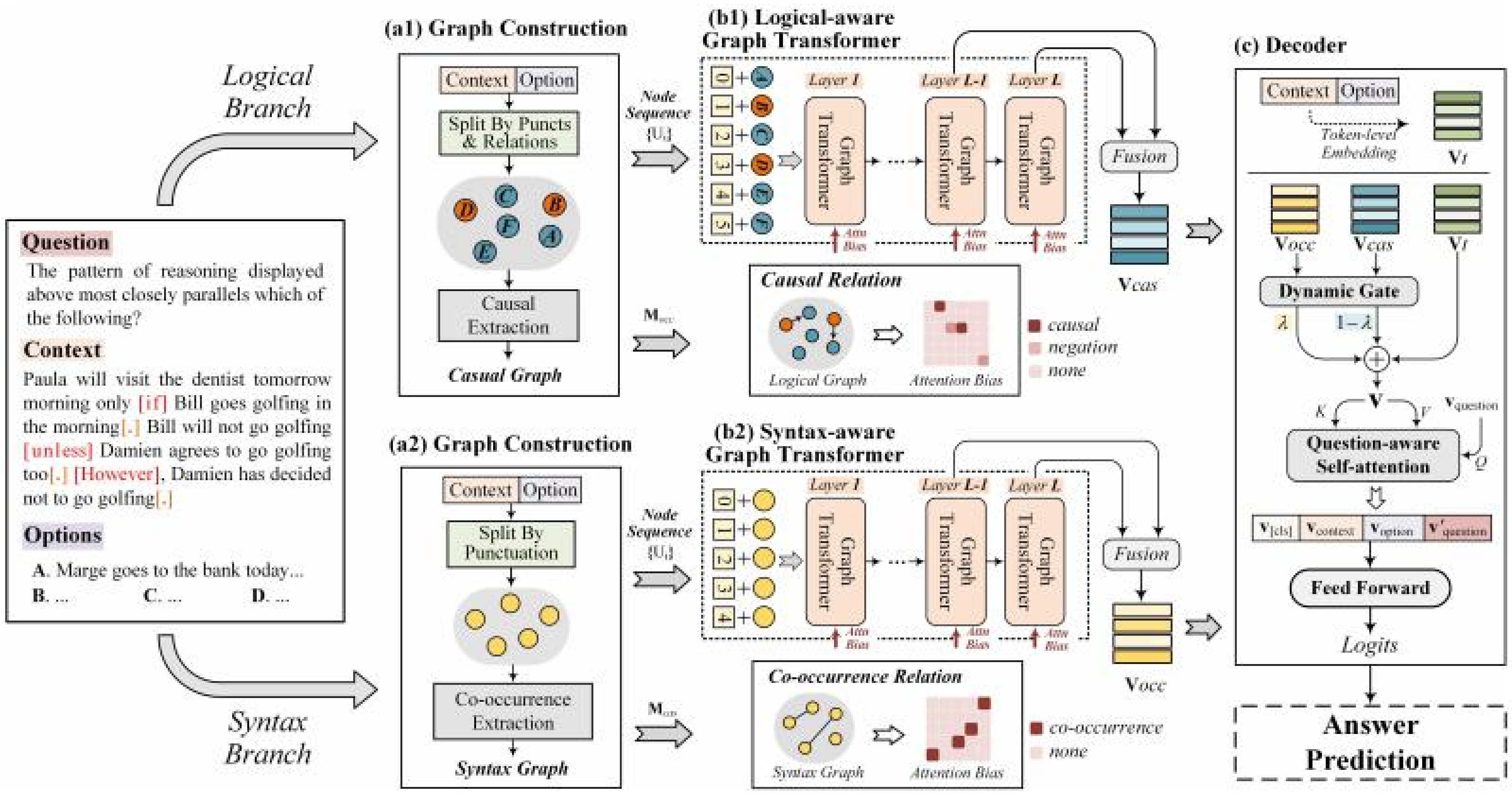
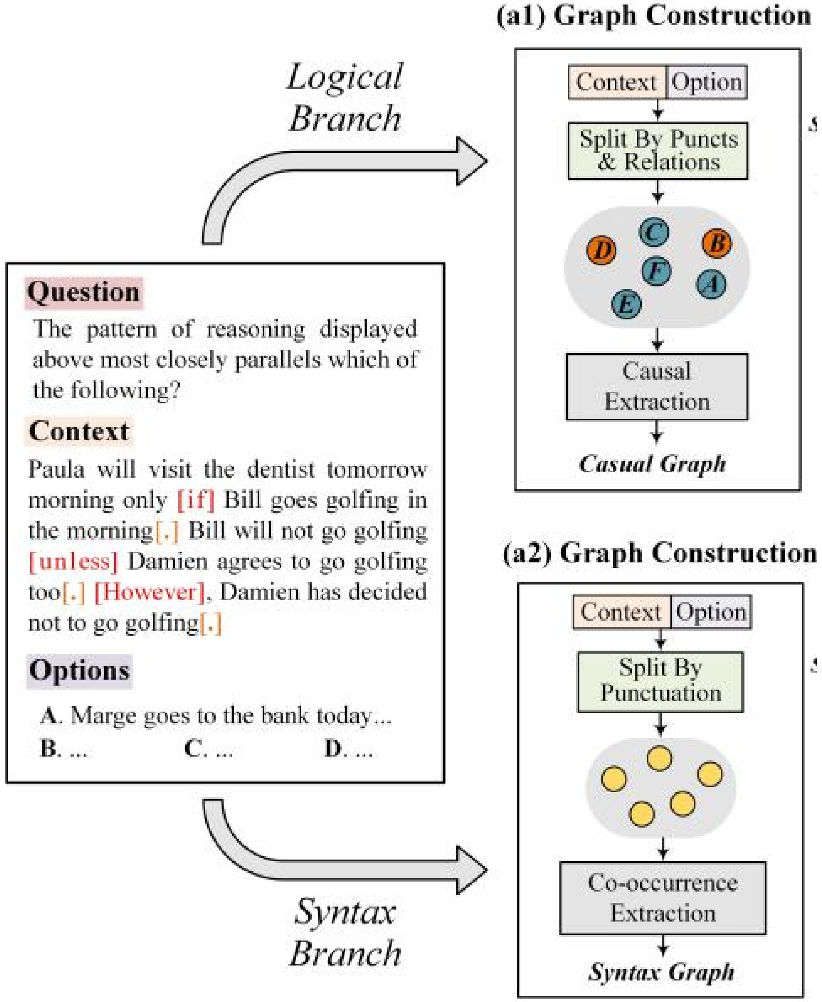


Figure 2: The architecture of Logiformer. The left part is an input example of the dataset. The graph construction modules ($a1,a2$) split the text into logical units and build two graphs from two branches respectively. The graph transformer structures ($b1,b2$) update the text features combined with the logical and syntactic relations. Finally, the decoder module (c) is utilized to conduct the feature fusion and predict the answers.

Method



Logical Graph

According to the extracted causal node pairs, we can create directed connection from each condition node p to result node q.

This kind of connection is reflected in the adjacent matrix $\mathrm{M}_{cas} \in \mathbb{R}^{K_{cas} \times K_{cas}}$ of the logical graph as $\mathrm{M}_{cas}[p-1, q-1] = 1$.

Also, to avoid the semantic reverse brought by the negation, we mark the nodes with the explicit negation words (e.g., *not*, *no*). The node k with negation semantics are expressed in the adjacent matrix as $\mathrm{M}_{cas}[k-1, k-1] = -1$.

Syntax Graph

$$\mathrm{M}_{occ} \in \mathbb{R}^{K_{occ} \times K_{occ}}$$

Method

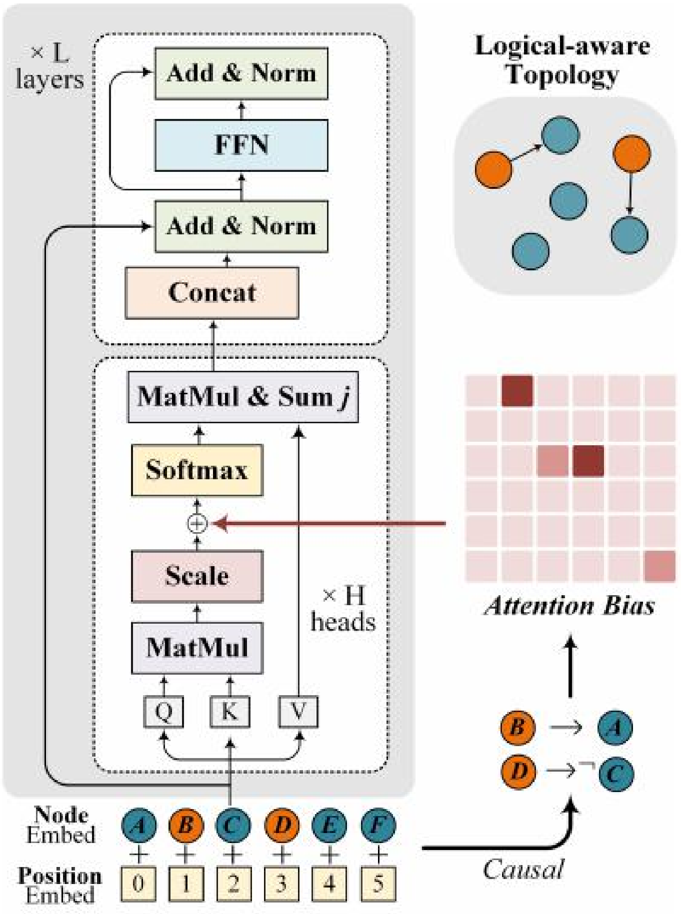


Figure 3: The illustration of logical-aware graph transformer. The inputs are the node sequence as well as the topology and the outputs are omitted.

Graph Transformer

$$\text{Input}(c_i, a_{i,j}) = [CLS]c_i[SEP]a_{i,j}[SEP], \quad (3)$$

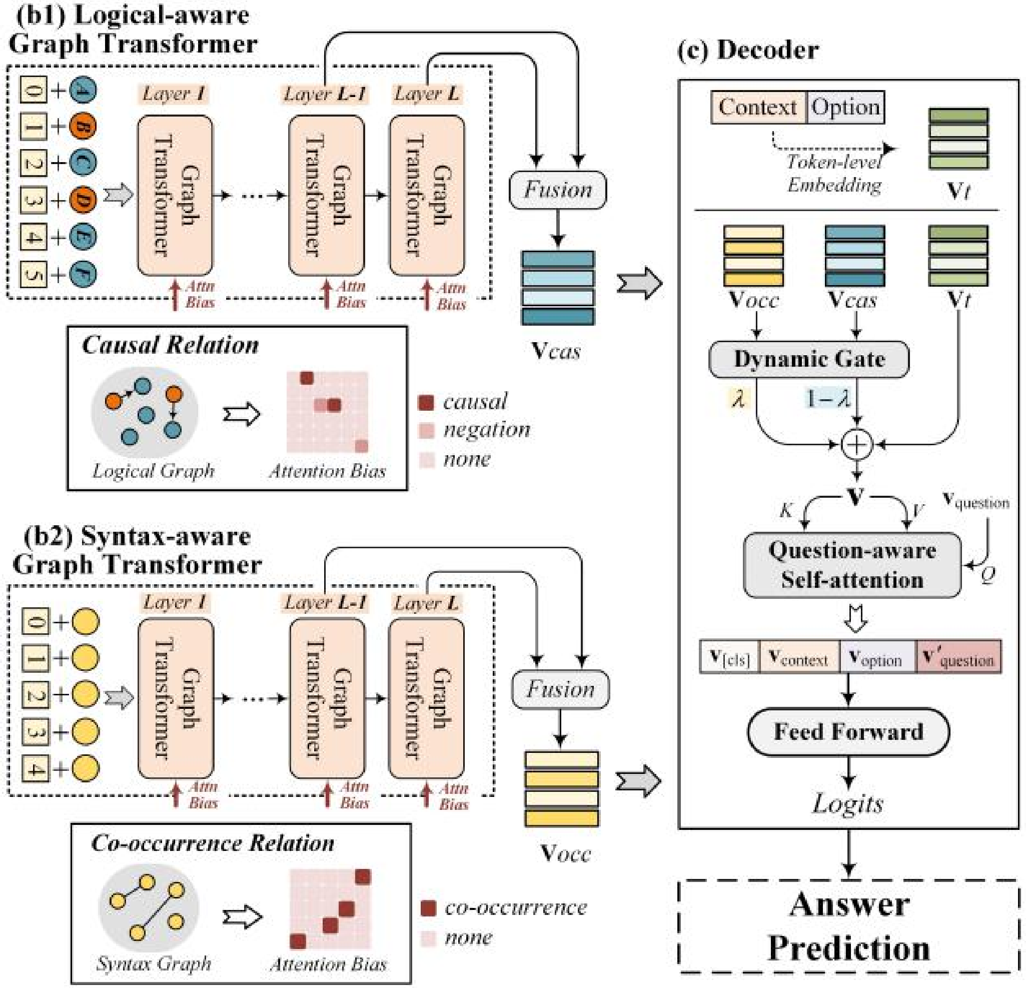
we employ the RoBERTa model [19] as the encoder for the token-level features. For the token sequence $\{t_1^{(k)}, t_2^{(k)}, ..., t_T^{(k)}\}$ with the length T of each node U_k, the obtained token embedding is represented as $\{v_{t_1}^{(k)}, v_{t_2}^{(k)}, ..., v_{t_T}^{(k)}\}$. We take the average embedding of T tokens as the original feature for node U_k:

$$v_k = \frac{1}{M}\sum_{i=1}^{M} v_i^{(k)}. \quad (4)$$

$$V_i = V_o + PosEmbed(V_o), \quad (5)$$

where $V_o = [v_1; v_2; ...; v_{K_{cas}}]$, $V_o \in \mathbb{R}^{K_{cas} \times d}$, d is the dimension of the hidden state, and K_{cas} is the number of nodes. $PosEmbed(\cdot)$ provides a d-dimensional embedding for each node in the input sequence.

Method



$$Q = \mathrm{V_i} \cdot \mathrm{W^Q},$$
$$K = \mathrm{V_i} \cdot \mathrm{W^K}, \quad (6)$$
$$V = \mathrm{V_i} \cdot \mathrm{W^V},$$

$$A = \frac{QK^{\mathrm{T}}}{\sqrt{d_k}}, \quad (7)$$
$$Att(Q, K, V) = \mathrm{softmax}(A) \cdot V,$$

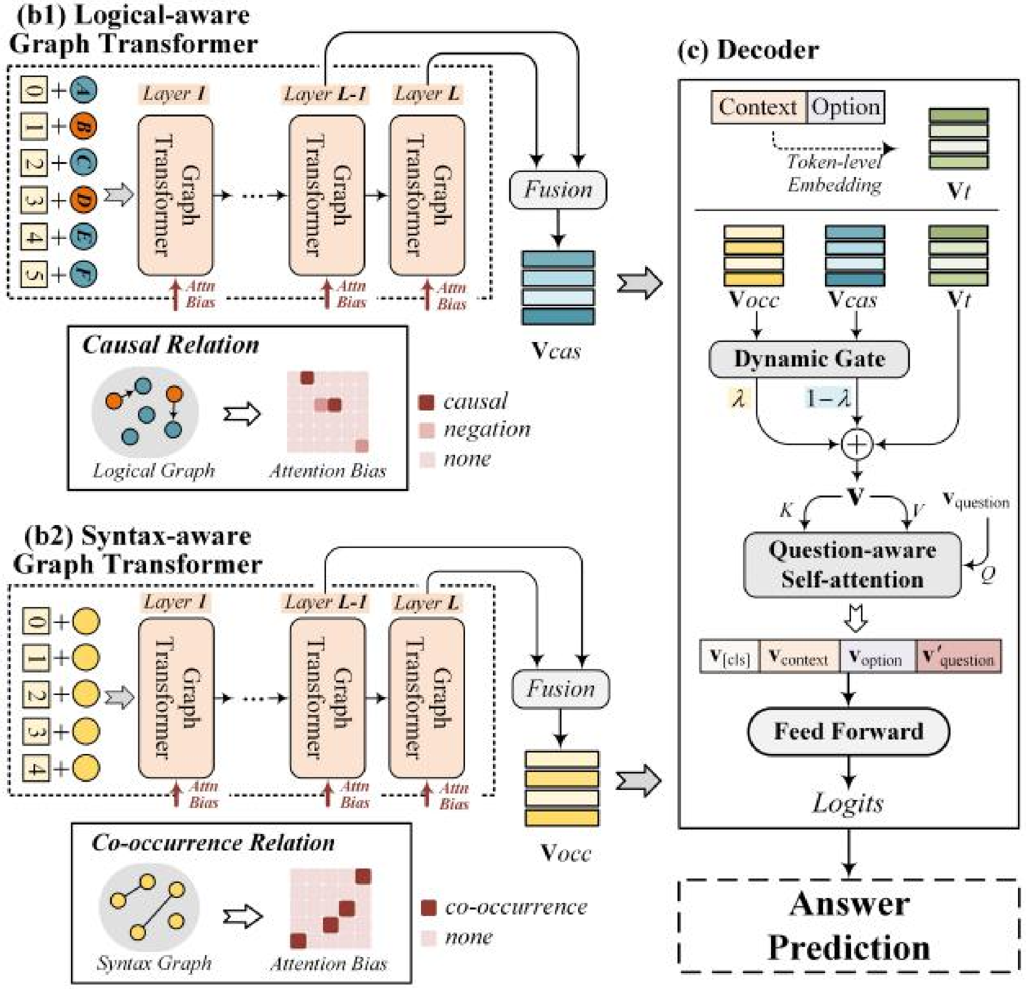
$$A' = \frac{QK^{\mathrm{T}}}{\sqrt{d_k}} + \mathrm{M}_{cas}. \quad (8)$$

$$Att_{MH}(Q, K, V) = [Head_1; ...; Head_H] \cdot \mathrm{W^H}, \quad (9)$$

$$\mathrm{V}_{cas} = \mathrm{V}_{cas}^{(L-1)} + \mathrm{V}_{cas}^{(L)}, \quad (10)$$

$$\mathrm{V}_{occ} \in \mathbb{R}^{K_{occ} \times d}.$$

Method



Decoder

$$V_t, V'_{occ}, V'_{cas} \in \mathbb{R}^{N\times d}.$$

$$\lambda = \mathrm{softmax}([V'_{occ}; V'_{cas}]W_g + b_g), \quad (11)$$

$$V = LN(V_t + \lambda \cdot V'_{occ} + (1-\lambda)\cdot V'_{cas}), \quad (12)$$

$$V_{cls} = LN(V_{t,cls} + \frac{1}{N-1}\sum_{i=1}^{N-1}(V'_{occ,i} + V'_{cas,i})), \quad (13)$$

$$V'_{question} = \mathrm{softmax}(\frac{V_{question}V^T}{\sqrt{d}})\cdot V. \quad (14)$$

$$V_{final} = [V_{cls}; V_{context}; V_{option}; V'_{question}]. \quad (15)$$

For each option in one example, we can get one specific final feature. They are fed into the feed forward network to obtain the scores, and we take the highest one as the predicted answer.

Experiment

Table 2: Detailed Splits of ReClor and LogiQA.

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

Table 3: The tuned hyper-parameters with search scopes.

Name of Parameter	Search Scope	Best
training batchsize	{1,2,4,8}	2
#epoch	{9,10,11,12,13}	12
#head in graph transformer	{4,5,6,7,8}	5
#layer in graph transformer	{4,5,6,7,8}	5
max sequence length	{128,256,512}	256
learning rate for RoBERTa	{4e-6. 5e-6, 6e-6, 5e-5}	5e-6

Experiment

Table 4: Experimental results on ReClor dataset. The percentage signs (%) of accuracy values are omitted. The optimal and sub-optimal results are marked in bold and underline respectively (same for the following tables).

Model	Valid	Test	Test-E	Test-H
Random	25.00	25.00	25.00	25.00
Human Performance[37]	-	63.00	57.10	67.20
BERT-Large [37]	53.80	49.80	72.00	32.30
XLNet-Large[37]	62.00	56.00	75.70	40.50
RoBERTa-Large [37]	62.60	55.60	75.50	40.00
DAGN [12]	65.80	58.30	75.91	44.46
FocalReasoner [21]	66.80	58.90	77.05	44.64
LReasoner [31]	66.20	62.40	-	-
Logiformer	**68.40**	**63.50**	79.09	51.25

Table 5: Experimental results on LogiQA dataset.

Model	Valid	Test
Random	25.00	25.00
Human Performance[17]	-	86.00
BERT-Large [17]	34.10	31.03
RoBERTa-Large [17]	35.02	35.33
DAGN [12]	36.87	39.32
FocalReasoner [21]	41.01	40.25
Logiformer	**42.24**	**42.55**

Experiment

Table 6: Ablation Studies. The improvements on the accuracy are marked in red.

Model	ReClor Valid	Δ	ReClor Test	Δ	Test-E	Test-H	LogiQA Valid	Δ	LogiQA Test	Δ
Logiformer	68.40	-	63.50	-	79.09	51.25	42.24	-	42.55	-
a) Graph Construction										
w/o syntax graph	66.40	-2.00	61.20	-2.30	77.50	48.39	38.56	-3.68	38.71	-3.84
w/o logical graph	63.60	-4.80	59.90	-3.60	75.00	48.04	38.25	-3.99	37.63	-4.92
b) Graph Transformer										
w/o co-occurrence bias	66.80	-1.60	62.80	-0.70	77.05	51.61	41.94	-0.30	42.55	-
w/o causal bias	65.20	-3.20	63.30	-0.20	76.82	52.68	39.94	-2.30	41.47	-1.08
w/o both of attention biases	66.20	-2.20	61.60	-1.90	75.23	50.89	41.63	-0.61	39.94	-2.61
c) Decoder										
w/o dynamic gates	67.00	-1.40	61.90	-1.60	76.14	50.71	41.32	-0.92	42.55	-
w/o question-aware attention	66.60	-1.80	60.40	-3.10	76.36	47.86	41.63	-0.61	42.09	-0.46

Experiment

Table 7: The details of ReClor Test Split on different question types. NA: Necessary Assumption, S:Strengthen, W:Weaken, I:Implication, CMP:Conclusion/Main Point, MSS:Most Strongly Supported, ER:Explain or Resolve, P:Principle, D:Dispute, R:Role, IF:Identify a Flaw, O:Others.

Model	NA	S	W	I	CMP	MSS	ER	P	D	R	IF	O
Logiformer	74.56	64.89	55.75	45.65	75.00	66.07	61.90	69.23	70.00	75.00	58.12	60.27
w/o syntax graph	70.18	59.57	55.75	45.65	66.67	57.14	67.86	56.92	56.67	50.00	62.39	57.53
Δ	-4.38	-5.32	-	-	-8.33	-8.93	+5.96	-12.31	-13.33	-25.00	+4.27	-2.74
w/o logical graph	68.42	61.70	51.33	41.30	66.67	51.79	59.52	55.38	43.33	59.38	63.25	65.75
Δ	-6.14	-3.19	-4.42	-4.34	-8.33	-14.28	-2.38	-13.85	-26.67	-15.62	+5.13	+5.48

Experiment

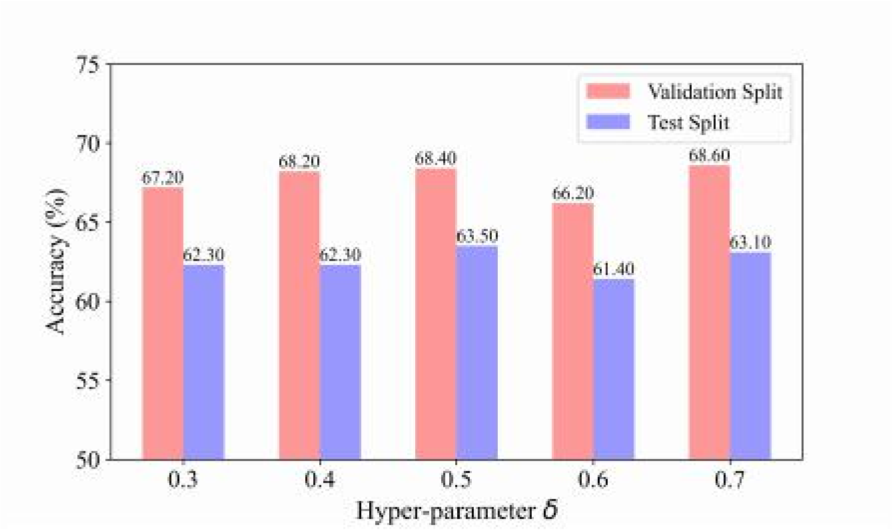


Figure 4: The model performances on the ReClor dataset under different δ.

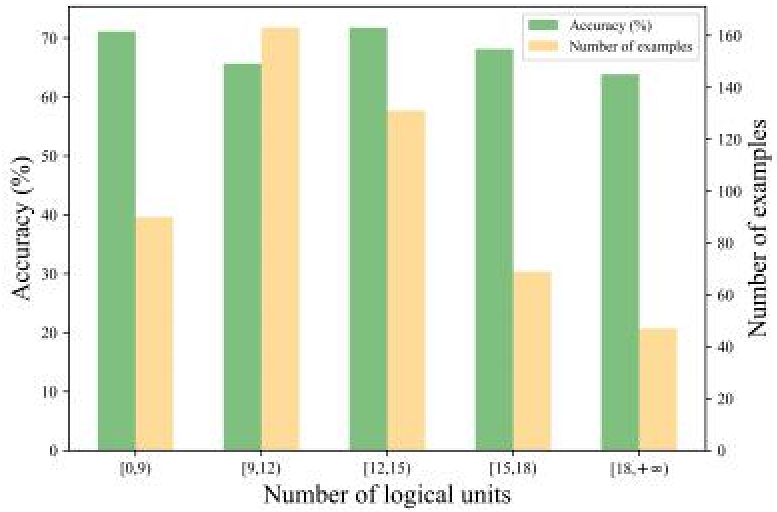


Figure 5: The model performances on under different numbers of logical units.

Experiment

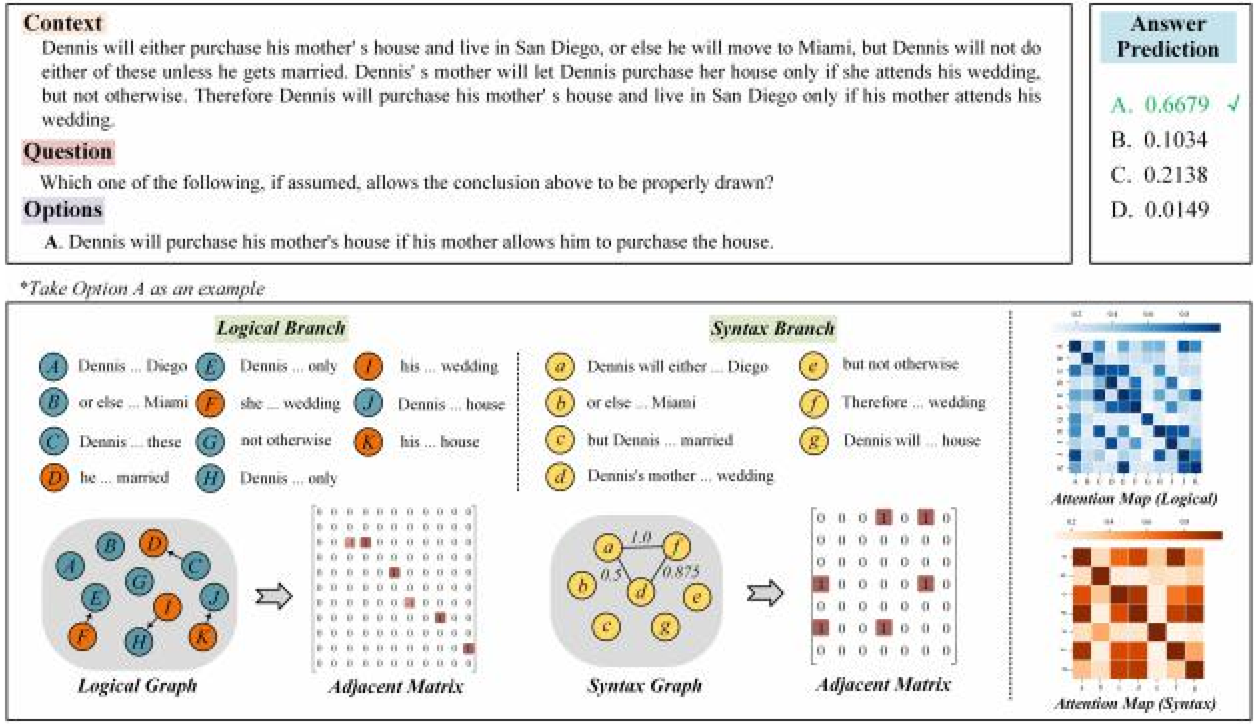


Figure 6: The illustration of an successful case. The interpretability of Logiformer lies in the logical units in text with explicit relations and the visualization of the weighted attention maps.

Thank you!