Multi-grained Logical Graph Network for Reasoning-Based Machine Reading Comprehension

Jiaqi Wang[1], Jia Zhong[1](✉), Hong Yin[1], Chen Wang[1], Qizhu Dai[1], Yang Xia[2], and Rongzhen Li[1]

[1] Chongqing University, Chongqing, China
{wangjiaqi,zhongjiang,yinhong,chenwang,daiqizhu,lirongzhen}@cqu.edu.cn
[2] Chongqing Academy of Big Data Co., Ltd., Chongqing, China

Abstract. Machine reading comprehension (MRC) is a crucial and challenging task in natural language processing (NLP). In order to equip machines with logical reasoning abilities, the challenging logical reasoning tasks are proposed. Existing approaches use graph-based neural models based on either sentence-level or entity-level graph construction methods which designed to capture a logical structure and enable inference over it. However, sentence-level methods result in a loss of fine-grained information and difficulty in capturing implicit relationships, while entity-level methods fail to capture the overall logical structure of the text. To address these issues, we propose a multi-grained graph-based mechanism for solving logical reasoning MRC. To combine the advantages of sentence-level and entity-level information, we mine elementary discourse units (EDUs) and entities from texts to construct graph, and learn the logical-aware features through a graph network for subsequent answer prediction. Furthermore, we implement a positional embedding mechanism to enforce the positional dependence, which facilitates logical reasoning. Our experimental results demonstrate that our approach provides significant and consistent improvements via multi-grained graphs, outperforming competitive baselines on both ReClor and LogiQA benchmarks.

Keywords: Machine Reading comprehension · Logical Reasoning · Multi-grained Graph

1 Introduction

Machine Reading Comprehension (MRC) is a fundamental task in Natural Language Processing (NLP) that seeks to teach machines to comprehend the meaning of human text and answer questions [50]. With the advancement of unsupervised learning and pre-trained language models (LM), many neural methods have achieved remarkable success on inchoate and simple datasets. For instance, BERT [7] has outperformed human performance in SQuAD [27]. Recently, MRC

X. Yang et al. (Eds.): ADMA 2023, LNAI 14179, pp. 47–62, 2023.
https://doi.org/10.1007/978-3-031-46674-8_4

tasks have become more challenging by raising the difficulty of contexts and questions, which aim to better evaluate model capabilities, such as multi-hop reasoning [17,23,39,43], numerical reasoning [8,52] and commonsense reasoning [13,35].

In addition to the above capabilities, logical reasoning is also a crucial aspect of human intelligence that plays a significant role in cognition and judgment [21]. It was also a primary research topic in the early days of AI [12,22]. However, most existing MRC models struggle to capture the logical structure of contexts due to the lack of logical reasoning ability. This limitation often leads to poor performance in logical reasoning MRC questions. To drive the development of logical reasoning, ReClor [48] and LogiQA [21] were proposed. These two datasets are multiple-choice MRC datasets, constructed by selecting logical reasoning questions from standardized exams. A logical reasoning problem example from LogiQA dataset is shown in Fig. 1. It contains a context, a question, and four answer options, among which only one option is correct.

Context

Left-handed people suffer from immune disorders, such as allergies, more often than right-handed people. However, left-handers often have an advantage over right-handers in accomplishing tasks controlled by the right hemisphere of the brain, and most people's mathematical reasoning ability is strongly affected by the right hemisphere of the brain.

Question

If the above information is true, which one of the following assumptions can it best support?

Option

A. Most people with allergies or other immune disorders are left-handed rather than right-handed.
B. Most left-handed mathematicians have some kind of allergy.
C. The proportion of left-handed people who have stronger mathematical reasoning ability than the average is higher than the proportion of left-handed people who have weaker mathematical reasoning ability than the average.
D. The proportion of people with immune dysfunction such as allergies is higher than that of left-handed people or people with unusually good mathematical reasoning skills.

Fig. 1. A logical reasoning based MRC example from LogiQA dataset.

To solve this task, previous research has employed methods that involves mining logical units and constructing a logical structure to facilitate reasoning over the context and question, ultimately predicting the correct answer. The two main granularities of information used are sentence-level and entity-level. For instance, at the sentence-level, AdaLoGN [18] mines a set of elementary discourse units (EDUs) from texts, converts discourse relations to logical relations to construct graphs, and then extends the graphs based on some inference rules. Finally, it uses a graph neural network (GNN) [32] to predict the answer. For another instance, at the entity-level, FocalReasoner [25] mines "Entity-Predicate-Entity" triplets as fact units from each sentence in the context, and finds the co-reference relations among entities in fact units. It then constructs a supergraph on the top of fact units and enhances graph presentation by GNN to predict the answer.

Although previous works have made significant advancements in logical reasoning, they also have their limitations. The sentence-level method is at a coarse-grained level, simply averaging the EDUs vector as the node representation, which could cause the loss of fine-grained information [1,9], particularly for the keyword in texts. Moreover, there are instances where there are no explicit logical conjunctions like "because" and "if" in the texts, making it challenging for prior works to mine explicit logical relations, as shown in the example in Fig. 1. In the LogiQA dataset [21], 3092 data points out of 8678 data points could not mine any explicit logical relations. If the logical relations cannot be extracted, AdaLoGN can only rely on the adjacent relations between EDUs to construct graphs and can't derive implicit logical relation via logical reasoning, resulting in sparse graph construction, significantly affecting graph information interaction and answer prediction. Additionally, the entity-level method is at a fine-grained level, ignoring the overall logical structure of texts. It only focuses on entity-level information, neglecting sentence-level interaction.

Inspired by previous work [9,41,51], we found that combining sentence-level and entity-level information can be more comprehensive and effective to make full use of the information in the text, which can address the above problems. In this paper, we present a new approach, MLGNet, for logical reasoning-based MRC with a multi-grained graph, as the overall model architecture depicted in Fig. 2. The aim is to combine the advantages of sentence-level and entity-level information in texts to create a better logical reasoning-based MRC model. We propose to construct graphs that contain EDUs and entities and present their relations, and then use graph network to learn the logical-aware features for subsequent answer prediction. With such multi-grained graphs, we can **(i)** not only mine logical structure via sentence-level information but also focus on local perception via entity-level information; and **(ii)** capture the implicit relations of EDUs through entities simultaneously. Furthermore, nodes in graphs are not arranged in a sequence, which may lead to a loss of order information for EDUs mined from the text, especially with the introduction of entity nodes and relations between EDUs and entities. Therefore, we propose a spatial encoding mechanism to strengthen the positional dependency of EDUs.

The contributions of this paper are three-fold:

- We introduce a heterogeneous multi-grained logical graph (MLG) with a graph-based neural network to model the logical relations of texts and offer logical-aware features.
- We present a positional embedding mechanism to reinforce the positional information to facilitate logical reasoning.
- Our experiment results demonstrate that MLGNet can boost the performance compared with strong baselines on two datasets ReClor and LogiQA.

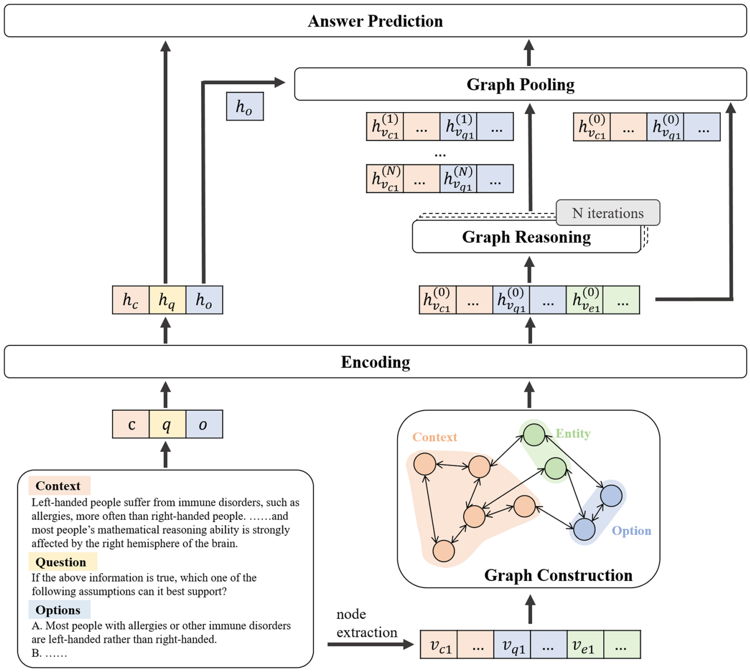

Fig. 2. Overall architecture of MLGNet.

2 Related Work

2.1 Machine Reading Comprehension

In recent years, there has been a surge of interest in complex machine reading comprehension (MRC) that evaluates model capabilities from various angles. For instance, HotpotQA [43], WikiHop [39], OpenBookQA [23], and MultiRC [17] require models to possess multi-hop reasoning capabilities, while DROP [8] and MA-TACO [52] require models to perform numerical reasoning. Besides, WIQA [35] and CosmosQA [13] test models' commonsense reasoning abilities. In addition to these abilities, logical reasoning is a crucial component of human intelligence and is receiving significant attention from researchers. Several MRC datasets that require logical reasoning have been proposed, including ReClor [48] and LogiQA [21]. ReClor is based on standardized exams such as GMAT and LSAT in the United States, while LogiQA is derived from the National Civil Servants Examination of China. These datasets contain 6138 and 8678 data points respectively, providing ample data to support logical reasoning in MRC.

2.2 Reasoning-Based MRC

Previous work has attempted to use semantic information extracted from text to construct logic graphs, which are then used to pass messages and update graph representations for answer prediction. These methods rely on two levels of information to construct graphs, sentence-level, and entity-level. For sentence-level, DAGN [14] constructs logic graphs using elementary discourse units split by discourse relations, but the chain-type graph is too sparse to facilitate effective node interaction. On this basis, AdaLoGN [18] extends the graph using symbolic logical reasoning to make it more densely connected, and Logiformer [42] constructs causal and syntax graphs simultaneously to capture logical and co-occurrence relations. However, these methods simply average the sentence vector as node representation, resulting in a loss of fine-grained information [1,9]. For entity-level, FocalReasoner [25] extracts fact units in the form of entity-predicate-entity triplets to construct supergraphs and updates nodes using GNNs, but it overlooks the logical relationship between sentences, which does not fully emphasize the logical structure of the text. Therefore, we propose to combine the advantages of sentence-level and entity-level information to build a multi-grained graph.

3 Methodology

In this work, we consider the multiple-choice MRC task, which can be described as a triplet $\langle c, q, O\rangle$, where c is a context, q is a question over c and O is a set of options. Our goal is to find only one correct option in O. Our framework is shown in Fig. 2. We first construct a multi-grained graph via texts, then conduct encoding and graph reasoning to make information fully interactive, and finally aggregate graph information for answer prediction.

3.1 Graph Construction

Multi-grained Logical Graph Definition. To model the logical information from text, a Multi-grained Logical Graph (MLG) is constructed. MLG is a directed graph, which can be represented as $G = \langle V, E\rangle$, where V is a set of nodes and E is a set of edges. MLG has two different kinds of nodes: EDUs V_E and entities V_e, where $V_E \cup V_e = V$. And the edges E present the relationship between EDUs and entities, which correspond to three situations.

- **Logic Edge:** Two EDUs having logical relation are connected with a logic edge. Similar to AdaLoGN [18], we set five types of common logical relations between EUDs as logical edge $L = \{conj, disj, impl, neg, rev\}$, where conjunction ($conj$), disjunction ($disj$), implication ($impl$), and negation ($neg$) are standard logical connectives in propositional logic and reversed implication (rev) is introduced to represent the inverse relation of $impl$.
- **Context Edge:** EDUs adjacent in the text, including the last EDU of c and the first EDU of o, are connected with a context edge. In this way, the context relations among the text could be modeled.
- **Containment Edge:** A EDU node is connected with a entity node if EDU containing the entity. With such connections, entity can enhance the representation of EDUs, at the same time EDUs containing the same entity can interact through the entity directly.

Multi-grained Logical Graph Construction. For each sample, we construct graph based on context and option, since the question is usually doesn't carry logical units in existing datasets [14].

For EDUs nodes, we follow the method of AdaLoGN [18], using Graphene [2] to mine EDUs from context and options, and mine the rhetorical relations between them. Rhetorical relations are mapping to logical relations via Table 1. For the relation $\langle v_i, impl, v_j \rangle$, we set the relation $\langle v_j, rev, v_i \rangle$. We also use syntactic rules based on part-of-speech tags and dependencies to find the EDUs which are negate each others, and connect them using *neg* relations.

Table 1. Mapping of logical relation with rhetorical relation.

Rhetorical relation	Logical relation
LIST, CONTRAST	*conj*
DISJUNCTION	*disj*
RESULT	*impl*
CAUSE, PURPOSE, CONDITION	*rev*

And for entity nodes, we employ an entity extractor based on part-of-speech tagging of each EDU, as nouns generally contain the richest semantic information in a sentence. We then select the top k nouns with the most occurrences as entity nodes, where k is a predefined hyper-parameter. For entities and EDUs containing this entity, we establish an *in* relation to indicate the EDU-entity containment relation.

To construct graph, we convert the relations extracted before to the edges in MLG. The logical relations between EDUs are converted to logic edges, and the *in* relations are converted into containment edge. Apart from that, for each EDUs that adjacent in text but are not connected with logic edge, we connect them with context edge. The last EDU of c and the first EDU of o are also connected with context edge.

3.2 Multi-grained Logical Graph Network

We propose the Multi-grained Logical Graph Network with the constructed graph to leverage the logical structure and multi-grained information of text for subsequent answer prediction. It consists of three module: graph encoding, graph filtering and graph pooling.

Graph Encoding. First of all, it's necessary to initialize representation for each node. Similar to previous works, we use RoBERTa [20] encoder to model graph nodes. It takes graph nodes as input and computes a context-aware representation for each token. Specifically, given V_c, V_o denoting EDUs nodes mining from context and option and V_e denoting entities node where $V_c \cup V_o = V_E$ and

$V_E \cup V_e = V$, we pack these nodes in to a single sequence and separate V_c, V_o, V_e with special tokens:

$$\begin{aligned} &[h_{\langle s\rangle}, h_{v_{1_1}}, \cdots, h_{|}, \cdots, h_{\langle/s\rangle}, h_{v_{|V_c|+1_1}}, \cdots, h_{\langle/s\rangle}, h_{v_{|V_E|+1_1}}, \cdots, h_{\langle/s\rangle}] \\ =&\text{RoBERTa}(\langle s\rangle v_{1_1} \cdots | \cdots \langle/s\rangle v_{|V_c|+1_1} \cdots \langle/s\rangle v_{|V_E|+1_1} \cdots \langle/s\rangle) \end{aligned} \tag{1}$$

where $\langle s\rangle$ and $\langle/s\rangle$ is the special tokens for RoBERTa and | is a special token to separate nodes inside V_c, V_o, V_e. For the representation of each node $v_i \in V$, we use the average hidden state.

$$h_{v_i} = \frac{1}{|v_i|}\sum_{j=1}^{|v_i|} h_{v_{i_j}} \tag{2}$$

For graphs always lack of sentence original position information, which aggravated with introduction of the entities nodes, we use the spatial encoding mechanism proposed in [42,46] to keep the original order information of EDUs in the text. Concretely, for each node $v_i \in V_E$ we compute the positional embeddings of v_i:

$$h_{v_i}^{(0)} = h_{v_i} + \text{PosEmbed}(idx(v_i)) \tag{3}$$

where $idx(v_i)$ returns the index of v_i, and PosEmbed() provides a $|V_E|$-dimensional embedding for each EDUs node. We take the result $h_{v_i}^{(0)}$ as initial representation of each node.

We also use the same pre-train language model RoBERTa as graph node encoding to model the texts in context, question and options for subsequent operation. Given context $c = \{c_i\}_{i=1}^{|c|}$, question $q = \{q_j\}_{j=1}^{|q|}$ and option $o = \{o_k\}_{k=1}^{|o|}$, we calculate for each token a context-aware representation:

$$\begin{aligned} &[h_{\langle s\rangle}, h_{c_1}, \cdots, h_{\langle/s\rangle}, h_{q_1}, \cdots, h_{\langle/s\rangle}, h_{o_1}, \cdots, h_{\langle/s\rangle}] \\ =&\text{RoBERTa}(\langle s\rangle\, c_1 \cdots \langle/s\rangle\, q_1 \cdots \langle/s\rangle\, o_1 \cdots \langle/s\rangle) \end{aligned} \tag{4}$$

We take the average embedding as output of the representations of c, q, o:

$$h_c = \frac{1}{|c|}\sum_{i=1}^{|c|} h_{c_i}, h_q = \frac{1}{|q|}\sum_{i=1}^{|q|} h_{q_i}, h_o = \frac{1}{|o|}\sum_{i=1}^{|o|} h_{o_i} \tag{5}$$

Graph Reasoning. We utilize an iterative neural reasoning method, proposed in [18], to extend previous constructed graph, as well as update nodes representation. We construct the graph described in Sect. 3.1 and then initialize the nodes representation described in Sect. 3.2. Since the entities nodes may introduce irrelevant information, we implement a entity selection strategy to filter the entities nodes in the graph to avoid too much noise. For each candidate entity node $v_i \in V_e$, we calculate the matching score between entity and the text to judge whether relevant to answering the question:

$$rel_e = sigmoid(linear(v_i \,||\, h_o)) \tag{6}$$

where $||$ represents vector concatenation. We set a predefined threshold τ_e to judge which entity we choose to construct graph. If $rel_e > \tau_e$, we select the entity.

Then, we feed the filtered graph into the iterative reasoning mechanism. In the $(n+1)$-th iteration, we start graph reasoning with the node representation $h_{v_i}^{(n)}$ from the n-th iteration. Since some logical relations are implicit in text which is hard to mine from text, we perform logical inference over the extracted explicit logical relations to derive implicit logical relation according to inference rules. Here we apply three logical equivalence rules:

- Transposition:
$$v_i \rightarrow v_j \Rightarrow \neg v_i \rightarrow \neg v_j \tag{7}$$
- Hypothetical Syllogism:
$$(v_i \rightarrow v_j) \wedge (v_j \rightarrow v_k) \Rightarrow v_i \rightarrow v_k \tag{8}$$
- Adjacency-Transmission:
$$(v_i \sim v_j) \wedge (v_j \mid v_k) \Rightarrow v_i \sim v_k \tag{9}$$
where $\sim \in \{\wedge, \vee, \rightarrow\}$ and $|$ represents the context edge, which is adjacency in text.

While these rules may cause misleading, we introduce a mechanism to judge whether the candidate extension is relevant to answering the question. For each candidate extension ϵ applied inference rule over a set of nods $V_\epsilon \in V$, we calculate the relevance score of ϵ:

$$\begin{aligned} rel_\epsilon &= sigmoid(linear(h_\epsilon \,||\, h_o)), \\ h_\epsilon &= \frac{1}{V_\epsilon} \sum_{v_i \in V_\epsilon} h_{v_i}^{(n)} \end{aligned} \tag{10}$$

where $||$ represents vector concatenation. We set a predefined threshold τ_ϵ to judge which extension can be admitted to extend graph. If $rel_\epsilon > \tau_\epsilon$, we accept this extension.

After graph extension, we performs to fuse the multi-grained information by interaction of nodes and update node representation from $h_{v_i}^{(n)}$ to $h_{v_i}^{(n+1)}$. Let $\mathcal{N}^i$ indicate the neighbors of node v_i, and $\mathcal{N}_r^i \subseteq \mathcal{N}^i$ indicate the subset under relation $r \in R$. The node representations are updated with message propagation mechanism in R-GCN [?]:

$$\begin{aligned} &h_{u_i}^{(n+1)} = ReLU(\sum_{r \in R} \sum_{v_j \in \mathcal{N}_r^i} \frac{\alpha_{i,j}}{\mathcal{N}_r^i} W_r^{(n)} h_{v_j}^{(n)} + W_0^n h_{v_j}^{(n)}), \text{ where} \\ &\alpha_{i,j} = softmax_{idx(a_{i,j})}([\cdots, a_{i,j}, \cdots]^T), \text{ for all } u_j \in N_i, \\ &a_{i,j} = LeakyReLU(linear(h_{v_i}^{(n)} \,||\, h_{v_j}^{(n)})), \end{aligned} \tag{11}$$

where $W_r^{(n)}, W_0^{(n)}$ are matrices and $idx(a_{i,j})$ returns the index of $a_{i,j}$.

Graph Pooling. After N iterations, for each node v_i we fuse the representation over all iterations:

$$h_{v_i}^{fus} = h_{v_i}^{(0)} + linear(h_{v_i}^{(1)}||\cdots||h_{v_i}^{(N)}) \tag{12}$$

In order to avoid the influence of entities on the text sequence, we only consider the representation of the node $v_i \in V_E$ and feed it into a bidirectional residual GRU layer [4], ignoring the representation of nodes $v_i \in V_e$.

$$[h_{v_i}^{fnl}, \cdots, h_{v_{|V_E|}}^{fnl}] = \text{Res-BiGRU}([h_{v_i}^{fnl}, \cdots, h_{v_{|V_E|}}^{fnl}]) \tag{13}$$

We aggregate the node representations by computing an o-attended weighted sum:

$$\begin{aligned} h_{V_E} &= \sum_{v_i \in V_E} \alpha_i h_{v_i}^{fnl}, \text{ where} \\ \alpha_i &= softmax_i([a_1, \cdots, a_{|V_E|}]^T), \\ a_i &= LeakyReLU(linear(h_o \,||\, h_{v_i}^{fnl})) \end{aligned} \tag{14}$$

We concatenate h_{V_E} and the relevance scores to form the representation of G:

$$\begin{aligned} h_G &= (h_{V_E} \,||\, rel_{\mathcal{E}^{(1)}} \,||\cdots||\, rel_{\mathcal{E}^{(N)}}), \text{ where} \\ rel_{\mathcal{E}^{(n)}} &= \frac{1}{\mathcal{E}^{(n)}} \sum_{\epsilon \in \mathcal{E}^{(n)}} rel_\epsilon \end{aligned} \tag{15}$$

where $\mathcal{E}^{(n)}$ is the set of candidate extensions in the n-th iteration.

3.3 Answer Prediction

To predict the correct answer, we concatenate the representation of text from backbone pre-train model and the representation of our Multi-grained Logical Graph.

$$score_o = linear(tanh(linear(h_c \,||\, h_q \,||\, h_o \,||\, h_G))) \tag{16}$$

where h_c, h_q, h_o is the results of Eq. 5 and $score_o$ is the final score of each option in one example. Finally we choose the option with the highest score as the predicted answer.

3.4 Loss Function

Let $o_t \in O$ be the ground truth of the sample. We use cross-entropy loss with label smoothing optimizing.

$$\begin{aligned} \mathcal{L} &= -(1-\gamma) score'_{o_t} - \gamma \frac{1}{|O|} \sum_{o_i \in O} score'_{o_i}, \text{ where} \\ score'_{o_i} &= \log \frac{\exp(score_{o_i})}{\sum_{o_j \in O} \exp(score_{o_j})} \end{aligned} \tag{17}$$

where γ is a predefined smoothing factor.

4 Experiment

4.1 Datasets

We evaluate the performance on two logical reasoning based MRC datasets: ReClor [48] and LogiQA [21].

- **ReClor:** The Reading Comprehension dataset requiring logical reasoning for reasoning-based MRC. It consists of 6138 four-option multi-choice questions sourced from actual exams of GMAT and LSAT, which are split into 4638 for training, 500 for validation and 1000 for testing. In order to fully assess the logical reasoning ability, the dataset divided into EASY set and HARD set according to the performance of pre-trained language models.
- **LogiQA:** It consists of 8678 four-option multi-choice questions sourced from National Civil Servants Examination of China, which are split into 7376 for training, 651 for validation and 651 for testing.

4.2 Baselines

To compare our multi-grained graph-based method with prior work, we main employ several sentence-level and entity-level baselines on logical reasoning based MRC task as follow:

- **DAGN** [14]: It propose a discourse-aware graph network that reasongs relying on the extracted discourse structure of texts, which used the sentence-level information of texts, and facilitates logical reasoning via graph neural networks.
- **FocalReasoner** [25]: It defines and extracts fact units from text, which are the entity-level information of text, to construct a supergraph, and enhance the supergraph with graph attention network.
- **AdaLoGN** [18]: It extracts the discourse structure and the explicit logical relation, and further extend them to find implicit logical relation based on several logical rules via a iterative mechanism, which is realized on sentence-level information.
- **Logiformer** [42]: It utilizes two different strategies to extract logic and syntax units, and construct the logical graph and the syntax graph respectively. After that it feed the extracted node sequence to the fully connected transformer to each graph, and use a dynamic gate mechanism to fuse the features from two branches.

4.3 Overall Results

Table 2 presents the overall results of the logical reasoning-based MRC task, comparing our method with baselines such as sentence-level methods DAGN, AdaLoGN, Logiformer, and entity-level method FocalReasoner. Our multi-grained graphs approach achieve the best performance among all other graph-based method on both ReClor and LogiQA. MLGNet reaches 64.07% of test accuracy on ReClor, and reaches 43.39% of test accuracy on LogiQA. Specifically, MLGNet achieves the highest test accuracy among all models, with 64.07% on ReClor and 43.39% on LogiQA. These results confirm our hypothesis that multi-grained graphs are effective in capturing and utilizing the logical structure of texts.

Table 2. Experimental results (accuracy %) compared with baselines on ReClor and LogiQA.

Methods	ReClor				LogiQA	
	Valid	Test	Test-E	Test-H	Valid	Test
DAGN	65.80	58.30	75.91	44.46	36.87	39.32
FocalReasoner	66.80	58.90	77.05	44.64	41.01	40.25
AdaLoGN	65.20	60.20	79.32	45.18	39.94	40.71
Logiformer	68.40	63.50	79.09	51.25	42.24	42.55
MLGNet	70.02	64.07	79.32	51.60	43.08	43.39

4.4 Ablation Study

We design an ablation study to verify the feasibility of the main contributions in our method: multi-grained logical graph construction and positional embedding mechanism. The results are reported in Table 3.

Table 3. Ablation study results (accuracy %) on LogiQA.

Methods	**Valid**	Δ	**Test**	Δ
MLGNet	43.08		43.39	
multi-grained logical graph				
MLGNet w/o entities	40.70	-2.38	41.08	-2.31
MLGNet w/ all entities	41.31	-1.77	42.20	-1.19
positional embedding				
MLGNet w/o position	42.93	-0.15	42.93	-0.46

Multi-grained Logical Graph. In graph construction, we build multi-grained graph by selecting and introducing entities as nodes on the existing methods using EDUs, hence we ablate the effects of whether introducing entities nodes and whether selecting entities nodes. Using the modified graphs with introducing no entities or introducing all extracted entities, the results show that the performance all decrease whether introduce no entities or all entities. This verifies the feasibility of multi-grained logical graph construction and entity selection strategy.

Positional Embedding. We remove the positional embedding and only use the average of RoBERTa outputs as node initial representation. The accuracy results decrease 0.15% in dev set and 0.46% in test set, which indicates positional embedding beneficial for subsequent graph reasoning and graph pooling.

4.5 Effect of Entities Nodes Introduction

To evaluate the effectiveness of entity nodes introduction, we compare MLGNet with other MRC models. We suspected that our method would be more effective for data points where explicit logical relations could not be extracted. To verify this, we split the original dev set and test set of LogiQA into four subsets based on the number of extracted explicit logical relations, as the statistics shown in Table 4. We display the accuracy of AdaLoGN and our proposed MLGNet in Fig. 3, and find that our model outperforms the baseline models on all divided subsets, demonstrating the effectiveness of our model for different extracted explicit logical relations. While MLGNet performs better when the number of extracted explicit logical relations is in the range of $[0, 3)$ and $[3, 6)$, the reason for this could be that our method effectively supplements information when the available information is less.

Table 4. Distribution of explicit logical relations on dev set and test set of LogiQA.

Dataset	$[0, 3)$	$[3, 6)$	$[6, 9)$	$[9, \infty)$
LogiQA-dev	55.4%	20.0%	12.1%	12.5%
LogiQA-test	52.8%	25.0%	11.8%	10.4%

4.6 Case Study

This section provides a case study, using the example described in Fig. 1 which is fail with previous works but successful with our method, to vividly show the effectiveness of our method. The case is shown in Fig. 4. We totally extract six nodes based on Graphene and part-of-speech tags, including five EDUs nodes and one entity node. Among them four pairs of context edges (U1-U2, U2-U3, U3-U4, U4-U5) and two pairs of containment edges (U1-U6, U5-U6) are detected. We can see that MLGNet can build a bridge for context and option to interact with each other, i.e. the path U1-U6-U5, especially in the graph without logic edge. In the same time, the entity “allergies” is the key word of sentence that it appears, so the entity node can also enhance vital information to the sentences.

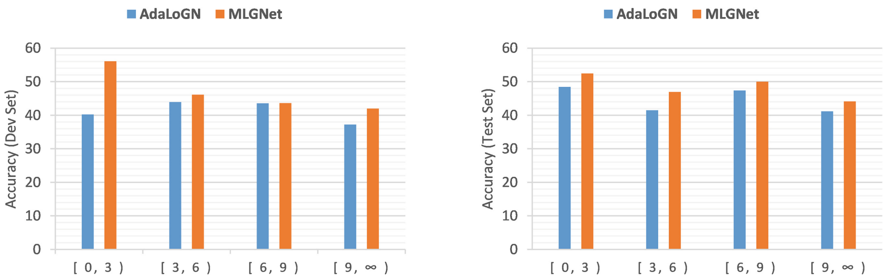

Fig. 3. Accuracy of models on number of extracted explicit logical relations on dev set (left) and test set (right) of LogiQA.

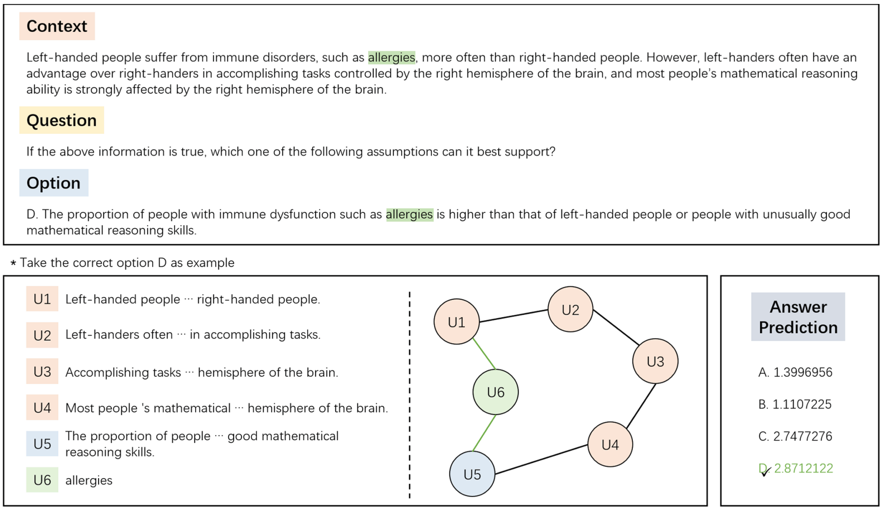

Fig. 4. A successful example in our method.

5 Conclusion

This paper presents MLGNet, a novel approach for logical reasoning based machine reading comprehension (MRC) that leverages both sentence-level and entity-level information. The approach involves the extraction of elementary discourse units (EDUs) and entity nodes, and the construction of multi-grained logical graphs containing three types of relations between nodes. An entity selection process is applied to filter the entity nodes, and the resulting graphs are used to facilitate information interaction and prediction. The proposed multi-grained graph-based mechanism effectively captures the logical structure of texts, and a positional embedding mechanism is employed to intensify the positional dependency of EDUs. The results show that MLGNet outperforms baseline models on two datasets ReClor and LogiQA. This study represents the first exploration of multi-grained graph-based methods for logical reasoning, which investigate and demonstrate the feasibility of multi-grained logical graphs for logical reasoning MRC, opening up potential avenues for future research.

Acknowledgements. The authors would like to thank the Associate Editor and anonymous reviewers for their valuable comments and suggestions. This work is funded in part by the National Natural Science Foundation of China under Grants No.62176029, and in part by the graduate research and innovation foundation of Chongqing, China under Grants No.CYB21063. This work also is supported in part by the Chongqing Technology Innovation and Application Development Special under Grants CSTB2022TIAD-KPX0206.

References

1. Cai, L., Li, J., Wang, J., Ji, S.: Line Graph Neural Networks for Link Prediction (2020)
2. Cetto, M., Niklaus, C., Freitas, A., Handschuh, S.: Graphene: Semantically-Linked Propositions in Open Information Extraction (2018)
3. Chen, K., et al.: Question Directed Graph Attention Network for Numerical Reasoning over Text (2020)
4. Cho, K., et al.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation (2014)
5. Chowdhury, G.G.: Introduction to Modern Information Retrieval. Facet Publishing (2010)
6. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs (2019)
7. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 (2019)
8. Dua, D., et al.: DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs (2019)
9. Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium, pp. 3433–3442. Association for Computational Linguistics (2018)
10. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems, vol. 28. Curran Associates Inc. (2015)
11. Hirschman, L., Light, M., Breck, E., Burger, J.D.: Deep Read: a reading comprehension system. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. College Park, Maryland, USA, pp. 325–332. Association for Computational Linguistics (1999)
12. "History of programming languages–II — ACM Other Books. https://dl.acm.org/doi/abs/10.1145/234286
13. Huang, L., Bras, R.L., Bhagavatula, C., Choi, Y.: Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning (2019)
14. Huang, Y., Fang, M., Cao, Y., Wang, L., Liang, X.: DAGN: Discourse-Aware Graph Network for Logical Reasoning (2021)
15. Jia, R., Cao, Y., Tang, H., Fang, F., Cao, C., Wang, S.: Neural extractive summarization with hierarchical attentive heterogeneous graph network. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3622–3631. Online: Association for Computational Linguistics (2020)

16. Jiao, F., Guo, Y., Song, X., Nie, L.: MERIt: Meta-Path Guided Contrastive Learning for Logical Reasoning (2022)
17. Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., Roth, D.: Looking beyond the surface: a challenge set for reading comprehension over multiple sentences. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 252–262. Association for Computational Linguistics, New Orleans, Louisiana (2018)
18. Li, X., Cheng, G., Chen, Z., Sun, Y., Qu, Y.: AdaLoGN: Adaptive Logic Graph Network for Reasoning-Based Machine Reading Comprehension (2022)
19. Liu, S., Zhang, X., Zhang, S., Wang, H., Zhang, W.: Neural machine reading comprehension: methods and trends. Appl. Surf. Sci. **471**, 18–22 (2019)
20. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019)
21. Liu, J., Cui, L., Liu, H., Huang, D., Wang, Y., Zhang, Y.: LogiQA: a challenge dataset for machine reading comprehension with logical reasoning (2020)
22. McCarthy, J.: Artificial intelligence, logic and formalizing common sense. In: Thomason, R.H. (ed.) Philosophical Logic and Artificial Intelligence, pp. 161–190. Springer, Netherlands, Dordrecht (1989). https://doi.org/10.1007/978-94-009-2448-2_6
23. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? A new dataset for open book question answering (2018)
24. Nguyen, T. et al.: MS MARCO: A Human Generated MAchine Reading COmprehension Dataset (2016)
25. Ouyang, S., Zhang, Z., Zhao, H.: Fact-driven Logical Reasoning (2021)
26. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
27. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text (2016)
28. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: unanswerable questions for SQuAD (2018)
29. Ran, Q., Lin, Y., Li, P., Zhou, J., Liu, Z.: NumNet: machine reading comprehension with numerical reasoning (2019)
30. Riloff, E., Thelen, M.: A rule-based question answering system for reading comprehension tests. In: ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems (2000)
31. Sahu, S.K., Christopoulou, F., Miwa, M., Ananiadou, S.: Inter-sentence relation extraction with document-level graph convolutional neural network (2019)
32. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2009)
33. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional Attention Flow for Machine Comprehension (2018)
34. Sugawara, S., Aizawa, A.: An analysis of prerequisite skills for reading comprehension. In: Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods, pp. 1–5. Association for Computational Linguistics, Austin, TX (2016)
35. Tandon, N., Mishra, B.D., Sakaguchi, K., Bosselut, A., Clark, P.: WIQA: a dataset for "What if..." reasoning over procedural text (2019)
36. Wang, S., Jiang, J.: Machine Comprehension Using Match-LSTM and Answer Pointer (2016)
37. Wang, S., et al.: Logic-driven context extension and data augmentation for logical reasoning of text (2021)

38. Wang, S., et al.: From LSAT: the progress and challenges of complex reasoning (2021)
39. Welbl, J., Stenetorp, P., Riedel, S.: Constructing datasets for multi-hop reading comprehension across documents. Trans. Assoc. Comput. Linguist. **6**, 287–302 (2018)
40. Xu, K., Wu, L., Wang, Z., Yu, M., Chen, L., Sheinin, V.: Exploiting rich syntactic information for semantic parsing with graph-to-sequence model (2018)
41. Xu, M., Li, L., Wong, D.F., Liu, Q., Chao, L.S.: Document graph for neural machine translation (2021)
42. Xu, F., Liu, J., Lin, Q., Pan, Y., Zhang, L.: Logiformer: a two-branch graph transformer network for interpretable logical reasoning. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1055–1065 (2022)
43. Yang, Z., et al.: HotpotQA: a dataset for diverse, explainable multi-hop question answering (2018)
44. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding
45. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. Proc. AAAI Conf. Artif. Intell. **33**(01), 7370–7377 (2019)
46. Ying, C.T., et al.: Transformers really perform bad for graph representation?"
47. Yu, A.W., et al.: QANet: combining local convolution with global self-attention for reading comprehension (2018)
48. Yu, W., Jiang, Z., Dong, Y., Feng, J.: ReClor: a reading comprehension dataset requiring logical reasoning (2020)
49. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction (2020)
50. Zhang, Z., Zhao, H., Wang, R.: Machine reading comprehension: the role of contextualized language models and beyond. arXiv:2005.06249 (2020)
51. Zheng, B., et al.: Document Modeling with Graph Attention Networks for Multi-grained Machine Reading Comprehension (2020)
52. Zhou, B., Khashabi, D., Ning, Q., Roth, D.: Going on a vacation takes longer than "Going for a walk": a study of temporal commonsense understanding (2019)