Multi-hop Reading Comprehension across Documents with Path-based Graph Convolutional Network

Zeyun Tang , **Yongliang Shen** , **Xinyin Ma** , **Wei Xu** , **Jiale Yu** and **Weiming Lu**[*]
College of Computer Science and Technology, Zhejiang University, China
luwm@zju.edu.cn

Abstract

Multi-hop reading comprehension across multiple documents attracts much attention recently. In this paper, we propose a novel approach to tackle this multi-hop reading comprehension problem. Inspired by human reasoning processing, we construct a path-based reasoning graph from supporting documents. This graph can combine both the idea of the graph-based and path-based approaches, so it is better for multi-hop reasoning. Meanwhile, we propose Gated-RGCN to accumulate evidence on the path-based reasoning graph, which contains a new question-aware gating mechanism to regulate the usefulness of information propagating across documents and add question information during reasoning. We evaluate our approach on WikiHop dataset, and our approach achieves state-of-the-art accuracy against previously published approaches. Especially, our ensemble model surpasses human performance by 4.2%.

1 Introduction

Machine reading comprehension has been a popular topic in the past years, and a variety of models have been proposed to address this problem, such as BIDAF [Seo *et al.*, 2017], Reinforced mnemonic reader [Hu *et al.*, 2017], and ReasoNet [Shen *et al.*, 2017]. However, most existing works focus on finding evidence and answer in a single document.

In fact, in reality, many questions can only be answered after reasoning across multiple documents. Table 1 shows a multi-choice style reading comprehension example, which is from WIKIHOP dataset [Welbl *et al.*, 2018]. In the example, we can only answer the question '*what is the place of death of alexander john ellis?*' after extracting and integrating the facts '*Alexander John Ellis is buried in Kensal Green Cemetery*' and '*Kensal Green Cemetery is located in Kensington*' from multiple documents, which is a more challenging task.

Question: place of death, alexander john ellis, ?
Support doc1: Alexander John Ellis, was an English mathematician ... is buried in Kensal Green Cemetery. **Support doc2**: The areas of College Park and Kensal Green Cemetery are located in the London boroughs of Hammersmith & Fulham and Kensington & Chelsea, respectively. ......
Candidates: college park, france, Kensington, London
Answer: Kensington

Table 1: An example of multi-hop reading comprehension across documents.

The main challenge is that the evidence is distributed in different documents and there is a lot of noise in the documents. We need to extract this evidence from multiple documents, but it is difficult to capture their dependencies for reasoning. Many works used graph convolution networks(GCNs) to deal with this problem, such as Entity-GCN [De Cao *et al.*, 2019], BAG [Cao *et al.*, 2019] and HDE [Tu *et al.*, 2019]. They transform documents into an entity graph, and then import the entity graph into graph convolution networks(GCNs) to simulate the process of multi-hop reasoning.

However, these GCN-based approaches have some disadvantages. Firstly, they generated the entities only from the question and candidate answers, lacking much key information for multi-hop reasoning. For example, as the example in Table 1, the entity '*Kensal Green Cemetery*' is an important clue to answer the question, but the above approaches ignored this information. Secondly, the traditional GCNs only update the central node based on the aggregated information of adjacent nodes and use this to simulate the process of reasoning. But the question information is not fully utilized and there is a lot of irrelevant information during information propagating across documents in the multi-hop reasoning.

In this paper, we propose a novel approach to solve the above problem. We introduce a path-based reasoning graph for multiple documents. Compared to traditional graphs, the path-based reasoning graph contains multiple reasoning paths from questions to candidate answers, combining both the idea of the GCN-based and path-based approaches. Thus, we construct a path-based reasoning graph by extracting reasoning paths(e.g., *Alexander John Ellis* → *Kensal Green Cemetery* → *Kensington*) from supporting documents and then adding reasoning nodes (e.g., *Kensal Green Cemetery*) in these path-

[*]corresponding author

s to the entity graph. And then, we apply a Gated-RGCN to learn the representation of nodes. Compared to GCNs, Gated-RGCN utilizes attention and question-aware gating mechanism to regulate the usefulness of information propagating across documents and add question information during reasoning, which is closer to human reasoning processes.

Our contributions can be summarized as follows:

- We propose a path-based reasoning graph, which introduces information about reasoning paths into the graph;
- We propose Gated-RGCN to optimize the convolution formula of RGCN, which is more suitable for multi-hop reading comprehension;
- We evaluated our approach on WIKIHOP dataset [Welbl *et al.*, 2018], and our approach achieves new state-of-the-art accuracy. Especially, our ensemble model surpasses the human performance by 4.2%.

2 Related Work

Recently, there are several categories of approaches that have been proposed to tackle the problem of multi-hop reading comprehension across documents, including GCN-based approaches (Entity-GCN [De Cao *et al.*, 2019], BAG [Cao *et al.*, 2019], HDE [Tu *et al.*, 2019], MHQA-GRN [Song *et al.*, 2018], DFGN [Qiu *et al.*, 2019]), memory based approaches (Coref-GRU [Dhingra *et al.*, 2018], EPAr [Jiang *et al.*, 2019]), path based approaches (PathNet [Kundu *et al.*, 2019]), and attention based approaches (CFC [Zhong *et al.*, 2019], DynSAN [Zhuang and Wang, 2019]).

GCN-based approaches organize supporting documents into a graph, and then employ Graph Neural Networks based message passing algorithms to perform multi-step reasoning. For example, Entity-GCN [De Cao *et al.*, 2019] constructed an entity graph from supporting documents, where nodes are mentions of subject entity and candidates, and edges are relations between mentions. BAG [Cao *et al.*, 2019] applied bi-directional attention between the entity graph and the query after GCN reasoning over the entity graph. HDE [Tu *et al.*, 2019] constructed a heterogeneous graph where nodes correspond to candidates, documents, and entities. MHQA-GRN [Song *et al.*, 2018] constructed a graph where each node is either an entity mention or a pronoun representing an entity, and edges fall into three types: same-typed, window-typed and coreference-typed edge. DFGN [Qiu *et al.*, 2019] proposed a dynamic fusion reasoning block based on graph neural networks. Our work proposes Gated-RGCN to optimize the graph convolution operation, it is better for regulating the usefulness of information propagating across documents and add question information during reasoning.

Memory-based approaches try to aggregate evidences for each entity from multiple documents through a memory network. For example, Coref-GRU [Dhingra *et al.*, 2018] aggregated information from multiple mentions of the same entity by incorporating coreference in the GRU layers. EPAr [Jiang *et al.*, 2019] used a hierarchical memory network to construct a 'reasoning tree', which contains a set of root-to-leaf reasoning chains, and then merged evidences from all chains to make the final prediction.

PathNet [Kundu *et al.*, 2019] proposed a typical path-based approach for multi-hop reading comprehension. It extracted paths from documents for each candidate given a question, and then predicted the answer by scoring these paths. Our work introduces the idea of path-based approach on GCN-based approach which is better for multi-hop reasoning.

CFC [Zhong *et al.*, 2019] and DynSAN [Zhuang and Wang, 2019] are two typical attention-based approaches. CFC applied co-attention and self-attention to learn query aware node representations of candidates, documents and entities. While DynSAN proposed a dynamic self-attention architecture to determine what tokens are important for constructing intra-passage or cross-passage token level semantic representations. In our work, we employ an attention mechanism between graphs and the question at each layer of Gated-RGCN.

Meanwhile, in order to promote the research on multi-hop QA, several datasets have been designed, including WikiHop [Welbl *et al.*, 2018], OpenBookQA [Mihaylov *et al.*, 2018], NarrativeQA [Kociský *et al.*, 2017], MultiRC [Khashabi *et al.*, 2018] and HotpotQA [Yang *et al.*, 2018]. For example, WikiHop is a multi-choice style reading comprehension data set, where the task is to select the correct object entity from candidates when given a query $\langle s, r, ?\rangle$ and a set of supporting documents. While OpenBookQA focuses on the multi-hop QA which needs a corpus of provided science facts (open book) with external broad common knowledge.

In addition, *knowledge completion over knowledge graph (KG)* and *KG based query answering* are also related to our task, since they both need multi-hop reasoning, i.e., finding the reasoning path between two entities in KG. For example, MINERVA [Das *et al.*, 2017] formulates the multi-hop reasoning as a sequential decision problem, and uses the REINFORCE algorithm [Williams, 1992] to train an end-to-end model for multi-hop KG query answering. Meta-KGR [Lv *et al.*, 2019] also used the reinforcement learning method to learn a relation-specific multi-hop reasoning agent to search for reasoning paths and target entities. They further used meta-learning to perform multi-hop reasoning over few-shot relations of knowledge graphs.

3 Approach

In this section, we first formulate the task of multi-hop reading comprehension across documents, and then elaborate our approach in detail.

3.1 Task Formulation

The task of multi-hop reading comprehension across documents can be formally defined as: given a question $q = (q_1, q_2, ..., q_M)$ and a set of supporting documents S_q, the task is to find the correct answer a from a set of answer candidates $C_q = (c_1, c_2, ..., c_N)$, where M is the number of words in the question q and N is the number of candidates in C_q.

In the WIKIHOP dataset [Welbl *et al.*, 2018], the question q is given in the form of a tuple $\langle s, r, ?\rangle$, where s represents the subject entity, and r represents the relation between s and the unknown tail entity. In our example, $q =$

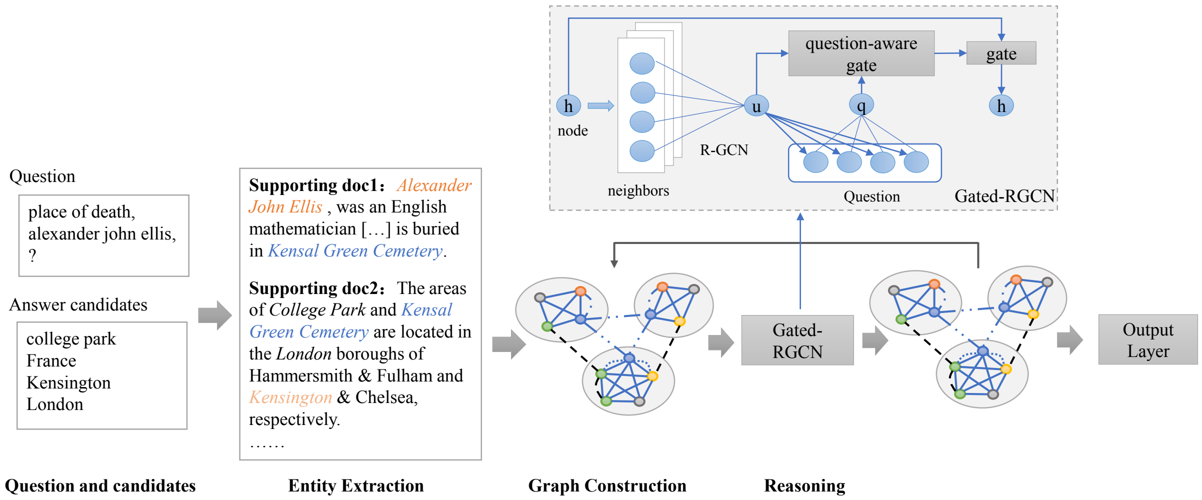

Question
place of death,
alexander john ellis,
?
Answer candidates
college park
France
Kensington
London
Supporting doc1：Alexander John Ellis , was an English mathematician […] is buried in Kensal Green Cemetery.
Supporting doc2：The areas of College Park and Kensal Green Cemetery are located in the London boroughs of Hammersmith & Fulham and Kensington & Chelsea, respectively.
……
h
node
neighbors
R-GCN
u
question-aware gate
q
Question
gate
h
Gated-RGCN
Gated-RGCN
Output Layer
Question and candidates
Entity Extraction
Graph Construction
Reasoning

Figure 1: Overview of our approach

$\langle alexander\ john\ ellis, place\ of\ death, ?\rangle$ means *where did alexander john ellis die*, and the answer candidates is $C_q = (college\ park, france, Kensington, London)$. When given the supporting documents, e.g., *supporting doc1, doc2* in Table 1, we should identify the correct answer $a = Kensington$ from the candidates by reasoning across these documents.

As shown in Figure 1, our approach mainly consists of three components, including graph construction, reasoning with Gated-RGCN, and output layer. In the following sections, we will elaborate on each component in detail.

3.2 Graph Construction

We construct an entity graph based on the Entity-GCN [De Cao *et al.*, 2019], which extracts all mentions of entities in $C_q \cup \{s\}$ in S_q as nodes in the graph. Besides, inspired by the human reasoning processing, *reasoning paths* from the subject entity in question to the candidates could be helpful for reasoning across documents, so we add *reasoning entities* in the paths into our entity graph. In our example, the path *alexander john ellis* → *Kensal Green Cemetery* → *Kensington* from documents indicate that the candidate *Kensington* may be the correct answer for the question ⟨*alexander john ellis, place of death, ?* ⟩. Thus, we treat *Kensal Green Cemetery* as a reasoning entity, and add it into the entity graph.

Formally, for a given question $q = \langle s, r, ?\rangle$, we would like to extract paths from s to $c_i \in C_q$ from S_q, e.g., $p_i = s \rightarrow e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_l \rightarrow c_i$, where e_i is a reasoning entity. In order to find a path, we first find a document d_1 which contains the mention m_s of the subject entity s in S_q, and then find all the named entities and noun phrases that appear in the same sentence with m_s. In our example, we find *Kensal Green Cemetery* and *alexander john ellis* appear in the same sentence in *supporting doc1*, so we extract *Kensal Green Cemetery* as one of reasoning entities. Then, we find another document d_2 which contains any of the reasoning entities. In our example, *supporting doc2* contains the reasoning entity *Kensal Green Cemetery*. Finally, we check whether the reasoning entity appears with one of the candidates in the same sentence. If so, we would add the path to the entity graph. For example, *Kensal Green Cemetery* and *Kensington* appear in the same sentence in *supporting doc2*. Therefore, the path *alexander john ellis* → *Kensal Green Cemetery* →*Kensington* can be added to the entity graph.

Since each entity in different documents has different contexts, so we use mentions of the subject entity, reasoning entities, and candidate answers as nodes in the entity graph. In our example, *Kensal Green Cemetery* appears in two different sentences, so we need to add nodes for different positions Figure 2 shows an example of an entity graph, where m_s, m_c and m_a are mentions of the subject entity s, reasoning entities C_c, and candidate answers C_q respectively.

Then, we define the following types of edges between pairs of nodes to encode various structural information in the entity graph.

1. an edge between a subject node and a reasoning node if they appear in the same sentence in a document, e.g., e_{sc} in Figure 2.
2. an edge between two nodes if they are reasoning nodes and are adjacent nodes on the same path, e.g., e_{cc} in Figure 2.
3. an edge between a reasoning node and a candidate node if they appear in the same sentence in a document, e.g., e_{ca} in Figure 2.
4. an edge between two nodes if they are mentions of the same candidate, e.g., e_{aa} in Figure 2.
5. an edge between two nodes if they appear in the same document.
6. nodes that do not meet previous conditions are connected.

3.3 Reasoning with Gated-RGCN

We first use pretrained word embeddings GLoVe [Pennington *et al.*, 2014] to represent each node in the entity graph, and

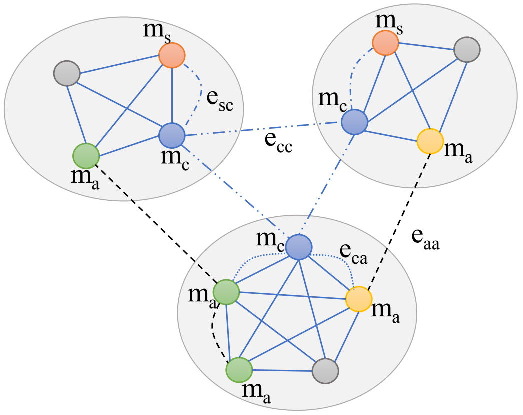


Figure 2: The entity graph for supporting documents (ellipses), where nodes are mentions of the subject entity (m_s), reasoning entities (m_c), and candidates (m_a). Nodes with same color indicate they refer to the same entity. Meanwhile, nodes are connected by six types of edges, which are elaborated in Section 3.2.

then use ELMo [Peters *et al.*, 2018] to model contextual information for each node in different documents. These two vectors are concatenated, and then encoded through 1-layer linear network. Thus, the features for all nodes can be denoted as $f_n \in \mathbb{R}^{T\times d}$, where T is the number of nodes in the graph, and d is the dimension of the node feature.

After graph initialization, we employ a Gated Relational Graph Convolutional Network (Gated-RGCN) to realize multi-hop reasoning. First, we use R-GCN to aggregate messages from its direct neighbors. Specifically, at lth layer, the aggregated message z_i^l for node i can be obtained via

$$z_i^l = \sum_{j\in\mathcal{N}_i} \sum_{r\in R_{ij}} \frac{1}{|\mathcal{N}_i|} W_r^l h_j^l$$

where $\mathcal{N}_i$ is the neighbors of node i, R_{ij} is the set of relations between i and j, $W_r^l \in \mathbb{R}^{d\times d}$ is a relation-specific weight matrix, $|\cdot|$ indicates the size of $\mathcal{N}_i$, and h_j^l is the hidden state of node j at lth layer. Then, the update message u_i^l for node i can be obtained by combining the aggregated message with its original node information:

$$u_i^l = W_0^l h_i^l + z_i^l$$

where $W_0^l \in \mathbb{R}^{d\times d}$ is a general weight.

In Entity-GCN [De Cao *et al.*, 2019], HDE [Tu *et al.*, 2019] and BAG [Cao *et al.*, 2019], a gating mechanism is applied on the update vector u_i^l and the hidden state of node i for updating the hidden state of node i at the next layer.

$$\begin{aligned} w_i^l &= \sigma(f_g([u_i^l; h_i^l])) \quad (1) \\ h_i^{l+1} &= w_i^l \odot \tanh(u_i^l) + (1 - w_i^l) \odot h_i^l \quad (2) \end{aligned}$$

where σ is the sigmoid function, $[u_i^l; h_i^l]$ is the concatenation of u_i^l and h_i^l, f_g is implemented with a single-layer multi-layer perceptron (MLP), $\tanh(\cdot)$ is a non-linear activation function, and $\odot$ denotes element-wise multiplication.

This gating mechanism regulates how much of the update message propagates to the next step, so it can prevent overwriting past information. However, the traditional GCN method only updates the central node based on the aggregated information of adjacent nodes, but there is a lot of irrelevant information during information propagating.

When humans do reasoning problems, they always choose the supports information based on the query information. Inspired by human reasoning processing, we add another question-aware gate to optimize graph convolution procedure, which is suitable for multi-hop reading comprehension. This gating mechanism can regulate the aggregated message according to the question, and introduce the question information into the update message simultaneously.

First, we represent the question q using a bidirectional LSTM (BiLSTM) network [Hochreiter and Schmidhuber, 1997], where GLoVe is used as word embeddings.

$$p = BiLSTM(q), p \in \mathbb{R}^{M\times d}$$

Then, the final question representation can be obtained by a weighted sum of these vectors.

$$\begin{aligned} w_{ij} &= \sigma(W_q^T[u_i^l; p_j] + b_q) \\ \alpha_{ij} &= \frac{\exp(w_{ij})}{\sum_{k=1}^{M} \exp(w_{ik})} \\ q_i^l &= \sum_{j=1}^{M} \alpha_{ij} p_j \end{aligned}$$

Finally, u_i^l could be updated via:

$$\begin{aligned} \beta_i^l &= \sigma(W_s^T[q_i^l; u_i^l] + b_s) \\ u_i^l &= \beta_i^l \odot \tanh(q_i^l) + (1 - \beta_i^l) \odot u_i^l \end{aligned}$$

With this new u_i^l, we use Equation 1 and 2 to obtain h_i^{l+1}, which is the hidden state of node i at the $(l+1)$th layer.

We stack the networks for L layers where all parameters are shared, and finally obtain $h^L = \{h_i^L\}_{i=1}^T$ for the entity graph.

3.4 Output Layer

Similar to BAG [Cao *et al.*, 2019], we apply a bi-directional attention between the entity graph and the question. The similarity matrix $S \in \mathbb{R}^{T\times M}$ is first calculated via

$$S = avg_{-1} f_a([h^L; p; h^L \odot p])$$

where avg_{-1} is the average operation in last dimension, and f_a is a single-layer MLP. Then, the node-to-question attention g_{n2q} and question-to-node attention g_{q2n} are calculated via

$$\begin{aligned} g_{n2q} &= softmax_{col}(S) \cdot p \\ g_{q2n} &= dup(softmax(max_{col}(S)))^T \cdot h^L \end{aligned}$$

where $softmax_{col}$ and max_{col} denote performing softmax and max function across columns respectively, dup is the function to duplicate the result $softmax(max_{col}(S)) \in \mathbb{R}^{1\times M}$ for T times into shape $\mathbb{R}^{T\times M}$.

The output of the bi-directional attention layer is $[h^L; g_{n2q}; h^L \odot g_{n2q}; h^L \odot g_{q2n}]$, which is then fed to a 2-layer fully connected feed-forward network with $\tanh$ as the activation function in each layer. Finally, the softmax function is applied among the output, which can generate the prediction result for each node in the graph. Each candidate may correspond to several nodes, since it may appear in multiple documents. We use the maximal probability of these nodes as the result of the candidate, and use cross-entropy as the loss function.

4 Experiments

4.1 Dataset and Experimental Settings

We use WikiHop [Welbl *et al.*, 2018] to validate the effectiveness of our proposed approach, which is a multi-choice style reading comprehension data set. The dataset contains about 43K/5K/2.5K samples in training, development, and test set respectively. The test set is not public and can only be evaluated online blindly.

In our implementation, we use NLTK [Bird, 2006] to tokenize the supporting documents, question, and candidates into word sequences, and then find mentions of subject entity and candidates in supporting documents through the exact matching strategy. In order to extract reasoning entities, we use Stanford CoreNLP [Manning *et al.*, 2014] to perform entity recognition on the supporting documents.

We use the standard 1024-dimension ELMo and 300-dimension pre-trained GLoVe (trained from 840B Web crawl data) as word embeddings. The dimensions of hidden states in BiLSTM and GCN are set as $d = 256$, and the number of nodes and the query length is truncated as 600 and 25 respectively. We stack $L = 4$ layers of the Gated-RGCN blocks. During training, we set the mini-batch size as 16, and use Adam [Kingma and Ba, 2015] with learning rate 0.0002 for learning the parameters.

4.2 Main Results

We compare our approach with several previously published models, and present our results in Table 2. The performance of multiple choice QA is evaluated by the accuracy of choosing the correct answer. Table 2 shows the performance of approaches both on development and test set respectively. As shown in the table, we can see that our approach achieves the state-of-the-art accuracy on both development and test set against all types of approaches, including GCN-based approaches (Entity-GCN [De Cao *et al.*, 2019], BAG [Cao *et al.*, 2019], HDE [Tu *et al.*, 2019], MHQA-GRN [Song *et al.*, 2018]), memory-based approaches (Coref-GRU [Dhingra *et al.*, 2018], EPAr [Jiang *et al.*, 2019]), path-based approaches (PathNet [Kundu *et al.*, 2019]), and attention-based approaches (CFC [Zhong *et al.*, 2019], DynSAN [Zhuang and Wang, 2019]).

For ensemble models, our approach also achieves state-of-the-art performance, which surpasses the reported human performance [Welbl *et al.*, 2018] by about 4.2%.

Single Models	**Dev**	**Test**
BIDAF [Seo *et al.*, 2017]	49.7	42.9
Coref-GRU [Dhingra *et al.*, 2018]	56.0	59.3
MHQA-GRN [Song *et al.*, 2018]	62.5	65.4
Entity-GCN [De Cao *et al.*, 2019]	64.8	67.6
PathNet† [Kundu *et al.*, 2019]	67.1	-
BAG [Cao *et al.*, 2019]	66.5	69.0
EPAr [Jiang *et al.*, 2019]	67.2	69.1
CFC [Zhong *et al.*, 2019]	66.4	70.6
HDE [Tu *et al.*, 2019]	68.1	70.9
DynSAN [Zhuang and Wang, 2019]	70.1	71.4
Proposed	**70.8**	**72.5**
Ensemble Models		
Entity-GCN [De Cao *et al.*, 2019]	68.5	71.2
DynSAN‡ [Zhuang and Wang, 2019]	-	73.8
HDE‡ [Tu *et al.*, 2019]	-	74.3
Proposed	**74.0**	**78.3**
Human	-	74.1

Table 2: The performance of different models on WikiHop dataset. ”-” indicates missing results, † indicates that the missing results are not reported in their papers, ‡ indicates that the results are not reported in their papers but available on WikiHop leaderboard, and ♯ indicates that the models are unpublished but only available on the leaderboard.

Model	**Dev**	△
Full model	**70.8**	-
(a) w/o reasoning entities	69.1	-1.7
(b) w/o question	70.2	-0.6
(c) w/o attention in question encoding	68.8	-2.0
(d) w/o edge types	69.0	-1.8
(e) reduce edge types	69.5	-1.3

Table 3: Ablation results on the WikiHop dev set.

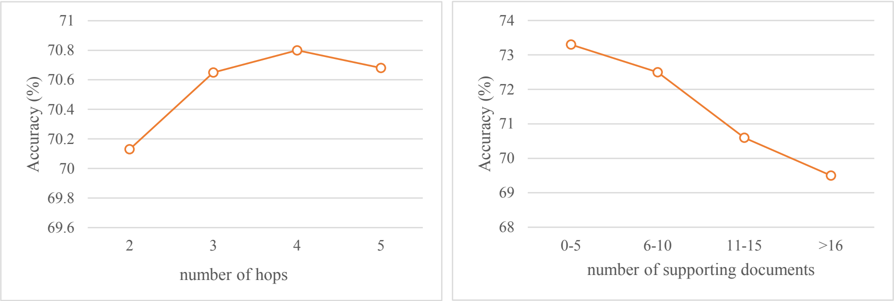
4.3 Ablations Studies

We conduct an ablation study to evaluate the contribution of each model component, and show the results in Table 3.

In (a), we delete all reasoning entities, so our graph degenerates to the entity graph in [De Cao *et al.*, 2019]. The accuracy on the development set drops to 69.1%, but it is still higher than the accuracy of Entity-GCN [De Cao *et al.*, 2019]. This proves the effectiveness of reasoning entities and the Gated-RGCN mechanism. In (b), we do not bring in the question in the reasoning. While in (c), we only use BiLSTM to encode the question, but do not adopt the attention mechanism in BiLSTM. From (b) and (c), we can see that question is useful for reasoning. In (d), we treat all edge types equally, so Gated-RGCN is replaced by Gated-GCN in reasoning, which reduces 1.8% accuracy absolutely. In (e), we only define the edge types as in BAG [Cao *et al.*, 2019], and the accuracy drops to 69.5%. From (d) and (e), we learn that different types of edges is also critical in the reasoning.

4.4 Analysis

In this section, we conduct a series of experiments with different setting in our approach.



(a) number of hops

(b) number of supporting docs

Figure 3: The results of our approach on the WIKIHOP dev dataset with different experimental settings.

Different number of Gated-RGCN Layers. We evaluate our approach with different number of GCN layers (L), and the result is shown in Figure 3(a). From the figure, we can see that the accuracy increases gradually and then drops finally, which proves the effectivity of Gated-RGCN. It is expected that the accuracy drops finally, because more hops would bring in noise. As for why 4-hops is the best, this may be because many samples are 1-hop problems, and the length of path in these samples is exactly 4.

Different number of supporting documents. We split the development set into subsets according to the number of supporting documents, and then evaluate our approach on each subset. The results are shown in Figure 3(b). From the figure, we can see that more documents may bring in more irrelevant information, which is harmful to the multi-hop QA. However, even on a number greater than 16, our model achieved an accuracy of 69.5%, which is better than the overall effect of most models.

Different word embeddings. Table 4 shows the results of our approach when using different word embeddings. From the table, we can see that only using ELMo or GLoVe in our approach will cause a severe drop, and replacing ELMo with BERT [Devlin *et al.*, 2019] delivers a competitive result. Meanwhile, we can see that for the GCN-based approach, the initialization of nodes is extremely critical, because this type of approach has greatly compressed the information when constructing the graph.

Embeddings	**Dev**
ELMo	66.1
GLoVe	64.5
Bert+GLoVe	70.78
ELMo+GLoVe	**70.81**

Table 4: The results with different word embeddings on the WIKIHOP dev set.

5 Conclusion

In this paper, we propose a novel approach for multi-hop reading comprehension across documents. Our approach extends the entity graph by introducing reasoning entities, which can form the reasoning path from question to candidates. In addition, our approach incorporates the question in the multi-hop reasoning through a new gate mechanism to regulate how much useful information propagating from neighbors to the node. Experiments show that our approach achieves state-of-the-art accuracy both for single and ensemble models.

Our future work would focus on the interpretability of the multi-hop reading comprehension across documents. In addition, we would like to build the entity graph with reasoning entities dynamically during the reasoning, and apply our model on other datasets.

Acknowledgments

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900), the Fundamental Research Funds for the Central Universities (No. 2019FZA5013), the Zhejiang Provincial Natural Science Foundation of China (No. LY17F020015), the Chinese Knowledge Center of Engineering Science and Technology (CKCEST) and MOE Engineering Research Center of Digital Library.

References

[Bird, 2006] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[Cao *et al.*, 2019] Yu Cao, Meng Fang, and Dacheng Tao. Bag: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *NAACL-HLT*, 2019.

[Das *et al.*, 2017] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alexander J. Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *ICLR*, 2017.

[De Cao *et al.*, 2019] Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Dhingra *et al.*, 2018] Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Neural models for reasoning over multiple mentions using coreference. In *NAACL-HLT*, 2018.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[Hu *et al.*, 2017] Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. Reinforced mnemonic reader for machine reading comprehension. In *IJCAI*, 2017.

[Jiang *et al.*, 2019] Yichen Jiang, N. Joshi, Yen-Chun Chen, and Mohit Bansal. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *ACL*, 2019.

[Khashabi *et al.*, 2018] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL-HLT*, 2018.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kociský *et al.*, 2017] Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2017.

[Kundu *et al.*, 2019] Souvik Kundu, Tushar Khot, Ashish Sabharwal, and Peter Clark. Exploiting explicit paths for multi-hop reading comprehension. In *ACL (1)*, pages 2737–2747. Association for Computational Linguistics, 2019.

[Lv *et al.*, 2019] Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. *ArXiv*, abs/1908.11513, 2019.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, 2014.

[Mihaylov *et al.*, 2018] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[Qiu *et al.*, 2019] Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. Dynamically fused graph network for multi-hop reasoning. In *ACL*, 2019.

[Seo *et al.*, 2017] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.

[Shen *et al.*, 2017] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM, 2017.

[Song *et al.*, 2018] Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *ArXiv*, abs/1809.02040, 2018.

[Tu *et al.*, 2019] Ming Tu, Guangtao Wang, Jing Huang, Yun Shu Tang, Xiaodong He, and Bowen Zhou. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *ACL*, 2019.

[Welbl *et al.*, 2018] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018.

[Williams, 1992] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018.

[Zhong *et al.*, 2019] Victor Zhong, Caiming Xiong, Nitish Shirish Keskar, and Richard Socher. Coarse-grain fine-grain coattention network for multi-evidence question answering. In *ICLR*, 2019.

[Zhuang and Wang, 2019] Yimeng Zhuang and Huadong Wang. Token-level dynamic self-attention network for multi-passage reading comprehension. In *ACL*, 2019.