Two-Phase Hypergraph Based Reasoning with Dynamic Relations for Multi-Hop KBQA

Jiale Han, **Bo Cheng** and **Xu Wang**
State Key Laboratory of networking and switching technology,
Beijing University of Posts and Telecommunications
{hanjl,chengbo,wxx}@bupt.edu.cn

Abstract

Multi-hop knowledge base question answering (KBQA) aims at finding the answers to a factoid question by reasoning across multiple triples. Note that when human performs multi-hop reasoning, one tends to concentrate on specific relation at different hops and pinpoint a group of entities connected by the relation. Hypergraph convolutional networks (HGCN) can simulate this behavior by leveraging hyperedges to connect more than two nodes more than pairwise connection. However, HGCN is for undirected graphs and does not consider the direction of information transmission. We introduce the directed-HGCN (DHGCN) to adapt to the knowledge graph with directionality. Inspired by human's hop-by-hop reasoning, we propose an interpretable KBQA model based on DHGCN, namely two-phase hypergraph based reasoning with dynamic relations, which explicitly updates relation information and dynamically pays attention to different relations at different hops. Moreover, the model predicts relations hop-by-hop to generate an intermediate relation path. We conduct extensive experiments on two widely used multi-hop KBQA datasets to prove the effectiveness of our model.

1 Introduction

Knowledge base question answering (KBQA) is a challenging task and has attracted many researchers to work on it. Previous work [Xu *et al.*, 2016; Huang *et al.*, 2019] mainly focuses on single-hop QA which can be answered by one fact triple in KB. Although the performance has been improved a lot over the last years, these methods lack the ability to reason across multiple triples when one triple is not enough to answer the question. And this type of questions is defined as multi-hop QA, which requires a reasoning path involving multiple triples to obtain the answers $(e_1 \rightarrow r_1 \rightarrow e_2 \rightarrow ... \rightarrow r_{n-1} \rightarrow e_n)$. As shown in Figure 1, we give a two-hop KBQA example. The reasoning path starts from the entity mentioned in the query and consists of the relations at each hop and the intermediate entities.

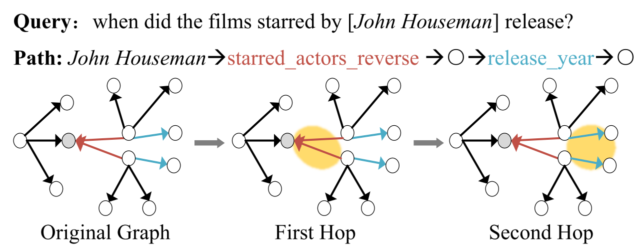


Figure 1: Exemplar of mulit-hop question answering based on knowledge graph. Our proposed model dynamically concentrates on different relations at different hops to guide the model to reason along the right path and find the final answers.

In order to solve the multi-hop KBQA, some work has been proposed recently, which can be roughly divided into two research directions. The first one is based on previous neural network models, such as the studies in [Zhou *et al.*, 2018; Xu *et al.*, 2019]. The other direction is based on graph neural networks [Wu *et al.*, 2019] which develops rapidly in recent years, and has achieved desirable performance [Sun *et al.*, 2018; Sun *et al.*, 2019]

Note that when human performs multi-hop reasoning, one starts from the entity mentioned in the query and walks down to the next hop following specific relations at different hops, and repeats this step to form a reasoning path and find the final answers, as shown in Figure 1. We believe that relation information is vital for multi-hop reasoning. However, previous work does not fully utilize the relation information. The work in Schlichtkrull *et al.* [2018] calculates relation-specific transformation by separating different relations, which does not consider the semantic relation information. And work in Xiong *et al.* [2019] takes relation information to acquire static graph attention without updating relation information during the multi-hop reasoning process.

Morever, the pairwise connection between nodes based on graph convolutional networks (GCN) may be not sufficient to fully represent the knowledge graph's high-order relationship among entities and relations. And some leading work on hypergraph convolutional networks (HGCN) [Feng *et al.*, 2019; Yadati *et al.*, 2019] has been proposed recently. HGCN leverages hyperedges to connect more than two nodes simultaneously, which is conductive to mimicing human reasoning of pinpointing a group of entities connected by same relation, rather than entity-by-entity reasoning. On the downside, HGCN is for undirected hypergraphs, but knowledge graph is directed which each triple has specific direction meaning. Also HGCN learns the relation representation potentially by gathering from the connected entities, but does not reveal and further utilize it.

This paper firstly proposes the Directed Hypergraph Convolutional Networks (DHGCN) for directed hypergraph to take the direction of information transmission into account. Inspired by human's hop-by-hop reasoning behavior, we further propose a novel and interpretable directed hypergraph convolutional network based model for multi-hop KBQA, namely Two-Phase Hypergraph Based Reasoning with Dynamic Relations (2HR-DR), which explicitly updates the relation information and dynamically focuses on specific relation at different hops. Specifically, we separate the DHGCN into two-phase to let the model firstly learns relation representation explicitly by connected entity feature, and allocates weight dynamically for different relations, then updates the entity states based on dynamic relation weights. Moreover, we predict relations hop-by-hop and form a sequential relation path to make the reasoning interpretable.

The main contributions of this paper can be summarized as follows:

- This paper proposes a directed hypergraph convolutional network that incorporates direction information into HGCN to deal with directed knowledge graph.
- We further propose a two-phase hypergraph based reasoning with dynamic relations which explicitly learns and updates relation information and dynamically concentrates on different relations at different hops.
- Our model is interpretable which predicts relations hop-by-hop and form an observable path for reasoning analysis and failure diagnosis.
- We experiment on two widely used multi-hop KBQA datasets and the experimental results demonstrate the effectiveness of our model.

2 Related Work

2.1 Multi-hop Question Answering

Multi-hop KBQA models can be roughly divided in two types. The first one is to apply the previous neural networks. These models adapt the previous single-hop QA methods [Bordes *et al.*, 2014; Hao *et al.*, 2017; Min *et al.*, 2018] to multi-hop QA. Xu *et al.* [2019] improves KVMemNet to achieve better results across multiple triples. Zhong *et al.* [2019] employs a coarse-grain and a fine-grain module. However, these methods lack of considering graph-structure information. The other is based on graph neural networks. Tu *et al.* [2019b] employs GCN to reason over heterogeneous graphs. Sun *et al.* [2019] learns what to retrieve from the KB and corpus and then reasons over the built graph. Xiong *et al.* [2019] applies graph attention networks to achieve better performance. Cao *et al.* [2019] proposes a bi-directional attention entity graph convolutional network. These models either use r-GCN [Schlichtkrull *et al.*, 2018] which does not consider the semantic relation information, or use graph attention networks to assign static weights. Different from these models, this paper proposes a dynamic relation strategy, which dynamically updates relation states during the reasoning process and focuses on different relations at different hops.

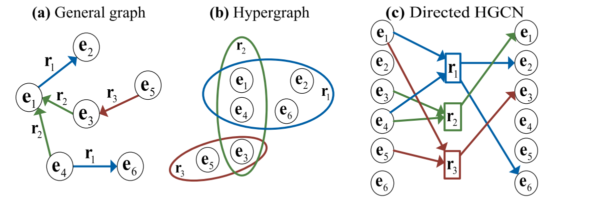


Figure 2: (a) shows a general structure of knowledge graph, (b) is the corresponding hypergraph representation, and (c) is the proposed directed hypergraph convolutional network, which considers the direction of information transmission.

2.2 Explainable Reasoning

Most deep learning models employ end-to-end fashion and lack interpretability. There is some work to try to solve this problem. Zhou *et al.* [2018] proposes a interpretable reasoning network. Chen *et al.* [2019] trains a model to extract a reasoning chain and then puts the chain information to a BERT-based QA methods. Tu *et al.* [2019a] puts forward a RC model which jointly predicts the answer and supporting sentences. Inspired by these work, we put forward an interpretable model based on HGCN to predict relations hop-by-hop under the supervision of reasoning path.

2.3 Hypergraph Convolutional Networks

Feng *et al.* [2019] presents a hypergraph neural networks, which applies hypergraph structure instead of general graph and encodes high order data correlation effectively. Bai *et al.* [2019] further enhances the ability of representation learning by using attention modules. However, existing HGCN methods are for undirected hypergraph and not suitable for the directed KG which will cause the lack of direction information.

3 Directed Hypergraph Convolutional Networks

In this section, we firstly review the hypergraph analysis theory and HGCN. Following, we introduce the proposed directed HGCN based on directed hypergraph.

3.1 Hypergraph Convolutional Networks Revisited

In Figure 2.(a) and (b), we show the form of general graph and the structure of corresponding hypergraph. Mathematically, given a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, which includes a node set $\mathcal{V}$, a hyperedge set $\mathcal{E}$, and a diagonal matrix $\mathbf{W}$ of edge weights. Different from the general graph, the hypergraph defines hyperedge that can connect more than two nodes, and focuses on the connection between nodes and hyperedges indicated by the incidence matrix $\mathbf{H} \in \mathbf{R}^{|\mathcal{V}| \times |\mathcal{E}|}$,

$$H(i,j) = \begin{cases} 0, & v_i \notin e_j \\ 1, & v_i \in e_j \end{cases}$$

$\mathbf{D}_e$ and $\mathbf{D}_v$ represent the diagonal matrices of the edge degrees and the vertex degrees respectively.

$$D_e(j,j) = \sum_{i=0}^{|\mathcal{V}|} H(i,j) \qquad D_v(i,i) = \sum_{j=0}^{|\mathcal{E}|} W(j,j)\, H(i,j)$$

The mathematical equation of hypergraph convolutional networks (HGCN) is as follows [Feng *et al.*, 2019],

$$\mathbf{X}^{(l+1)} = \mathbf{D}_v^{-1}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{X}^{(l)}\mathbf{P}$$

where $\mathbf{X}^{(l)}$ is the signal of hypergraph at l layer, $\mathbf{P}$ is the learnable parameter. The features of nodes are transformed to the connected hyperedges by multiplying $\mathbf{H}^{\mathrm{T}}$, and then aggregated from the hyperedges by multiplying $\mathbf{H}$. However, the form of HGCN is for undirected hypergraph and do not consider direction information, which may lead to noise and redundancy.

3.2 Directed Hypergraph Convolutional Networks

Inspired by the pagerank algorithm for directed hypergraph [Tran *et al.*, 2019], we propose the directed hypergraph convolutional networks. Each hyperedge $e \in \mathcal{E}$ in directed hypergraph is written as $e = (e^{tail}, e^{head})$, where e^{tail} is called the tail of the hyperedge e and e^{head} is called the head of the hyperedge e. The directed hypergraph can be denoted by two incidence matrices $\mathbf{H}^{tail}$ and $\mathbf{H}^{head}$.

$$H^{tail}(i,j) = \begin{cases} 0, & v_i \notin e_j^{tail} \\ 1, & v_i \in e_j^{tail} \end{cases}$$

$$H^{head}(i,j) = \begin{cases} 0, & v_i \notin e_j^{head} \\ 1, & v_i \in e_j^{head} \end{cases}$$

Let $\mathbf{D}_e^{head}$ and $\mathbf{D}_v^{tail}$ be the diagonal matrices of head degrees of hyperedges and tail degrees of nodes,

$$D_e^{head}(j,j) = \sum_{i=0}^{|\mathcal{V}|} H^{head}(i,j)$$

$$D_v^{tail}(i,i) = \sum_{j=0}^{|\mathcal{E}|} W(j,j)\, H^{tail}(i,j)$$

Our proposed directed hypergraph convolutional network can be formulated by

$$\mathbf{X}^{(l+1)} = \mathbf{D}_v^{tail^{-1}}\mathbf{H}^{tail}\mathbf{W}\mathbf{D}_e^{head^{-1}}\mathbf{H}^{head^{\mathrm{T}}}\mathbf{X}^{(l)}\mathbf{P}$$

As shown in Figure 2.(c), DHGCN first forms hyperedges feature by aggregating information from nodes that point to hyperedges, and then updates the node states from the hyperedge information that points to nodes.

4 Model

4.1 Task Definition

A knowledge graph is denoted as $\mathcal{K} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ is the set of entities and $\mathcal{E}$ is the set of relations in KB. $\mathcal{T}$ consists of a set of triples (e_h, r, e_t) which represent the relation $r \in \mathcal{E}$ holds between $e_h \in \mathcal{V}$ and $e_t \in \mathcal{V}$. Given a natural language question $q = (w_1, w_2, ..., w_{|q|})$, where w_i denotes a word in the question, the task aims to pick the answers from $\mathcal{V}$. The overview of the entire model is shown in Figure 3.

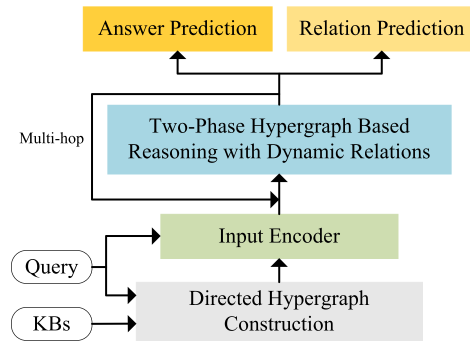


Figure 3: Overview of the model.

4.2 Directed Hypergraph Retrieval and Construction

To retrieve the hypergraph for a L-hop question, we firstly obtain seed entities from the question by entity linking and then traverse the KG to get the entities set $e = (e_1, e_2, ..., e_n)$ and relations set $r = (r_1, r_2, ...r_{m-1})$ within L hops, where n and $m-1$ denote the number of entities and relations. Here we add <STOP> to the relation set as the end mark of relation prediction to let the model learn when to stop reasoning.

Based on these, we figure out two incidence matrices $\mathbf{H}^{tail} \in \mathbf{R}^{n\times m}$ and $\mathbf{H}^{head} \in \mathbf{R}^{n\times m}$. For each triple (e_{i1}, r_j, e_{i2}), we put

$$H^{head}(i1,j) = 1, \qquad H^{tail}(i2,j) = 1.$$

4.3 Input Encoder

The input encoder encodes a natural language question and all the candidate entities and relations in hypergraph to vector representation, and applies co-attention [Zhong *et al.*, 2019] to learn the correlation between question and entities to further enhance the query-aware entity representation.

Let $\mathbf{X}_q \in \mathbf{R}^{|q|\times d}$, $\mathbf{X}_e \in \mathbf{R}^{n\times d}$, and $\mathbf{X}_r \in \mathbf{R}^{m\times d}$ be the embedding matrices of question, entities, and relations, where d denotes the embedding dimension. We firstly apply a bi-directional long short-term memory network (bi-LSTM) [Hochreiter and Schmidhuber, 1997] to encode question and obtain question hidden states $\mathbf{H}_q = \mathrm{biLSTM}(\mathbf{X}_q) \in \mathbf{R}^{|q|\times h}$. h is the hidden dimension of bi-LSTM and here we assume $h = d$.

Following the work of [Zhong *et al.*, 2019], we employ co-attention to learn query-aware entity representation,

$$\mathbf{A}_{eq} = \mathbf{X}_e(\mathbf{H}_q^{\mathrm{T}}) \in \mathbf{R}^{n\times|q|}$$

$$\mathbf{C}_e = softmax(\mathbf{A}_{eq})\mathbf{H}_q \in \mathbf{R}^{n\times h}$$

$$\mathbf{C}_q = softmax(\mathbf{A}_{eq}^{\mathrm{T}})\mathbf{X}_e \in \mathbf{R}^{|q|\times d}$$

$$\mathbf{D}_e = \mathrm{biLSTM}(softmax(\mathbf{A}_{eq})\mathbf{C}_q) \in \mathbf{R}^{n\times h}$$

$$\mathbf{E}_{co\_attn} = f_c([\mathbf{C}_e; \mathbf{D}_e]) \in \mathbf{R}^{n\times h}$$

where T denotes matrix transposition, $softmax()$ represents column-wise normalization, $[;]$ is column-wise concatenation, and f_c is a linear network which converts $2h$ dimension to h dimension.

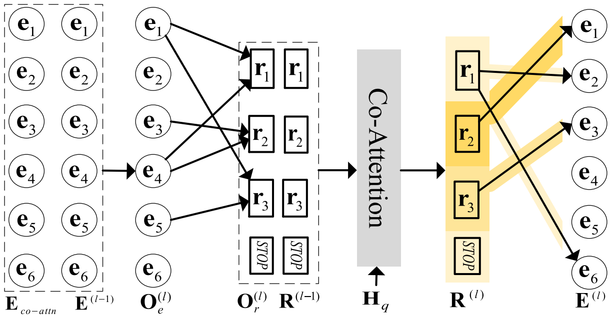


Figure 4: Two-phase hypergraph based reasoning with dynamic relations.

4.4 Reasoning over Hypergraph

This work proposes an interpretable two-phase hypergraph based reasoning with dynamic relations. We separate the hypergraph convolutional networks into two steps to directly learn the relation representation and further update the entity states. Specifically, the feature of nodes are firstly gathered to the connected hyperedges to learn the representation of the relations explicitly. Then the model dynamically allocates the weight of relations based on the similarity between question and relation representation and predicts the current relation. Finally, we update the node states through the connected relation information. Figure 4 depicts the structure of our proposed two-phase hypergraph based reasoning with dynamic relations.

Learn Relation Representation Explicitly

Formally, at the current hop l, let $\mathbf{O}_e^{(l)}$ be the input entities representation,

$$\mathbf{O}_e^{(l)} = f_{in}([\mathbf{E}^{(l-1)}; \mathbf{E}_{co\_attn}]) \in \mathbf{R}^{n \times h}$$

where $\mathbf{E}^{(l-1)}$ is the node status obtained by the previous $l-1$ layer, $\mathbf{E}^{(0)} = \mathbf{X}_e$, and f_{in} is a linear network which converts $2h$ dimension to h dimension. The model firstly explicitly learn the relation representation $\mathbf{O}_r^{(l)}$ by aggregating the connected entity feature,

$$\mathbf{O}_r^{(l)} = \mathbf{D}_e^{head^{-1}} \mathbf{H}^{head^{\mathrm{T}}} \mathbf{O}_e^{(l)} \mathbf{P}_r \in \mathbf{R}^{m \times h}$$

where $\mathbf{P}_r \in \mathbf{R}^{h \times h}$ is the relation specific learnable parameter. Then we concatenate $\mathbf{R}^{(l-1)}$ and $\mathbf{O}_r^{(l)}$ and pass through a linear network f_r to get representation of relations at hop l.

$$\mathbf{R}^{(l)} = f_r([\mathbf{R}^{(l-1)}; \mathbf{O}_r^{(l)}]) \in \mathbf{R}^{m \times h}$$

$$\mathbf{R}^{(0)} = \mathbf{X}_r$$

Allocate Relation Weights Dynamically

Co-attention is employed to learn the correlation between question and relations as the same way before to get the query-aware relation representation $\mathbf{R}_{co\_attn}$. Then we apply max pooling to get weights for each relation,

$$\mathbf{a}_r^l = softmax(MaxPooling(\mathbf{R}_{co\_attn})) \in \mathbf{R}^m$$

and the diagonal matrix $\mathbf{W}$ of edge weights is

$$\mathbf{W} = diag(\mathbf{a}_r^l) \in \mathbf{R}^{m \times m}$$

Relation weights are dynamically allocated which depend on the relation representation updated hop-by-hop. The model predicts the current relation based on relation weights.

Update Entity Adaptively

Based on dynamic relation weights, the model adaptively updates entity states by accumulating connected relation feature,

$$\mathbf{E}^{(l)} = \mathbf{D}_v^{tail^{-1}} \mathbf{H}^{tail} \mathbf{W} \mathbf{R}^{(l)} \mathbf{P}_e \in \mathbf{R}^{n \times h}$$

where $\mathbf{P}_e \in \mathbf{R}^{h \times h}$ is the entity specific learnable parameter.

4.5 Answer Prediction

For L hop question, we sum the entity representation of each layer to get the final representation and pass through a liner layer f_{out} to predict the answer distribution.

$$\mathbf{P} = \sigma(f_{out}(\sum_{l=1}^{L} \mathbf{E}^{(l)}))$$

where σ is the sigmoid function, f_{out} converts h dimension to 1 dimension.

4.6 Loss

The training loss consists of two terms, one is binary cross-entropy loss of the final answers prediction, the other is the negative log likelihood of the intermediate prediction of relations. Formally,

$$\mathcal{L} = \mathcal{L}_e + \lambda \times \sum_{l=1}^{L+1} \mathcal{L}_r(l)$$

$$\mathcal{L}_e = -\sum_{i=1}^{n} [y_i log(p_i) + (1 - y_i) log(1 - p_i)]$$

$$\mathcal{L}_r(l) = -log(\mathbf{a}_r^l(r_l^*))$$

where y is the golden distribution over entities. r_l^* is the golden relation index at hop l. And λ is a hyper parameter to balance the two terms. Since we add <STOP> mark in, the predicted relation path number is $L+1$.

5 Experiment

5.1 Datasets

PQL [Zhou *et al.*, 2018] is a single-answer KBQA dataset, which includes a knowledge base with 5035 entities and 364 relations, and two types of questions. Specifically, PQL-2H consists of 1594 two-hop questions and PQL-3H consists of 1031 three-hop questions. The questions can be answered by walking down a reasoning path consisting of a few relations and the intermediate entities. And the paths are already given. We list two exemplars in dev datasets in Table 1.

MetaQA [Zhang *et al.*, 2018] is a large-scale multi-answer KBQA dataset, with 116045 1-hop questions and 148724 2-hop questions. The knowledge base contains 40128 entities and 9 relations. Since the dataset do not give the reasoning path, we apply a heuristic method to traverse through the graph to find the path which starts from the seed entities in question and the length is exactly equal to $2L+1$. If the answer in it, the path is considered as golden reasoning path. We skip the cases which can not find the path.

Datasets	Question	Answer	Path
PQL-2H	what is the [Habanera] 's artist 's album ?	Tirami_Su	Habanera#_music_recording_artist#Al_Di_Meola #_music_artist_album#Tirami_Su
PQL-3H	what is the artist of release of [Free] 's recordings ?	Ultra_Naté	Free#_music_composition_recordings#Free#_music_release _track_release#Free#_music_recording_artist#Ultra_Naté

Table 1: Exemplars of data details for PQL-2H and PQL-3H datasets.

Model	PQL-2H	PQL-3H
KVMemNet	0.690	0.617
IRN	0.725	0.710
GraftNet	0.707	0.910
SGReader	0.719	0.893
2HR-DR (ours)	**0.755**	**0.921**

Table 2: Experimental results of Hits@1 on PQL dataset.

Model	MetaQA 1-Hop		MetaQA 2-Hop	
	Hits@1	F1	Hits@1	F1
KVMemNet	0.958	-	0.251	-
VRN	0.975	-	0.898	-
GraftNet	0.970	0.910	**0.948**	0.727
SGReader	0.967	0.960	0.807	0.798
2HR-DR (ours)	**0.988**	**0.973**	0.937	**0.814**

Table 3: Experimental results of Hits@1 and F1 on metaQA dataset.

Model	PQL-2H	PQL-3H
r-GCN	0.678	0.859
HGCN	0.693	0.872
Directed-HGCN	0.700	0.883
2HR-DR (ours)	0.755	0.921
- sequential relation prediction	0.748	0.918
- relation prediction	0.734	0.914
- dynamic relations	0.725	0.894
- relation weights	0.712	0.887

Table 4: Experimental results on PQL dataset of ablation study and some variant models.

5.2 Baselines

We compare our model with the following baselines:

KVMemNet [Miller *et al.*, 2016] is an end-to-end memory network which divides the memory into two parts, the key memory stores the head entity and relation, and the value memory stores the tail entity.

IRN [Zhou *et al.*, 2018] is an interpretable reasoning network that employs a hop-by-hop reasoning process for question answering based on knowledge graph.

VRN [Zhang *et al.*, 2018] proposes an end-to-end variational learning algorithm that can effectively solve the multi-hop reasoning while deal with the noise in the question.

GraftNet [Sun *et al.*, 2018] introduces text information to construct graph together with entities, and applies GCN to reasoning.

SGReader [Xiong *et al.*, 2019] also combines the unstructured text and knowledge graph to figure out the incompleteness of knowledge graph. The model employs graph attention to reason effectively.

For GraftNet and SGreader, we use full KB setting and no documents here.

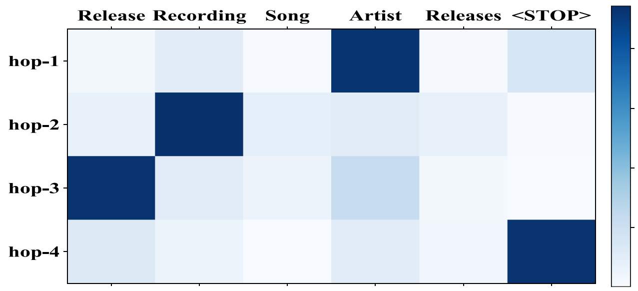
5.3 Implementation Details

Throughout our experiments, we apply 768 dimensional BERT embedding [Devlin *et al.*, 2019] for query words and relations, and 400 dimensional TransE embedding [Bordes *et al.*, 2013] for entities. The hidden size of LSTM and directed hypergraph convolutional networks are all set to 400. During training, the Adam optimizer [Kingma and Ba, 2015] is employed to minimize the loss with a learning rate of $1e^{-4}$. λ and dropout are set to 1 and 0.4. We set original relation path with end mark <STOP> as golden relation path. The layer of reasoning is set to $L+1$. For testing, the hop of question is unknown, the model predicts relation hop-by-hop and stops when the current relation meets end mask.

5.4 Results

Main Results and Analysis

We employ Hits@1 and F1 to measure the performance of the models. For PQL dataset, there is one answer for each question, so we only adopt Hits@1 for evaluation. As shown in Table 2, our method achieves best Hits@1 compared with the baseline models on PQL dataset. Specifically, 2HR-DR achieves a great improvement on PQL-2H, which is 3% higher than the second best model. It also obtains good result on PQL-3H, 1.1% higher than the second best one. Table 3 demonstrates the performance of the baseline methods and our model on the metaQA dataset. Our model outperforms all the baselines on metaQA1-Hop, improving Hits@1 and F1 by 1.8% and 1.3% respectively. And for metaQA2-Hop, we obtain competitive Hits@1 and improve F1 from 79.8% to 81.4%. The reasons why our method performs well include: 1) The model reconstructs the knowledge graph to hypergraph structure and applies directed hypergraph convolutional networks to reason over it, which fully consider the high-order data correlation. 2) We dynamically concentrate on relation information at different hops to guide the model to follow the golden relation path and select the final answers. 3) The information of intermediate reasoning path is introduced



Query: what is the release of recording of amsterdam's artist?
Answer: Krush
Path: Amsterdam→Artist→Another_day→Recording →Another_day→Release→Krush→<STOP>
KBs: [Amsterdam, Artist, Another_Day]
[Amsterdam, Release, Red_Morning]
[Amsterdam, Song, Amsterdam]
[Another_Day, Recording, Another_Day]
[Another_Day, Releases, The_Muffs]
[Another_Day, Release, Krush]

Figure 5: Case study of the predicted relations at each hop.

Datasets		AvgRelation	Relation EM
PQL	2H	3.61	0.998
	3H	5.61	0.973
MetaQA	1-Hop	5.43	0.927
	2-Hop	8.37	0.872

Table 5: Experimental results of Relation EM for the relation prediction of the model. AvgRelation represnets the average number of relations in the subgraph of each question.

into our model to supervise the model focus on the dynamic relations at different hops.

To illustrate the model's ability to predict relational path, we also define the relational EM. If the output relation path and the golden relation path are completely identical, this case is counted as an exact match, and the relation EM is set to 1. Otherwise, the relation EM is 0. We average all cases to obtain the final value. AvgRelation denotes the average number of relations in the subgraph of each question. The experimental results are shown in Table 5 which achieve a promising performance on sequential relation prediction even we adopt a strict metric.

Ablation Study on Model Components

We compare our model with a few variants. As shown in the first three rows of Table 4, the first model r-GCN only applies r-GCN [Schlichtkrull *et al.*, 2018] to reason over knowledge graph, where adding skip connection and gate mechanism. The second model HGCN and the third Directed-HGCN apply HGCN and DHGN with the same setting. HGCN outperforms the r-GCN which shows the superiority of HGCN to represent high-order relation among data. Moreover, HGCN requires less parameters than r-GCN. Directed-HGCN achieves higher performance than HGCN, illustrating the effectiveness of our proposed DHGCN in reducing the effects of noise by direction information.

To evaluate the performance of different components in our model, we also perform ablation study. 2HR-DR w/o sequential relation prediction predicts all relations by the final relations states which do not consider the sequential order. 2HR-DR w/o relation prediction does not consider the relation loss part. 2HR-DR w/o dynamic relations assigns static relation weights according to the similarity between relations and query. And 2HR-DR w/o relation weights represents our proposed two phase reasoning network without relation weights. The above two methods do not consider relation prediction. The experimental results prove the validity of each component. Shown in Table 4, compared 2HR-DR w/o relation weights with 2HR-DR w/o dynamic relations, we can see relation information is vital for reasoning and our proposed dynamic relation significantly improves performance by focusing different relations at different hops. By comparing 2HR-DR w/o relation prediction and 2HR-DR w/o sequential relation prediction, the results illustrate the significance of relation path to guide the model to reason hop-by-hop.

Case Study on Interpretibility

As Figure 5 shows, we give an exemplar question from PQL-3H and its corresponding reasoning path and triples in KB, and plot the attention heatmap of relations' probability distribution at each hop to illustrate dynamic relations of our model. It is clear that 2HR-DR has the ability to predict relations hop-by-hop and stop reasoning automatically. For question "what is the release of recording of amsterdam's artist?", the model firstly detects relation "Artist", then "Recording" and "Release" successively, finally meets <STOP> to end the reasoning process. From the visualization of heatmap, we can observe our model's predicted relation path (Artist → Recording → Release → <STOP>).

6 Conclusion

This paper proposes a QA model based on hypergraph convolutional networks. We expand the HGCN to directed hypergraph and propose the directed-HGCN. In order to imitate humans' hop-by-hop reasoning, we further propose the two-phase hypergraph based reasoning with dynamic relations to firstly update the relation states and assign dynamic weights for relations, then learn the node representation adaptively. Meantime, we predict the relation at each hop and form a sequential relation path to make the model interpretable. Experimental results prove the effectiveness of the model.

Acknowledgements

This work is supported by the National Key Research and Development Program of China under grant 2018YFB1003804, in part by the National Natural Science Foundation of China under grant 61921003, 61972043. Bo Cheng is the corresponding author.

References

[Bai *et al.*, 2019] Song Bai, Feihu Zhang, and Philip H. S. Torr. Hypergraph convolution and hypergraph attention. *CoRR*, abs/1901.08150, 2019.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

[Bordes *et al.*, 2014] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *ECML-PKDD*, pages 165–180, 2014.

[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*, pages 357–362, 2019.

[Chen *et al.*, 2019] Jifan Chen, Shih-ting Lin, and Greg Durrett. Multi-hop question answering via reasoning chains. *CoRR*, abs/1910.02610, 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*, pages 4171–4186, 2019.

[Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, pages 3558–3565, 2019.

[Hao *et al.*, 2017] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *ACL*, pages 221–231, 2017.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2019] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge graph embedding based question answering. In *WSDM*, pages 105–113, 2019.

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

[Miller *et al.*, 2016] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, pages 1400–1409, 2016.

[Min *et al.*, 2018] Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. Efficient and robust question answering from minimal context over documents. In *ACL*, pages 1725–1735, 2018.

[Schlichtkrull *et al.*, 2018] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, pages 593–607, 2018.

[Sun *et al.*, 2018] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. Open domain question answering using early fusion of knowledge bases and text. In *EMNLP*, pages 4231–4242, 2018.

[Sun *et al.*, 2019] Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *EMNLP-IJCNLP*, pages 2380–2390, 2019.

[Tran *et al.*, 2019] Loc Tran, Tho Quan, and An Mai. Pagerank algorithm for directed hypergraph. *CoRR*, abs/1909.01132, 2019.

[Tu *et al.*, 2019a] Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. *CoRR*, abs/1911.00484, 2019.

[Tu *et al.*, 2019b] Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *ACL*, pages 2704–2713, 2019.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019.

[Xiong *et al.*, 2019] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. Improving question answering over incomplete kbs with knowledge-aware reader. In *ACL*, pages 4258–4264, 2019.

[Xu *et al.*, 2016] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. Question answering on freebase via relation extraction and textual evidence. In *ACL*, 2016.

[Xu *et al.*, 2019] Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. Enhancing key-value memory neural networks for knowledge based question answering. In *NAACL-HLT*, pages 2937–2947, 2019.

[Yadati *et al.*, 2019] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In *NIPS*, pages 1509–1520, 2019.

[Zhang *et al.*, 2018] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *AAAI*, pages 6069–6076, 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.

[Zhou *et al.*, 2018] Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. An interpretable reasoning network for multi-relation question answering. In *COLING*, pages 2010–2022, 2018.