IEICE TRANS. INF. & SYST., VOL.E105–D, NO.4 APRIL 2022

PAPER

MKGN: A Multi-Dimensional Knowledge Enhanced Graph Network for Multi-Hop Question and Answering

Ying ZHANG[†], **Fandong MENG**[††], **Jinchao ZHANG**[††], ***Nonmembers***, **Yufeng CHEN**[†a)], ***Member***, **Jinan XU**[†], ***and*** **Jie ZHOU**[††], ***Nonmembers***

SUMMARY Machine reading comprehension with multi-hop reasoning always suffers from reasoning path breaking due to the lack of world knowledge, which always results in wrong answer detection. In this paper, we analyze what knowledge the previous work lacks, e.g., dependency relations and commonsense. Based on our analysis, we propose a **M**ulti-dimensional **K**nowledge enhanced **G**raph **N**etwork, named MKGN, which exploits specific knowledge to repair the knowledge gap in reasoning process. Specifically, our approach incorporates not only entities and dependency relations through various graph neural networks, but also commonsense knowledge by a bidirectional attention mechanism, which aims to enhance representations of both question and contexts. Besides, to make the most of multi-dimensional knowledge, we investigate two kinds of fusion architectures, i.e., in the *sequential* and *parallel* manner. Experimental results on HotpotQA dataset demonstrate the effectiveness of our approach and verify that using multi-dimensional knowledge, especially dependency relations and commonsense, can indeed improve the reasoning process and contribute to correct answer detection.

key words: *machine reading comprehension, multi-hop reasoning, multi-dimensional knowledge enhancement, graph neural networks*

1. Introduction

Machine reading comprehension (MRC) has recently prevailed in natural language processing. It is the task of answering natural language questions given a set of contexts to evaluate the capability of systems on language understanding and reasoning. With the prevalence of deep neural network, recently proposed models have outperformed human on SQuAD 2.0 [1]. However, most of them focus on answering the questions with a single context, which cannot model multi-hop reasoning on questions with several contexts. Therefore, it is still challenging for existing methods to conduct multi-hop reasoning between questions and multiple contexts. As shown in Fig. 1, to answer question "According to the 2001 census, what was the population of the city in which Kirton End is located?", the correct reasoning path is "in which city Kirton End is located? -> the population of city at the 2001 census?". At step-I, we firstly need to detect the location entity "Kirton End" in contexts to find related supporting fact "Kirton End is a hamlet in the civil parish of Kirton in the Boston. . . ", and then analyze the dependency relations between "Kirton End" and "Boston". At the second step, we detect supporting fact "Boston is a town. . . It is the largest town of Borough. . . " with entity "Boston". The next supporting fact "the town itself had a population of 35,124 at the 2001 census" is found based on dependency relations of "town".

Paragraph I: Kirton End

Kirton End is a hamlet in the civil parish of Kirton in the Boston district of Lincolnshire, England.

It lies on the B1391 road, 4 mi south-west from Boston, and 1.5 mi north-ease from Kirton.

...

Paragraph IV: Boston

Boston is a town and small port in Lincolnshire, on the east coast of England.

It is the largest town of the wider Borough of Boston local government district.

The borough had a total population of 66,900 at the ONS mid 2015 estimates, while the town itself had a population of 35,124 at the 2001 census.

...

Question: According to the 2001 census, what was the population of the city in which Kirton End is located?

Answer: 35,124

Fig. 1 An example in the HotpotQA dataset. Words in orange color represent commonsense knowledge, and words in blue, green, purple and brown represent various entities and their mentions.

Apart from dependency relations, it is essential to exploit commonsense knowledge to find the correct answer. In question, the word "city" is mentioned but does not appear in context, while we only find "Boston is the largest town" in the context. If the commonsense knowledge "Boston is also a city" is available, it is easy for us to find the right answer "35,124".

Above observation and analyses illustrate that not only entities but also dependency relations and commonsense have an significant impact on reasoning process on questions and contexts. It is necessary to utilize multi-dimensional knowledge to enhance representations and interactions between questions and contexts.

Manuscript received July 15, 2021.
Manuscript revised November 21, 2021.
Manuscript publicized December 29, 2021.

[†]The authors are with Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, P. R. China.
[††]The authors are with Pattern Recognition Center, WeChat AI, Tecent Inc, Beijing, P.R. China.
a) E-mail: chenyf@bjtu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2021EDP7154

Additionally, previous work on multi-hop reasoning can be categorized into three classes: a) Advances in evidence extraction [2], [3], which concentrate on extracting as accurate evidences as possible by iteratively utilizing an question- or answer-based selector. These methods only consider available information in dataset and ignore the utilization of external knowledge, which is necessary for reasoning. b) Advances in representation enhancement [4]–[8], which focus on enhancing the representations of questions and contexts and performing implicit multi-hop reasoning with external knowledge and graph neural networks (GNNs) [9]. Although this work introduces external knowledge for representation enhancing, but only limited external knowledge is considered, i.e., named entities. c) Advances in reasoning interpretation [10]–[13], which explicitly model the reasoning process through defining various reasoning modules or decomposing multi-hop questions. These approaches maybe suffer from error propagation due to the step-by-step reasoning process. Different from all these researches, our work focus on using multi-dimensional knowledge to enhance representations and interactions between questions and contexts, especially two specific external knowledge, i.e., dependency relations and commonsense. Besides, we design an end-to-end multi-hop question and answering framework to avoid error propagation and which also can be combined with other methods on evidence extraction for further improvements.

In this paper, we propose a novel model called **M**ulti-dimensional **K**nowledge enhanced **G**raph **N**etwork (MKGN) to fully utilize different dimensional knowledge for question and contexts, i.e., named entities, dependency relations and commonsense. Different from previous work, we pay attention to not only entity knowledge, but also dependency relations of both questions and contexts, and commonsense knowledge in the real world. Specifically, we incorporate the aforementioned knowledge into multi-hop QA models through various GNNs and bidirectional attention mechanism [14] to enhance representations and interactions between questions and contexts. To explore the effects of using multi-dimensional knowledge in different orders, we design both *sequential* and *parallel* architectures for knowledge incorporation. Experimental results on the HotpotQA (distractor) test set have verified the effectiveness of multi-dimensional knowledge enhancement. The contributions of our work can be summarized as follows.

- We explore the knowledge gap existing in multi-hop reasoning process of MRC task and observe that lacking dependency relations and commonsense knowledge can cause reasoning path breaking.
- To narrow the knowledge gap in the reasoning process, we propose two different architectures to incorporate multi-dimensional knowledge, i.e., dependency relations, commonsense knowledge and entity mentions, through graph network, named **M**ulti-dimensional **K**nowledge enhanced **G**raph **N**etwork (MKGN) for the multi-hop QA.
- Experimental results illustrate that our approach yields significant improvements over the baseline on most evaluation metrics and demonstrate the effectiveness of multi-dimensional knowledge in improving multi-hop reasoning process.

2. Problem Investigation

2.1 Task Definition

Given a question and several contexts with scattered evidences, the system not only needs to detect the accurate answer span for complex, multi-hop questions, but also to collect corresponding evidences. The HotpotQA dataset is a typical multi-hop QA dataset. For each question, there are ten corresponding contexts with two gold and each contexts contains multiple sentences. Our goal is to detect the accurate answer spans on them and collect corresponding sentences as supporting evidences.

2.2 Error Analysis on Previous Work

We explore how the knowledge gap affects multi-hop reasoning process in MRC task based on the Dynamic Fusing Graph Network (DFGN) [4], which is an simple but strong baseline on HotpotQA dataset.

We conduct error analysis on predictions of DFGN on the development set of HotpotQA, which contains 7405 examples and 2047 examples are wrongly predicted. We sample 100 examples whose answer is predicted incorrectly and analyze their error types. Analysis results show that 50% of errors are caused by inability to find the correct dependency relations in sentences. For example, to answer questions “The arena where the Lewiston Maineiacs played their home games can seat how many people?”, the right answer is “3,677 seated” but DFGN gives the wrong one “1,400”. The specific analyses are as follows. The correct reasoning process are based on dependency relation “Lewiston Maineiacs played home games -> which arena -> seat how many people”. At step-I, according to the first supporting fact “The team played its home games at the Androscoggin Bank Colise” where “the team” refers to “Lewiston Maineiacs”, we infer that “the arena” is “Androscoggin Bank Colise’. And the second step aims to find “how many people Androscoggin Bank Colise can seat?”. Based on the second supporting fact “The Androscoggion Bank is a 4,000 capacity (3,677 seated) multi-purpose arena”, we obtain the correct answer is “3,677 seated”. However, DFGN wrongly find the second supporting fact that is “The main rink can seat up to 1,400 people and is the home to Niagara Purple Eagles men’s ice hockey team…”, since DFGN mistakenly supposes that “the main rink” refers to “Androscoggin Bank Colise” but ignore it is the home to “Niagara Purple Eagles men’s ice hockey team” rather than “Lewiston Maineiacs”. Therefore, it is critical to conduct co-reference resolution accurately and find the right dependency relations in sentences.

Another 15% are due to lack of commonsense knowledge. For example, to answer question "Are Random House Tower and 888 7th Avenue both used for real estate?", DFGN gives the wrong answer "Yes" based on two supporting facts: 1) "888 7th Avenue is a 628 ft(191m) tall modern-style office skyscraper in Midtown…" 2) "The Random House Tower, that is used as the headquarters of book publisher Random House and….", since "real estate" does not appear in contexts directly and the commonsense knowledge "book publisher does not belong to real estate" is missing. We also notice that some evidences contain similar concepts as confusing information and distract DFGN to detect the correct answer, which contributes to 20% of the errors. Besides, some spans and their sub-spans are both answers, but only one of them is annotated with gold labels. DFGN sometimes predicts the answer only covering part of the correct answer and 10% of errors are caused by the wrong span boundaries.

3. Our Approach

3.1 Overall Framework

As shown in Fig. 2, the overall framework contains five components: a context selector, a knowledge extractor, a question and context encoder, a knowledge enhancer and a predictor. We introduce other four parts in detail apart from the knowledge enhancer which we elaborate in Sect. 3.2. Its inputs are the contexts **C** and the question **Q**. These five components are illustrated as follows.

Context Selector adopts the same selector as DFGN, which applies a classifier based on a pre-trained language model, i.e., BERT [15] followed by a Sigmoid [16] activation layer to select related contexts for question Q, and takes each context and question pair as inputs and outputs the score for selection. To ensure the high recall of relevant contexts, the threshold of selected prediction scores is set to 0.1. After obtaining a set of selected contexts, we concatenate all of them as a long context C_s and then pass it to question answering model.

Knowledge Extractor extracts various knowledge from the contexts and question and formulates them as input for knowledge enhancer. Besides doing Named Entity Recognition (NER) on $\mathbf{C}_s$ through BERT, we use Stanford CoreNLP[†] tools to acquire dependency parsing information of $\mathbf{C}_s$ and **Q**.

To acquire related commonsense knowledge, we following [17] extract all related commonsense paths (e.g., "⟨ museum, UsedFor,art ⟩, ⟨ museum, UsedFor, developing cultural values ⟩, ⟨ museum, UsedFor, education ⟩,…" for concept "museum") for all concepts contained in each $\mathbf{C}_s$ and **Q** pair. For one concept in the context (e.g., museum), we extract all the ConceptNet triples (e.g., ⟨ $\mathrm{head}_concept, relationship, tail - concept$ ⟩) containing the same head concept "Museum". For extracted triples, we first combine those with the same relationship through template "⟨ head-concept, relationship, tail-concept-1, tail-concept-2,…⟩" and convert a relation to a text form based on relation templates, which are partially shown as Fig. 1. Then we concatenate these texts to generate the final commonsense knowledge context for concept "museum".

Question and Context Encoder encodes $\mathbf{C}_s$ and **Q** with the BERT encoder. Then representations of $\mathbf{C}_s$ and **Q** are passed through a bidirectional attention layer [14], short as *bi-attention*. The outputs are $\mathbf{C}_1 \in R^{l\times d_2}$ and $\mathbf{Q}_1 \in R^{M\times d_2}$, where l is the length of $\mathbf{C}_s$, M is the length of **Q** and d_2 is the dimension of hidden unit.

Knowledge Enhancer enhances representations of questions and contexts with each kind of knowledge generated by the knowledge extractor. It aims to capture the relations between entities, dependency relations in sentences and introduce commonsense knowledge to mitigate the gap between question and contexts.

Predictor adopts a cascade structure with four isomorphic long short-term memory (LSTM) [18] networks F_i stacked layer by layerto calculate four output dimensions of the predictor, i.e., supporting sentences $\mathbf{O}_{sup}$, the start postion of the answer $\mathbf{O}_{start}$, the end position of the answer

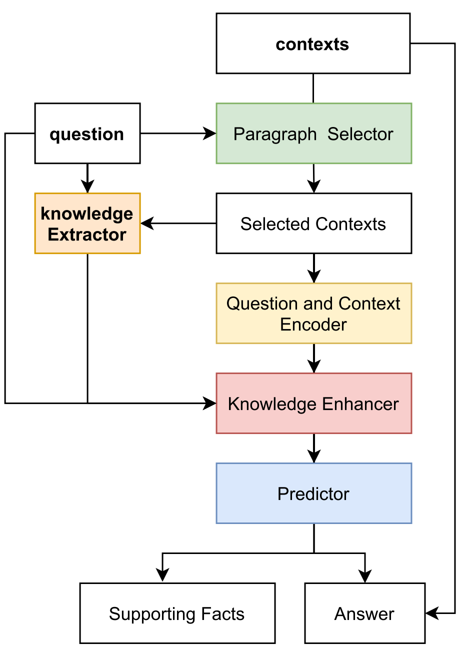

Fig. 2 Overall framework for multi-hop question answering with five components: (1) contexts Selector (2) Knowledge Extractor (3) Question and Context Encoder (4) Knowledge Enhancer (5) Predictor.

Table 1 Several examples for relation textual templates.

Relationship	Textual Expression
⟨ A, RelatedTo, B ⟩	There is some positive relationship between A and B.
⟨ A, FormOf, B ⟩	A is an inflected form of B.
⟨ A, PartOf, B ⟩	A is part of B.
⟨ A, UsedFor, B ⟩	A is used for B.
⟨ A, Causes, B ⟩	A and B are events, and it is typical for A to cause B.
⟨ A, Desires, B ⟩	A is a conscious entity that typically wants B.
⟨ A, LocatedNear, B ⟩	A and B are typically found near each other.

[†]https://stanfordnlp.github.io/CoreNLP/

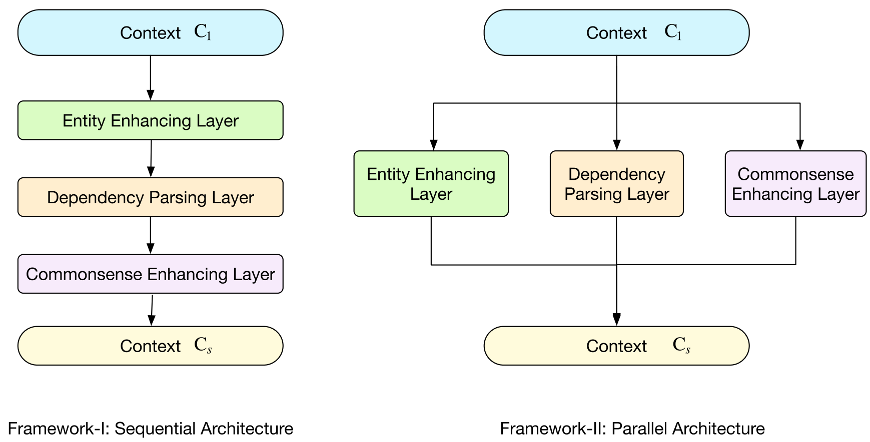

Fig. 3 Two different frameworks of multi-dimensional knowledge enhanced graph network: (a) *Framework-I: sequential architecture* and (b) *Framework-II: parallel architecture*.

$\mathbf{O}_{end}$, and the answer type $\mathbf{O}_{type}$. The first LSTM F_0 takes $\mathbf{C}_2$ as input, and each F_i outputs a logit $\mathbf{O} \in R^{l\times d_2}$ and computes a cross entropy loss over these logits, where l is the length of C_s. The prediction layer is formulated as follows:

$$\mathbf{O}_{sup} = F_0(\mathbf{C}_2) \tag{1}$$
$$\mathbf{O}_{start} = F_1(\mathbf{C}_2, \mathbf{O}_{sup}) \tag{2}$$
$$\mathbf{O}_{end} = F_2(\mathbf{C}_2, \mathbf{O}_{start}) \tag{3}$$
$$\mathbf{O}_{type} = F_3(\mathbf{C}_2, \mathbf{O}_{sup}, \mathbf{O}_{end}) \tag{4}$$

The loss fuction is defined as Eq. (5):

$$L = L_{start} + L_{end} + \lambda_s L_{sup} + \lambda_t L_{type} \tag{5}$$

3.2 Multi-Dimensional Knowledge Graph Network

In the previous section, we introduce the overall framework for mutli-hop QA task, which contains an important component, i.e., knowledge enhancer, also named as **M**ulti-dimensional **K**nowledge enhanced **G**raph **N**etwork (MKGN) as shown in the red box in Fig. 2. MKGN is designed to make the most of multi-dimensional knowledge for representation enhancement and interactions between question and contexts. Aiming at the two issues, we design two different architectures, i.e., in *sequential* and *parallel* manner, for knowledge enhancement as shown in Fig. 3.

3.2.1 Framework-I: Sequential Architecture

In Framework-I, we fuse entity information, dependency relations, and commonsense one by one, aiming to stimulate a sequential reasoning process with various knowledge. We take representations of context C_1 and question Q_1 as input for MKGN. M_E, P and E^{CS} denote three kinds of knowledge generated by named entity recognition, dependency parsing and commonsense extraction based on a commonsense knowledge base ConceptNet [19], respectively. The sequential architecture is implemented as follows:

1) we consider the knowledge of entities to enhancing the encoding of context and questions. We generate the representations of entity knowledge through Entity Enhancing Layer with context $\mathbf{C}_1$, question $\mathbf{Q}_1$, and entity mapping matrix $\mathbf{M}_E$ as inputs:

$$\mathbf{E}_u = EntityEnhancingModule(\mathbf{C}_1, \mathbf{Q}_1, \mathbf{M}_E) \tag{6}$$

where $\mathbf{M}_E$ is a binary mapping matrix generated through NER pre-processing progress. Concretely, $M_{i,j}$ is 1 if $i-th$ token in the context is within the span of the $j-th$ entity. Therefore, the shape of M_E is lxN, where N denotes the number of entities in the context. M_E is used to select the text span for the entity. The token embeddings, which is a matrix containing only selected columns of C_1, is passed into a mean-max pooling to calculate entity embeddings $E_0 = [e_0, e_1, \ldots, e_N]$. E_0 will be of size $2d_2XN$, and each of the $2d_2$ will produce both mean-pooling and max-pooling results. Then we use a residual layer to avoid forgetting initial context C_1 and a LSTM layer to model the long-distance dependency in context.

$$\mathbf{C}_E = LSTM(\mathbf{C}_1 + \mathbf{M}_E\mathbf{E}_u) \tag{7}$$

And a bidirectional attention layer [14] is used to enhance the representation of question:

$$\mathbf{Q}_E = Bi-Attention(\mathbf{Q}_1, \mathbf{E}_u) \tag{8}$$

2) In the sequential manner, we feed the outputs C_E and Q_E of the last step and dependency-relation matrix P to Parsing Enhancing Layer, which is also followed by a residual layer and LSTM layer for context and a bi-attention layer of questions. The process can be formulated as:

$$\mathbf{P}_u = ParsingEnhancingModule(\mathbf{C}_E, \mathbf{Q}_E, \mathbf{P}) \tag{9}$$
$$\mathbf{C}_P = LSTM(\mathbf{C}_1 + \mathbf{P}_u) \tag{10}$$
$$\mathbf{Q}_P = Bi-Attention(\mathbf{Q}_E, \mathbf{P}_u) \tag{11}$$

3) For the usage of commonsense knowledge, we conduct the same operation as previous steps:

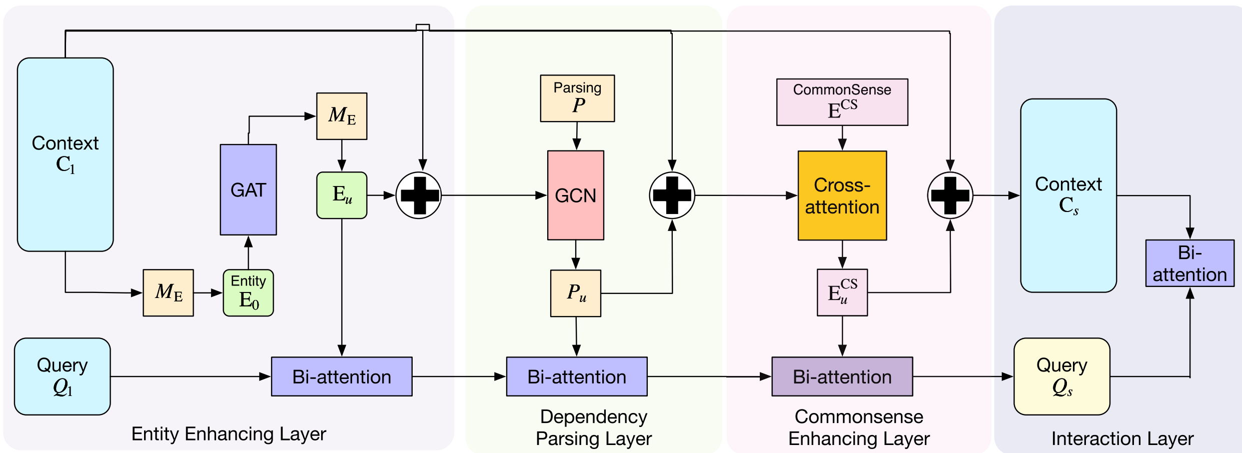

Fig. 4 Overview of MKGN on $\mathbf{Q}_1$ and $\mathbf{C}_1$ pairs in the sequential manner.

$$\mathbf{E}_u^{CS} = CommonsenseEnhancingModule(\mathbf{C}_P, \mathbf{E}^{CS}) \quad (12)$$

$$\mathbf{C}_S = LSTM(\mathbf{C}_1 + \mathbf{E}_u^{CS}) \quad (13)$$

$$\mathbf{Q}_S = Bi-Attention(\mathbf{Q}_P, \mathbf{E}_u^{CS}) \quad (14)$$

where $\mathbf{E}^{CS}$ represents the concatenation of words embedding in commonsense reasoning paths. We following [17] to extract commonsense reasoning paths. Briefly, we extract ConceptNet triples with different head concepts. If the head concept of one triple is the tail concept of another triple, we regard this relation as a reasoning path. This process can be formatted as follow: ⟨ concept-1, relationship-1, concept-2 ⟩ + ⟨ concept-2, relationship-2, concept-3 ⟩ + ...=> ⟨ concept-1, relationship-1, concept-2, relationship-2, concept-3, ...⟩. Since we convert each triple into a sentence, and for each commonsense reasoning path, it textual format is the concatenation of these sentences whose corresponding triples which combine the reasoning path.

4) Finally, to ensure the full interaction between questions and contexts, we apply a bidirectional attention operation again on knowledge-enhanced question and context. Different from DFGN, whose interaction only depends on the second fusion layer, we argue that the interaction between questions and contexts should be performed more frequently since questions and contexts are always updated with each knowledge. Therefore, every time when the question and context representations are enhanced, the interaction should be conducted in time.

$$\mathbf{C}_2, \mathbf{Q}_2 = Bi-Attention(\mathbf{C}_S, \mathbf{Q}_S) \quad (15)$$

3.2.2 Framework-II: Parallel Architecture

According to the fact that humans exploit multiple knowledge at the same time when making inferences and decisions. Therefore, we consider a parallel architecture for multi-dimensional knowledge utilization in Framework-II. Concretely, each knowledge enhancing layer of this architecture takes the initial question Q_1 and context C_1 as inputs. After obtaining the representations of each knowledge, we concatenate them with context representation $\mathbf{C}_1$ as follows:

$$\mathbf{C}_S = \mathbf{W}_i[\mathbf{C}_1; \mathbf{M}_E\mathbf{E}_u; \mathbf{P}_u; \mathbf{E}_u^{CS}] \quad (16)$$

$$\begin{aligned}(Q)_S = (W)_i[&\mathbf{Q}_1;\\ &Bi-Attention(\mathbf{Q}_1, \mathbf{E}_u);\\ &Bi-Attention(\mathbf{Q}_1, \mathbf{P}_u);\\ &Bi-Attention(\mathbf{Q}_1, \mathbf{E}_u^{CS})]\end{aligned} \quad (17)$$

where $\mathbf{E}_u$, $\mathbf{P}_u$, and $\mathbf{E}_u^{CS}$ represent separately entity representations, dependency relation representation and commonsense representation for word i in context, respectively. An interaction layer based on bidirectional attention is also applied as the last step of Framework-II.

3.3 Modules of MKGN

In this section, we elaborate our implementation of each modules based on *Framework-I Sequential Architecture* as shown in Fig. 4.

3.3.1 Entity Enhancing Module

This module is designed for information propagation among different entities and the use of GNN aims to capture relations across various entities better. Firstly, we extract entities from contexts with a named entity recognition (NER) model based on BERT, and then construct the entity graph following [5]. To propagate information across the entity graph, we apply graph attention network (GAT) to update entity representations. But the difference of our work is that we suppose each pair of entity nodes has an edge between them, and every kind of edge represents a type of relations. Different from DFGN using a binary matrix to represent three kinds of edges (i.e., sentence-level, context-level, and paragraph-level), we define the edge embeddings for each type. Besides, except for the above three types, we regard "no-find" as the fourth type for unknown relations between two entities, because different entities in Knowledge Base or the real world usually have some unknown relations. The initial representations of entities are calculated by a binary mapping matrix $\mathbf{M}_E$.

$$\mathbf{E}_0 = \mathbf{M}_E \mathbf{C}_1 \tag{18}$$

where $\mathbf{E}_0 = [\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_i, \ldots, \mathbf{e}_N]$. Therefore, the above process can be formulated as:

$$\mathbf{h}_i = \mathbf{U}\mathbf{e}_i + \mathbf{b} \tag{19}$$

$$\beta_{i,j} = LeakyReLU(\mathbf{W}_t^T[\mathbf{h}_i, \mathbf{h}_j, \mathbf{edge}_{i,j}]) \tag{20}$$

where $\mathbf{edge}_{i,j}$ denotes the edge embedding between the i-th entity and the j-th entity. During preprocessing, we construct a matrix $T \in NxN$ to record edge types among entities, where $T_{i,j} \in 1,2,3,4$ denotes the edge type between the i-th entity and the j-th entity and there are four kinds of edge types. We randomly initialize the edge embedding and edge emebeddings $\mathbf{EdgeEmbedding} \in R^{4x2d_2}$ are learnable in the training process. Therefore,

$$\mathbf{edge}_{i,j} = \mathbf{EdgeEmbedding}(T_{i,j}) \tag{21}$$

$$\alpha_{i,j} = \frac{exp(\beta_{i,j})}{\sum_k exp(\beta_{i,k})} \tag{22}$$

$$\hat{\mathbf{e}}_i = ReLU(\sum_{j \in B_i} \alpha_{j,i}\mathbf{h}_j) \tag{23}$$

where $U_t \in R^{d_2 \times 2d_2}$ is weight matrix, B_i represents the set of neighbors of entity i, the outputs of GAT is $\mathbf{E}_u = [\hat{\mathbf{e}}_0, \hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_n]$, and n is the number of entities in the context $\mathbf{C}_1$.

3.3.2 Parsing Enhancing Module

Inspired by [20], we enrich the representations of dependency information with graph convolution network (GCN) [9]. Firstly, we use Stanford Corenlp tools to perform dependency parsing on sentences in questions and contexts. Then we transform the dependency parsing tree to a binary adjacent matrix. Considering the sentence with n words, it can be modeled as a graph with n nodes and a $n \times n$ adjacency matrix P where $P_{ij} = 1$ if a dependency relation is going from word i to word j directly. GCN is used to update each token representations as [20]. If we denote by $\mathbf{h}_i$ the input vector $\mathbf{C}_E = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_i, \ldots, \mathbf{h}_l]$ and $\hat{\mathbf{h}}_i$ the output vector of word i, a graph convolution operation can be written as

$$\hat{\mathbf{h}}_i = \sigma(\sum_{j=1}^{n} \tilde{\mathbf{P}}_{ij}\mathbf{W}\mathbf{h}_j / d_i + \mathbf{b}) \tag{24}$$

where $\mathbf{P} = \tilde{}\mathbf{P} + \mathbf{I}$ with $\mathbf{I}$ is the $n \times n$ identity matrix, and $d_i = \sum_{j=1}^{n} \tilde{\mathbf{P}}_{ij}$ is the degree of token i in the resulting graphs. Besides, $\mathbf{W}$ is a linear transformation, b a bias term, and σ is a nonlinear function (e.g. ReLU). During graph convolution, each node gathers and summarizes information from its neighboring nodes in the graph. The output of GCN layer is $\mathbf{P}_u = [\hat{\mathbf{h}}_0, \hat{\mathbf{h}}_1, \ldots, \hat{\mathbf{h}}_i, \ldots, \hat{\mathbf{h}}_L]$.

3.3.3 Commonsense Enhancing Module

As for commonsense knowledge extractions, we following [17] extract commonsense reasoning sequence for each question and context pairs. We first select multi-hop relational commonsense information from ConceptNet via a point-wise mutual information and term-frequency based scoring function. Then we encode them with a BERT-based encoder. By concatenating the embedded commonsense sequence, we get a single vector representation, $\mathbf{e}_i^{CS}$ and $\mathbf{E}_{CS} = [\mathbf{e}_0^{CS}, \mathbf{e}_1^{CS}, \ldots, \mathbf{e}_i^{CS}, \ldots, \mathbf{e}_S^{CS}]$, where S denotes that the number of concepts selected from the context. Finally we project it into the same dimension as c_i^t and use an attention mechanism to model the interaction between commonsense and context or questions.

$$\mathbf{v}_i^{CS} = ReLU(\mathbf{W}\mathbf{e}_i^{CS} + \mathbf{b}) \tag{25}$$

$$S_{ij}^{CS} = \mathbf{W}_1^{CS}\mathbf{c}_i + \mathbf{W}_2^{CS}\mathbf{v}_j CS + \mathbf{W}_3^{CS}(\mathbf{c}_i \odot \mathbf{v}_j^{CS}) \tag{26}$$

$$p_{ij}^{CS} = \frac{exp(S_{ij}^{CS})}{\sum_{k=1}^{l} exp(S_{ij}^{CS})} \tag{27}$$

$$\mathbf{c}_i^{CS} = \sum_{j=1}^{l} p_{ij}^{CS}\mathbf{v}_j^{CS} \tag{28}$$

We use this extracted commonsense information through a selectively-gated attention mechanism to enrich this representations as follows:

$$\mathbf{z}_i = \sigma(\mathbf{W}_z[\mathbf{c}_i^{CS}; \mathbf{c}_i] + \mathbf{b}_z) \tag{29}$$

$$(\mathbf{e}_u^{CS})_i = \mathbf{z}_i \odot \mathbf{c}_i + (1 - \mathbf{z}_i) \odot \mathbf{c}_i^{CS} \tag{30}$$

And $\mathbf{E}_u^{CS} = [(\mathbf{e}_u^{CS})_0, (\mathbf{e}_u^{CS})_1, \ldots, (\mathbf{e}_u^{CS})_i, \ldots, (\mathbf{e}_u^{CS})_N]$, which denotes output of Commonsense Enhancing Module.

4. Experiments

4.1 Datasets

We evaluate our approach on the HotpotQA dataset. HotpotQA is a new machine reading comprehension benchmark that aims to test the model's capacity of multi-hop reasoning on several contexts with scattered evidence. It contains 130k wikipedia-based question-answering pairs and each question has ten corresponding passages with two gold contexts in them.

4.2 Implementation Details

We implement our MKGN based on DFGN, which is initialized with its default settings. We also apply the same

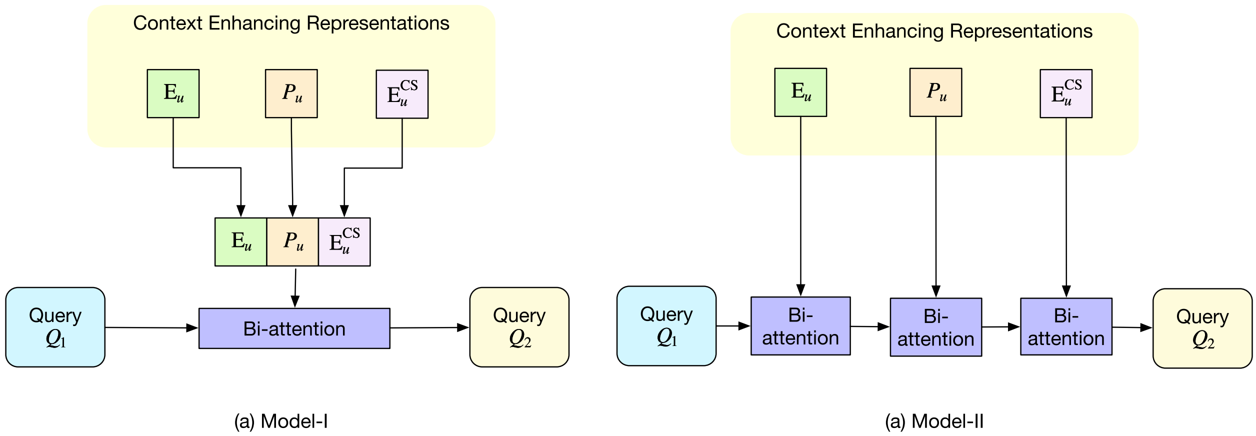

Fig. 5 Two different models of knowledge enhancement layer on question.

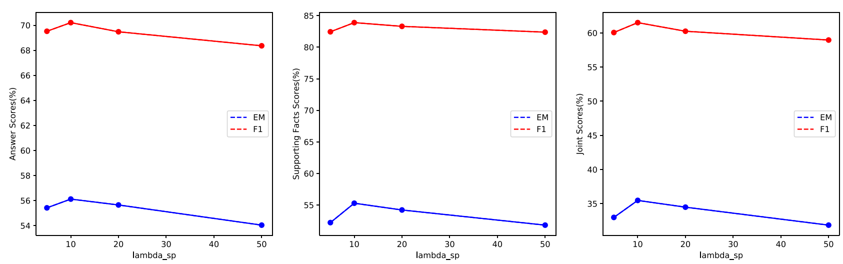

Fig. 6 Performances of MKGN model with different values of λ_{sp}. When $\lambda_{sp} = 10$, the performances of MKGN on both metrics are better than thoese under other setting values.

data preprocessing on training dataset. For entity encoding, we set the dimension of edge features to 300 and the number of edge type is 4. For parsing extraction, we use Stanford CoreNLP parser to do dependency parsing on both questions and contexts. The GCN is only one layer with a dropout of 0.5. The commonsense knowledge extraction is implemented† following [17]. We also use pretrained BERT model to encode selected commonsense sequence, and the dimension of word embedding is 768 based on Bert-base-uncased model. As for hyperparameters for training model, the learning rate is 1e-4, batchsize is 32.

4.3 Hyper-Parameters

We also train our model on several groups of hyperparameters to find the best model on the development set, as shown in Fig. 6. And we find the factor λ_{sp} of supporting facts has an obvious effect of the model performance. So we evaluate the model with different values of hyperparameters λ_{sp} on the development set of HotpotQA. As shown in Fig. 6, with different λ_{sp}, the EM and F1 scores of answering performance change a lot. When $\lambda_{sp} = 10$, we get the best model. And the factor to control type prediction λ_{type} is set to 1.

4.4 Evaluation Metrics

We use two different metrics on answer prediction, supporting facts and joint of the first two, which are provided by HotpotQA leaderboard to evaluate the model performance. **Exact Match** measures the percentage of predictions that match the corresponding ground truth answers exactly. **F1 score** measures the average overlap between the prediction and ground truth answer on fuzzy matching.

4.5 Overall Performance

We first submit our approach on the hidden test set of HotpotQA for evaluation, which is shown in Table 2. We use the Framework-I as the default model†† and only report the best result. As we can see, our system obtains a better results by achieving an EM score of 57.09 and a F1 score of 70.69 for answer predicting and two-point improvement with an EM score of 54.26 and F1 score of 83.54 for supporting facts on the test set, compared to another strong baseline DFGN.

Compared with SAE [22] model, there is still a gap between the performance of our MKGN and that of SAE. To further elaborate the differences between our MKGN and SAE, we compare our approach with SAE based on the ablation analysis of SAE as shown in Table 3. We observed that the improvements of SAE mainly comes from the new selector, which can reach 7.12 point (i.e., 66.45 vs 59.33), compared with baseline which uses the same selector as DFGN. Besides, their baseline method (“answer and explain” mod-

†https://github.com/yicheng-w/CommonSenseMultiHopQA

††We choose the Framework-I, the *sequential* one according to the performances of two frameworks on development set.

ule with the same selector as DFGN) can only achieve 59.33 on F1 score, while our method based on the DFGN selector which obtains 61.51 on F1 scores. Additionally, SAE and our method can be combined with each other to obtain a stacked improvements, which is an engineering work not research work and thus we do not discuss further.

We also conduct comparisons with other models (i.e., FFReader-large [23], and HGN [7]), which introduce their methods in their paper or open their codes, we make a detailed analysis about this gap.

For FFReader-large and SAE, they pay attention to optimize the paragraph selector with a long encoder to make full use of all the contexts for each question, while we only following DFGN adopts a simple BERT-based classifier. Therefore, it is unfair to directly compare our method with them. Besides, based on analysis of SAE, its main improvements comes from the new selector. If using the same DFGN selector, SAE only with encoder improves does not outperform our method.

For SAE and HGN, they both consider using the graph neural network to enhance the representations of context, but they does not consider using external knowledge (i.e., commonsense and dependency parsing). Our work focuses on explore not only he effectiveness of the above knowledge, but also how to exploit them together and the relationship among them. The motivation is different and the methods is not conflict but complementary, which can be combined with ours to obtain a stacked together. Since the combination of these methods is closer to an engineering project, not research work. Here we do not discuss a lot.

We also compare the performance of our model on various types of questions, shown in Table 4. By contrast, we find that MKGN achieves significant improvements mainly on“Bridge” type of examples, which suggests that MKGN does better in “Bridge” type of reasoning. However, there is no obvious improvement on “Comparison” Reasoning. Besides, both “DFGN” and “MKGN” demonstrate a same tendency that their performance under “Comparison” type is better than “Bridge” type. We conjecture that answers for “Comparison” type questions usually appear in the questions themselves and is easy to find in the context, while answers for “Bridge” type question always only occur in the supporting facts and they are more difficult to detect.

Table 2 Performance comparison on the private test set of HotpotQA in the distractor setting.

Models	Answer		Sup Fact		Joint	
	EM	F1	EM	F1	EM	F1
Bert-based						
Baseline Model [21]	45.60	59.02	20.32	64.49	10.83	40.16
DFGN [5]	56.31	69.69	51.50	81.62	33.62	59.82
SAE [22]	60.36	73.58	56.93	84.63	38.81	64.96
MKGN(ours)	57.09	70.69	54.26	83.54	35.59	61.69
Roberta-large						
FFReader-large [23]	68.89	82.16	62.10	88.42	45.61	73.78
HGN [7]	66.07	79.36	60.33	87.33	43.57	71.03

Table 3 Detailed comparisons with each module of SAE on HotpotQA dev set.

Model	joint EM	joint F1
SAE (full model)	39.89	66.45
SAE (DFGN selector)	31.87	59.33
MKGN (DFGN selector)	35.48	61.51

Table 4 Performance breakdown over different types on the dev set of HotpotQA in the distraction setting. ‘*’ denotes the results cited from [22].

Qtype	Bridge (5918 examples)		Comparison (1497 examples)	
	Joint-EM	Joint-F1	Joint-EM	Joint-F1
DFGN*	30.09	58.61	47.95	64.79
MKGN	**32.44**	**60.69**	**47.54**	**64.80**

5. Analysis and Discussion

5.1 Comparison of Different Frameworks

Table 5 provides details about the results of two architectures for the knowledge enhancement on context. Framework-I performs better on most of the evaluation metrics, achieving a F1 score of 70.81 on the answer prediction and 83.23 on the supporting facts. This illustrates that the sequential architecture can bring further improvement than parallel architecture in capturing the answer spans. Meantime, parallel architecture obtained a better scores of 53.30% on **E**xact **M**atching (EM) of supporting facts. This reflects the effectiveness of our knowledge enhanced modules in different fusion architectures. We can apply MKGN to more tasks which need to introduce external knowledge.

To further explore the effects of composing knowledge modules variously, we also compare two knowledge enhancement methods on question as Fig. 5. Table 6 displays the result of Model-I and Model-II. The knowledge enhancement on both question and contexts can always result in considerable performance gains, and Model-I obtains the best result over the other models, which achieves signif-

Table 5 Comparison of different fusion architectures for the knowledge enhancement context representation on the development set of HotpotQA in the distractor settings. *F-I* represents Framework-I sequential architecture; *F-II* represents Framework-II parallel architecture.

Config	Answer		Sup Fact		Joint	
	EM	F1	EM	F1	EM	F1
F-I	**56.66**	**70.81**	52.59	**83.23**	33.72	**61.43**
F-II	56.52	70.57	**53.30**	82.96	**34.50**	61.28

Table 6 Comparison of performances on the whole system.*M-I* : Model-I, *M-II*: Model-II in Fig. 6.

Config	Answer		Sup Fact		Joint	
	EM	F1	EM	F1	EM	F1
M-I	56.12	70.21	**55.29**	**83.89**	**35.48**	**61.51**
M-II	**56.66**	**70.81**	52.59	83.23	33.72	61.43

Table 7 Ablation study of question answering performances in the development set of HotpotQA in the distractor setting. "inter": ablate interaction between question and context; "Q":ablate the knowledge enhancement part on questions; "cs": ablate the commonsense enhancing layer; "gcn":ablate the parsing enhancing layer; "edge": ablate the edge change in the GAT modules; "cs+gcn": ablate both commonsense and parsing enhancing layer; "edge+gcn": ablate both edge change and parsing enhancing layer; "cs+edge": ablate both commonsense and edge change; "gold": only gold contexts; "sup":only supporting facts.

Setting	Answer		Sup Fact		Joint	
	EM	F1	EM	F1	EM	F1
ours	56.12	70.21	**55.29**	**83.89**	**35.48**	**61.51**
w/o inter	56.15	70.22	51.68	82.98	32.87	60.84
w/o Q	55.31	69.02	52.69	82.50	33.69	59.69
w/o cs	56.02	70.14	53.42	82.85	34.23	60.81
w/o gcn	**56.42**	**70.34**	54.21	83.72	34.76	61.50
w/o edge	56.04	69.71	53.18	82.96	33.87	60.41
w/o cs+gcn	55.57	69.90	50.16	82.71	31.72	60.23
w/o edge+gcn	55.67	69.76	52.23	83.24	32.56	60.46
w/o cs+edge	55.25	69.61	52.21	82.51	32.91	60.12
gold	58.60	72.90	60.47	88.34	38.38	65.80
sup	58.84	72.90	-	-	-	-

icant improvements on EM scores of supporting facts, (i.e., from 52.59 to 55.29). Similar observation can be found in parallel architecture, demonstrating that the gains are consistent and stable.

5.2 Ablation Study

We conduct ablation study on HotpotQA development set in the distractor setting to evaluate the effects of each individual component in MKGN. Table 7 displays results of different kinds of knowledge fused on the baseline model. Firstly, we delete the direct interaction between knowledge enhanced question and contexts. After removing the knowledge enhancement on question, all evaluate metrics drop obviously. This illustrates that the enhancement and update of question representations are important part to QA task.

Removing the commonsense enhancing parts results in a performance drop for all evaluation metrics, indicating that this module helps the model to better predict both answers and supporting facts. Deleting the modified edge module in entity enhancing layer causes a degradation on the overall performance in terms of EM and F1.

Remarkably, when ablating the parsing enhancing layer 'w/o gcn', the performance of model obtains improvements on answer prediction and decline on supporting facts predictions. It seems inconsistent with our error analysis on DFGN. Firstly, we observed that our MKGN uses the dependency parsing relations by GCN to enhance the word embeddings, which is more helpful for finding the supporting sentences than detecting the answer spans accurately, since dependency parsing models the relations among words or elements across sentences, not entity or phrase spans.

Furthermore, to clarify the benefits of multi-dimensional knowledge and the effects of their relationship, we conduct extra experiments by ablating two kinds of knowledge at the same time. The results are shown in

Table 8 Comparison between DFGN and MKGN based on RoBERTa-large.

Models	Answer		Sup Fact		Joint	
	EM	F1	EM	F1	EM	F1
DFGN-R	63.73	78.09	56.68	85.65	39.97	69.13
MKGN-R	**64.88**	**78.86**	**58.83**	**86.12**	**42.09**	**70.16**

Table 7, i.e., "w/o cs+gcn", "w/o edge+gcn", and "w/o cs+edge". There are three findings, 1) if only using one kind of knowledge, the effectiveness of knowledge from high to low is commonsense, entity, and dependency parsing; 2) When combining different kinds of knowledge, they can promote each other, especially dependency parsing knowledge for entity and commonsense; 3) Dependency parsing knowledge contributes more to supporting fact detection than answering prediction.

Besides, we also conduct the data ablation experiment in the "gold contexts only" and "supporting facts only" and the results show that our model is little affected by the noise data.

5.3 Impact of Pretrained Language Model

We implement our method with two pre-trained language model as encoder, i.e., *BERT-base* and *RoBERTa-large*. As shown in Table 8, our method achieves joint-F1 scores of 61.51 and 70.16 on them respectively, suggesting that enhancing the representative ability of text encoder does influence a lot. Furthermore, we also re-implement DFGN with *RoBERTa-large*. Results show that our method can still achieve significant improvements over a stronger pre-trained language models, which demonstrate that it is essential to introduce specific external knowledge.

5.4 Case Study

To further explore the effect of different knowledge, we choose three cases from the development sets of HotpotQA as shown in Fig. 7. We compare predictions from both MKGN and DFGN to show the differences of using knowledge before and after.

- In case (a), both of MKGN and DFGN find "the team" in the first supporting fact refer to "Lewiston Maineiacs". At the second-hop reasoning, MKGN depends on the co-reference relations of "Andrisciggin Rank *Colisée*" and obtains the right answers of "3,677". However, DFGN is unclear to the object that "the main rink" refer to and find the wrong answer "1,400" people.
- In case (b), compared to MKGN, DFGN is weak in modeling the relations of entities. Therefore, DFGN misses the supporting fact and get wrong answer. MKGN models the relation of all entities in an attention mechanism rather than defining in rules as DFGN.
- In case (c), although DFGN and MKGN find the same supporting facts, DFGN still can not obtain the right

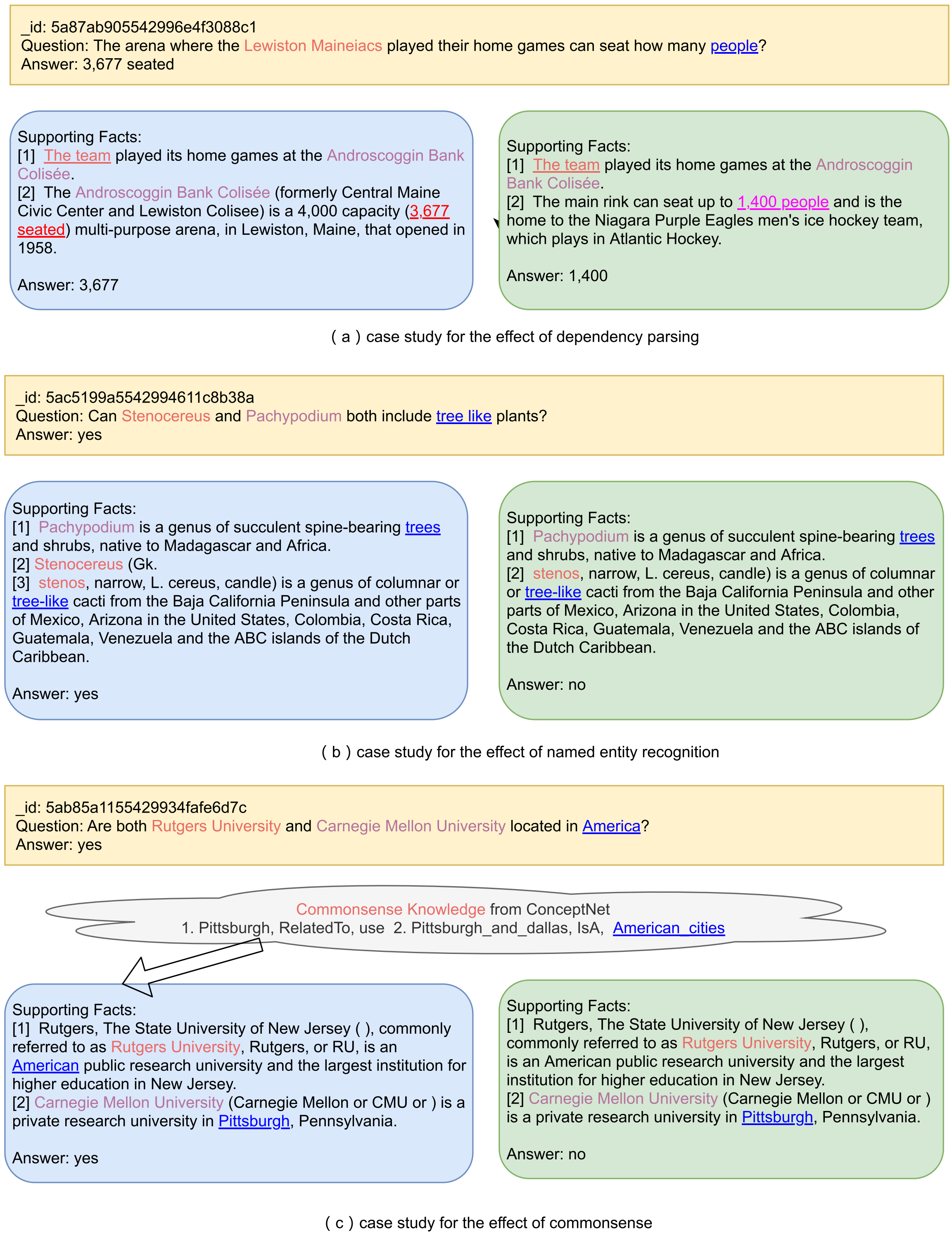

Fig. 7 Cases selected from HotpotQA development sets. The yellow rectangles exhibit the question, its unique id number and gold answer for each case. The blue and green rectangles exhibit the supporting facts and answers predicted by MKGN and DFGN, respectively. The cloud in case (c) represent commonsense knowledge selected from ConceptNet. Textual words in different colors represent different entities. Underlined words in deep blue colors are clues in questions or directly related to answers. For case (a), two answers are given in red or hot-pink are predicted answers.

answer, due to the gap between question and supporting facts. The commonsense that "Pittsburgh belongs to America" is necessary to correctly answer the questions, but can not find in context. For MKGN, we use ConceptNet to complement the knowledge gaps and make the reasoning path completed.

The above cases concretely display knowledge gaps exist in question and contexts, which demonstrate the necessity of injecting external knowledge into representations of question and contexts. How to extract knowledge more accurately and efficiently can be an open question.

6. Related Work

6.1 Multi-Hop Question Answering

With the release of HotpotQA, Yang et al. [21] modify the biDAF [14] as a baseline for multi-hop QA. Although this method is not capable, it provides a fundamental paradigm, i.e., context selection and question answering. To improve context selection, some work focuses on using query or answer information to guide iterative selection [2], [3] and others pay attention to design a specific selector according to the relations of sentences across documents [22], [24]. Different from above methods, our approach focuses on utilizing external knowledge to improve question answering process, not retrieving process. Therefore, our method can be combined with these methods to obtain further improvements.

For advances in question answering, most previous work pay attention to enhancing document representations through GNNs. Some of them exploit GNN to incorporate entity knowledge into representations of query and contexts [4], [5], [25]. Besides, Fang et al. [7] and Gao et al. [8] utilize hierarchical graphs and heterogeneous graphs to encode multi-grained information, respectively. Moreover, based on cognitive knowledge, Ding et al. [6] construct a cognitive graph to update representations of candidate answers. Different from these studies, our MKGN utilizes multi-dimensional knowledge i.e., not only entities but also dependency relations and commonsense. Besides, we exploit both GNNs and bidirectional attention mechanism to enhancing representation and interaction between contexts and queries.

Additionally, several studies focus on modeling reasoning process explicitly through decomposing complex, multi-hop questions to simple, single-hop questions [13] or design a discrete reasoning path in a step-by-step manner [10]–[12]. Our approach is different from above methods at jointly training the multi-hop QA model in an end-to-end manner, while these methods apply a step-by-step method to show an explicit reasoning process, which may suffer from error propagation.

6.2 Knowledge Enhancement

Our work is also inspired by recent studies on introducing external knowledge to other natural language processing tasks and their great success, e.g., dependency parsing for relation extraction [20], commonsense for multi-choice question answering [26], artificial reasoning rules [27], and heterogeneous knowledge [28]. Different from the aforementioned researches, our approach introduces multi-dimensional external knowledge, i.e., entities, dependency relations, and commonsense, to repair the knowledge gap and improve the reasoning process.

7. Conclusion

We propose a **M**ulti-dimensional **K**nowledge enhanced **G**raph **N**etwork (MKGN) for multi-hop question answering. The proposed model effectively exploits different kinds of knowledge, such as entities, dependency relations and commonsense, to enhance the representations of question and context through graph networks. To further mimic reasoning behaviours of humans, we investigate two various frameworks, *i.e.*, in the *sequential* and *parallel* manner. In addition, we add the bi-attention layer each time when the representations of contexts and question are updated. Experimental results show that the proposed MKGN in two architectures indeed bring improvements on HotpotQA dataset. Besides, the ablation studies verify the effectiveness of several proposed components in our model, and analyses show that the MKGN model is superior in solving relatively complex questions.

Acknowledgements

The research work descried in this paper has been supported by the National Nature Science Foundation of China (No. 61976016, 61976015, and 61876198) and the Key Technologies Research and Development Program of China (2019YFB1405200), and the authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

[1] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol.2, pp.784–789, 2018.

[2] Y. Feldman and R. El-Yaniv, "Multi-hop paragraph retrieval for open-domain question answering," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp.2296–2309, Association for Computational Linguistics, July 2019.

[3] K. Nishida, K. Nishida, M. Nagata, A. Otsuka, I. Saito, H. Asano, and J. Tomita, "Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp.2335–2345, Association for Computational Linguistics, July 2019.

[4] N. De Cao, W. Aziz, and I. Titov, "Question answering by reasoning across documents with graph convolutional networks," 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), Minneapolis, Minnesota, pp.2306–2317, Association for Computational Linguistics, June 2019.

[5] L. Qiu, Y. Xiao, Y. Qu, H. Zhou, L. Li, W. Zhang, and Y. Yu, "Dynamically fused graph network for multi-hop reasoning," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp.6140–6150, Association for

Computational Linguistics, July 2019.
[6] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp.2694–2703, Association for Computational Linguistics, July 2019.
[7] Y. Fang, S. Sun, Z. Gan, R. Pillai, S. Wang, and J. Liu, "Hierarchical graph network for multi-hop question answering," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.8823–8838, 2020.
[8] F. Gao, J.C. Ni, P. Gao, Z.L. Zhou, Y.Y. Li, and H. Fujita, "Heterogeneous graph attention network for multi-hop machine reading comprehension," arXiv preprint arXiv:2107.00841, 2021.
[9] T.N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.
[10] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over wikipedia graph for question answering," arXiv preprint arXiv:1911.10470, 2019.
[11] J. Chen, S.T. Lin, and G. Durrett, "Multi-hop question answering via reasoning chains," arXiv preprint arXiv:1910.02610, 2019.
[12] Y. Jiang and M. Bansal, "Self-assembling modular networks for interpretable multi-hop reasoning," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp.4474–4484, 2019.
[13] E. Perez, P. Lewis, W.-T. Yih, K. Cho, and D. Kiela, "Unsupervised question decomposition for question answering," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.8864–8880, 2020.
[14] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," arXiv preprint arXiv:1611.01603, 2016.
[15] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Mathematics of control, signals and systems, vol.2, no.4, pp.303–314, 1989.
[17] L. Bauer, Y. Wang, and M. Bansal, "Commonsense for generative multi-hop question answering tasks," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.4220–4230, Association for Computational Linguistics, Oct.-Nov. 2018.
[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol.9, no.8, pp.1735–1780, 1997.
[19] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," Thirty-first AAAI conference on artificial intelligence, pp.4444–4451, 2017.
[20] Y. Zhang, P. Qi, and C.D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.2205–2215, Association for Computational Linguistics, Oct.-Nov. 2018.
[21] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C.D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.2369–2380, Association for Computational Linguistics, Oct.-Nov. 2018.
[22] M. Tu, K. Huang, G. Wang, J. Huang, X. He, and B. Zhou, "Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents," Proceedings of the AAAI Conference on Artificial Intelligence, vol.34, no.5, pp.9073–9080, 2020.
[23] T. Alkhaldi, C. Chu, and S. Kurohashi, "Flexibly focusing on supporting facts, using bridge links, and jointly training specialized modules for multi-hop question answering," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.29, pp.3216–3225, 2021.
[24] D. Groeneveld, T. Khot, Mausam, A. Sabharwal, "A simple yet strong pipeline for hotpotqa," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.8839–8845, 2020.
[25] Y. Cao, M. Fang, and D. Tao, "Bag: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol.1, pp.357–362, 2019.
[26] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, pp.675–686, Association for Computational Linguistics, July 2018.
[27] L. Weber, P. Minervini, J. Münchmeyer, U. Leser, and T. Rocktäschel, "NLProlog: Reasoning with weak unification for question answering in natural language," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp.6151–6161, Association for Computational Linguistics, July 2019.
[28] M. Tu, G. Wang, J. Huang, Y. Tang, X. He, and B. Zhou, "Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs," Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp.2704–2713, Association for Computational Linguistics, July 2019.

Ying Zhang is currently pursuing the Ph.D. degree with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. Her current research interests include natural language processing, machine reading comprehension and information extraction.

Fandong Meng received the Ph.D. degree in Institute of Computing Technology, Chinese Academy of Sciences, and is now a research scientist and manager at Pattern Recognition Center, WeChat AI, Tencent Inc. His research interests include natural language processing, machine translation and dialogue system.

Jinchao Zhang was born in 1989. He received the Ph.D. degree in Chinese Academy of Sciences, and is now a senior researcher in Pattern Recognition Center, WeChat AI, Tencent Inc. His research interests include natural language processing and machine translation.

Yufeng Chen received the B.S. degree in Mechanical Electrical Engineering from Beijing Jiaotong University in 2003, and the Ph.D. degree in Pattern Recognition and Intelligent Systems from National Lab of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, in June, 2008. From July 2008 to September 2014, she worked in NLPR. From September 2014, she joined School of Computer and Information Technology, Beijing Jiaotong University and now works as an associate professor. Her research interests include natural language processing, machine translation, information extraction and so on.

Jinan Xu is currently a Professor in School of Computer Science and information technology, Beijing Jiaotong University, Beijing, China. He received the B.S. degree from Beijing Jiaotong University in 1992, the MS degree and Ph.D. degree in computer information from Hokkaido University, Sapporo, Japan, in 2003 and 2006, respectively. His research focuses on natural language processing, machine translation, information retrieve, text mining, and machine learning. He is a member of CCF, CIPSC, ACL and the ACM.

Jie Zhou received his bachelor degree from USTC in 2004 and his Ph.D. degree from Chinese Academy of Sciences in 2009, and is now a senior director of Pattern Recognition Center, WeChat AI, Tencent Inc. His research interests include natural language processing and machine learning.