sensors

MDPI

Article

Metaknowledge Enhanced Open Domain Question Answering with Wiki Documents

Shukan Liu [1,2,†], Ruilin Xu [2,†], Li Duan [2,*], Mingjie Li [3] and Yiming Liu [2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China; liusk@seu.edu.cn
[2] School of Electronic Engineering, PLA Naval University of Engineering, Wuhan 430033, China; kirinxu@foxmail.com (R.X.); liuyiming0507@foxmail.com (Y.L.)
[3] Ship Comprehensive Test and Training Base, PLA Naval University of Engineering, Wuhan 430033, China; jie19850625@126.com
* Correspondence: duanlidragon@126.com; Tel.: +86-1387-116-4786
† These two authors contributed equally to this work.

Abstract: The commonly-used large-scale knowledge bases have been facing challenges in open domain question answering tasks which are caused by the loose knowledge association and weak structural logic of triplet-based knowledge. To find a way out of this dilemma, this work proposes a novel metaknowledge enhanced approach for open domain question answering. We design an automatic approach to extract metaknowledge and build a metaknowledge network from Wiki documents. For the purpose of representing the directional weighted graph with hierarchical and semantic features, we present an original graph encoder GE4MK to model the metaknowledge network. Then, a metaknowledge enhanced graph reasoning model MEGr-Net is proposed for question answering, which aggregates both relational and neighboring interactions comparing with R-GCN and GAT. Experiments have proved the improvement of metaknowledge over main-stream triplet-based knowledge. We have found that the graph reasoning models and pre-trained language models also have influences on the metaknowledge enhanced question answering approaches.

Keywords: metaknowledge; graph modeling; question answering; graph neural networks; knowledge graph

check for updates

Citation: Liu, S.; Xu, R.; Duan, L.; Li, M.; Liu, Y. Metaknowledge Enhanced Open Domain Question Answering with Wiki Documents. *Sensors* **2021**, *21*, 8439. https://doi.org/10.3390/s21248439

Academic Editors: Friedhelm Schwenker and Mariofanna Milanova

Received: 3 November 2021
Accepted: 16 December 2021
Published: 17 December 2021

1. Introduction

With the rapid development of the artificial intelligence, the voice interaction devices are now becoming a significant application of the Internet, and the major Internet enterprises have all launched their own intelligent voice interaction devices. Intelligent voice interaction has already been used as a new generation of Internet portal after the search engine. It has also begun to enter a variety of application fields, such as mobile phones, smart homes, industrial control systems, etc. The prospect of intelligent voice interaction devices is extremely broad.

As a pivotal infrastructure of the *Metaverse*, the future voice interaction devices must not only support simple information retrieval tasks, but also have the capabilities of answering questions with complex semanteme and logicality, whereas current voice interaction devices are not able to deal with the complex application scenarios like open domain question answering.

Open domain Question Answering (QA) is a type of language task that asks models to answer the factoid questions described in natural language. Recently, large-scale Knowledge Bases (KBs), such as DBpedia [1], FreeBase [2], and YAGO [3], have proven to be effectively applied on the open domain QA tasks, while the idea of this kind of triplet-based knowledge is an adaptive variation of a complex network, which inherits its long-tail effect in the QA tasks due to triplets' sparsity and lack of logical association [4].

Obviously, the simplified triplet-based knowledge is not exactly the same as the knowledge in human beings' perception. Knowledge in human minds is a complex of hierarchical, structured, and systematized elements which has strongly logical or topological associations, especially presented in structure or sequence, while the very knowledge that exists in commonly-used knowledge bases is simplified and presented as *entity-relation* triplets.

While most of the existing works focus on triplet-based KBs, a more general definition about KBs and a various usages of KBs such as conceptual graph [5] and event evolutionary graph [6] have been proposed to improve the QA approaches and other task performance from different perspectives.

However, just like the taxonomy construction manufactured within the conceptual graph, the content of the documents and webs was hopefully to be explicitly represented through metadata in order to enable contents-guided search and other downstream tasks. However, the knowledge in the real world could hardly be strictly partitioned into the hand-craft-built or evolutional taxonomy [5] with accurate levels and divisions hierarchically. Since the taxonomy construction is tough, cumbersome and new knowledge always led to new partitioning and reconstruction problems, it is intuitively vital to consider another flexible presentation for the hierarchical knowledge.

To match human's natural intuition of knowledge, different from the strictly designed and partitioned conceptual graph, our previous work [7] introduces the concept of metaknowledge [8] into knowledge engineering research. Similar to the metadata, metaknowledge is a kind of graph data. It is a structural representation of knowledge and knowledge with fine-grained and hierarchical characteristics, but the knowledge triplets are weighted directional in hierarchy based on the structured information given by the original sources.

Firmly based on the open domain QA task, in this work, we have: (1) designed an automatic approach for generating metaknowledge and building metaknowledge network from Wiki documents; (2) proposed an original graph encoder GE4MK for modeling the metaknowledge (network) to the weighted directional graph with hierarchical and semantic features; (3) presented a graph reasoning model MEGr-Net for a metaknowledge enhanced open domain QA; and (4) carried out experiments for verifying the improvement of our metaknowledge-based open domain QA approach with triplet-based approaches.

2. Related Work

2.1. Knowledge Base Question Answering (KBQA)

The goal of KBQA is to use large-scale knowledge bases to answer questions described in natural language (natural questions), and the primary task is to understand and extract the actual semantic connotation from natural questions, then retrieve entities or relations in knowledge bases as the answers. Presently, there are two pipelines in KBQA: the Semantic-Parsing-based (SP-based) pipeline and the Information-Retrieval-based (IR-based) pipeline [4,9]. The early-days SP-based approaches mainly rely on hand-craft-established rules [10] and supervised learning [11]. Recently, the convolutional neural network [12], attention mechanism [13], graph2seq model [14], and reinforcement-learning-based approaches [15,16] are also used in SP-based KBQA.

With the rapid development of knowledge representation learning, the IR-based approaches have now become the mainstream in KBQA [17–20]. These approaches extract information from questions, retrieve the information in knowledge bases (knowledge graphs), and then use graph reasoning models to decide which entities or relations are the answers. Basically, the steps of the IR-based approaches are: (1) Getting the seed entities from the given natural question, retrieving seed entities in the knowledge base and then building a question subgraph, in which the entities and relations are all semantically associated with the seed entities. (2) Representing the given question with question encoder, which analyzes semantic features in the question and outputs a commanding vector (question embedding) for reasoning. (3) Reasoning with embedding of the given question and the question subgraph obtained in steps (1) and (2), and then getting the probability of whether it is the answer for each entity in the question subgraph. (4) Ranking the probability sequence and deciding the most-likely answer entity.

Meanwhile, it has been quite unsatisfying when using triplet-based KBs alone in complex KBQA tasks like multi-hop question answering, and the problem that triplet-based knowledge lacks structural logicality has become apparent. In order to make up for the capacity limitation of the existing KBs, a common practice is to introduce heterogeneous data like documents to enrich the semantic information, which is referred to as Document-based Question Answering (DbQA). Ref. [21] proposes a question answering model combining FreeBase and Wikipedia documents. In order to improve the QA effectiveness in the case of insufficient capacity of knowledge base, Ref. [22] proposes an early-fusion approach to link the entities of knowledge base with the text in the document. In the multi-hop QA task, Ref. [23] carries out multi-grained document modeling, constructs hierarchical graph, and demonstrates graph reasoning and answer prediction through the Machine Reading Comprehension (MRC) method. In the field of Visual Question Answering (VQA), Ref. [24] designs a model which uses adversarial learning with bidirectional attention to solve the VQA problem. Ref. [25] proposes the MESAN model, which is a multi-modal explicit sparse attention network, to solve the problem of attention distraction.

The inspiration of the above works is that the defects of the knowledge base can be made up for by improving the semantic parsing ability and introducing heterogeneous data represented by documents, with the intention of continuously improving the effectiveness of question answering.

2.2. Graph Neural Networks for Graph Embedding

The purpose of graph embedding is to represent the nodes, edges or subgraphs of a graph as low-dimensional vectors through neural networks. Classical graph embedding approaches are based on graph representation learning include DeepWalk, node2vec and LINE, etc. Recently, Graph Neural Networks (GNNs) have become the new tools for graph embedding. Ref. [26] proposes the Graph Convolutional Network (GCN) model and applies it to the self-supervised node classification task. On the basis of GCN, Ref. [27] models the complex relational data in the knowledge graph and puts forward the R-GCN (relational GCN) model, which uses two different parameters matrices for vertices and edges (relations). Inspired by the attention mechanism in Transformers [28], Ref. [29] proposes the Graph Attention Network (GAT) to comprehensively consider the influence of neighboring vertices on graph embedding.

The approaches for embedding relational graph proposed in R-GCN and the multi-heads attention mechanism in GAT provide enlightenment on how to realize the representation of graph data with complex relations and semantic information such as metaknowledge and metaknowledge networks.

3. Approach

Since the metaknowledge is different from the triplet-based knowledge, this work proposes an approach (Figure 1) to make metaknowledge available for question answering. (1) Metaknowledge generating: for each question, we use the Wiki retriever from DrQA [21] to get the top five relevant Wiki docs of the given question, then we design a novel metaknowledge extractor to generate metaknowledge from those documents. (2) Metaknowledge network construction and question subgraph retrieval: we use the question-entity link proposed in Ref. [12] to get the entities relevant to the question. We design a way to build semantic associations between metaknowledge extracted from docs. Then, we do subgraph retrieval to reduce the scale of data. (3) Metaknowledge encoding: we design a graph encoder for metaknowledge to transform the text-described metaknowledge into matrices for the further computation. (4) Graph reasoning: we propose a graph reasoning model MEGr-Net which turns the question answering into a node classification task, that is, for each vertex in the question subgraph, the MEGr-Net will decide whether the vertex is the right answer or not.

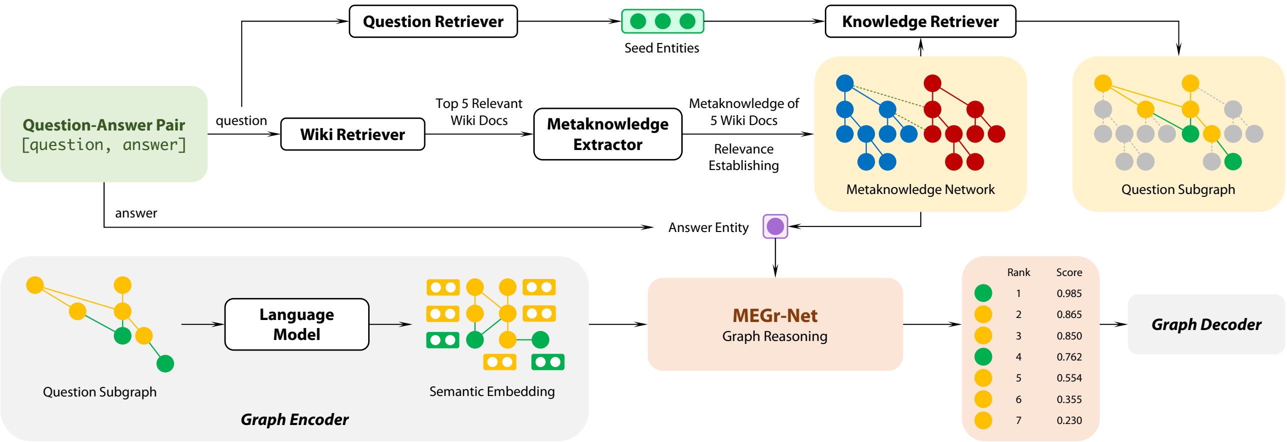

Figure 1. An overview of our approach. There are basically four steps for our approach: (1) generating metaknowledge; (2) building a metaknowledge network; (3) graph modeling and encoding; and (4) graph reasoning.

Essentially, metaknowledge is a special type of hierarchical graph; it generally has two different types of vertices and edges: (1) Hierarchical vertices and edges. The hierarchical vertices include multiple levels of section titles in documents, denoted as V_H, and the hierarchical edges represent a special relation *Hierarchical Belonging*, denoted as E_H. (2) Semantic vertices and edges, which are actually the entities and relations extracted from documents, denoted as V_S and E_S.

Thus, the metaknowledge extracted from document i is denoted as:

$$M_i = \{V_{Hi} \cup V_{Si}, E_{Hi} \cup E_{Si}\}. \tag{1}$$

Meanwhile,

$$V_{Hi}^{L} \xrightarrow{E_{Hi}} V_{Hi}^{L-1}, V_{Si}^{L} \xrightarrow{E_{Hi}} V_{Hi}^{L}, v_{Sij}^{L} \xrightarrow{E_{Si}} v_{Sik}^{L}, \tag{2}$$

where $\xrightarrow{E_{Hi}} V_{Hi}$ denotes the *hierarchical belonging* relation in the document structure, L denotes the hierarchical level of the vertices, $v_{Sij}^{L}, v_{Sik}^{L} \in V_{Si}^{L}$.

3.1. Generating Metaknowledge

Giving a question q described in natural language, this work uses a Wiki retriever proposed in DrQA [21] to get the top 5 relevant Wiki documents $\mathcal{D}_q = \{D_1, \cdots, D_5\}$. For each document D_i, we use open source NLP models to extract the entities and relations (referred to as metaknowledge semantic elements in this work) in paragraphs.

In this work, we transform the HTML script of each Wiki document web page into hierarchical XML files by parsing the HTML labels, such as `<h1>, <h2>, <h3>, <div id="toc"...>, <p>`, which represent the title, section titles, summary, or paragraphs (referred to as metaknowledge hierarchical elements in Ref. [7]).

Suppose Wiki document $D_i = \{P_i, C_i\}$, where $P_i = \left\{p_{i1}, p_{i2}, \cdots, p_{i|P_i|}\right\}$ denotes the paragraphs set in the document D_i ($|P_i|$ is the total number of paragraphs); $C_i = \left\{c_{i1}, c_{i2}, \cdots, c_{i|C_i|}\right\}$ denotes the hierarchical elements, then each paragraph $p_{ij}(j \in |P_i|)$ hierarchically belongs to their upper hierarchical elements $c_{ik}(k \in |C_i|)$ (e.g., section titles). Furthermore, this work extracts entities and relations paragraph by paragraph using Stanza [30] and OpenNRE [31], then links the metaknowledge semantic elements to hierarchical elements with a document structure (Figure 2).

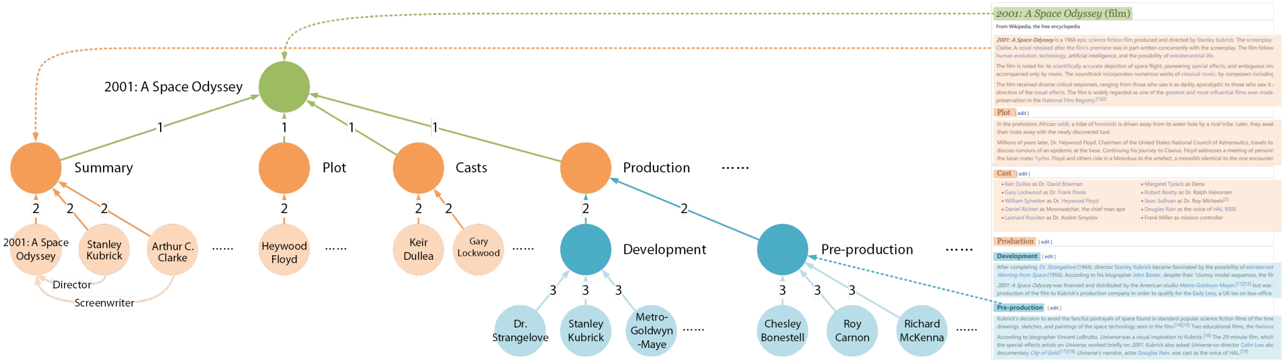

Figure 2. An example of metaknowledge extracted from Wiki documents. The number on the arrows indicates the hierarchical levels of the relations, which are also the weight of edges.

Each document metaknowledge is saved as a JSON file converted from Python dictionary (denoted as metaknowledge dictionary), the data structure is shown in Listing 1.

Listing 1. Denotations of keys in metaknowledge dictionary.

```
Dict:{
|-"Entities":{
|   |-"0":{
|   |   |-"ENT_ID":'ENT_1'
|   |   |-"type":'title'
|   |   |-"content":'2001: A Space Odyssey'
|   |   |-"weight":-1
|   |   |-"title":'2001: A Space Odyssey'
|   |   |-"up_ID":'ENT_1'}
|   |- ...
|   |-"30":{
|       |-"ENT_ID":'ENT_31'
|       |-"type":'PERSON'
|       |-"content":'Stanley Kubrick'
|       |-"weight":2
|       |-"title":'2001: A Space Odyssey'
|       |-"up_ID":'ENT_25'}}
|
|-"Relations":{
    |-"0":{
    |   |-"REL_ID":'REL_1'
    |   |-"head_ID":'ENT_5'
    |   |-"tail_ID":'ENT_1'
    |   |-"type":'Hierarchical Belongs'
    |   |-"weight":-1}
    |- ...
    |-"16":{
        |-"REL_ID":'REL_17'
        |-"head_ID":'ENT_25'
        |-"tail_ID":'ENT_18'
        |-"type":'performer'
        |-"weight":2}}}
```

The weights of hierarchical vertices and edges are set as negative, which are the opposite number of their level, for instance, weights of the 1st-level hierarchical vertices are −1, and the 2nd-level vertices are −2. In contrast, the semantic vertices and edges are set as positive, which are the exact level of the hierarchical vertices they belong to.

The denotations of keys in metaknowledge dictionary are shown in Table 1.

Table 1. Denotations of keys in metaknowledge dictionary.

Entities		Relations	
`ENT_ID`	Entity ID	`REL_ID`	Relation ID
`type`	Entity Type	`type`	Relation Type
`content`	Entity Textual Content	`head_ID`	Head Entity ID
`weight`	Entity Weight	`tail_ID`	Tail Entity ID
`title`	Document Title	`weight`	Relation Weight
`up_id`	Upper Hierarchical Entity ID		

3.2. Building Metaknowledge Network

For documents $\mathcal{D} = \{D_1, D_2, \cdots, D_N\}$, the semantic association between D_i and D_j is denoted as $\mathcal{R}_{ij}$, then the metaknowledge network built on $\mathcal{D}$ is denoted as $\mathcal{N} = \cup_{i,j \in N} \left\{ M_i \overset{\mathcal{R}_{ij}}{\longleftrightarrow} M_j \right\}$. When building a metaknowledge network from document metaknowledge, to avoid the loss of hierarchy caused by semantic entity fusion, this work only establishes semantic association between hierarchical vertices.

Supposing that $v_{H1} \in M_1, v_{H2} \in M_2$ are two semantically associated hierarchical vertices in document metaknowledge M_1 and M_2, their textual embedding vectors are:

$$emb_{H1} = LM(text_{v_{H1}} + text_{title1}), emb_{H2} = LM(text_{v_{H2}} + text_{title2}). \tag{3}$$

where $LM(\cdot)$ denotes the pre-trained language models (PLMs), such as BERT [32], RoBERTa [33], XLNet [34], etc.

Then, we use cosine similarity to calculate the semantic association between two hierarchical vertices:

$$\textbf{IF} \quad \text{cosinesim}(emb_{H1}, emb_{H2}) \geqslant tolerance, \quad \textbf{THEN} \quad v_{H1} \overset{r_{H1H2}}{\longleftrightarrow} v_{H2}. \tag{4}$$

To decide the appropriate tolerance threshold, we uses BERT as the PLM. Taking "South Africa" as the keyword, this work retrieves 10 relevant Wiki documents using Wikipedia Search. Through hand-craft selection, 78 groups of associated metaknowledge hierarchical vertices are picked up. The cosine similarity of semantic embedding in each group is encoded by BERT. Then, the statistical results of this test are shown in Figure 3.

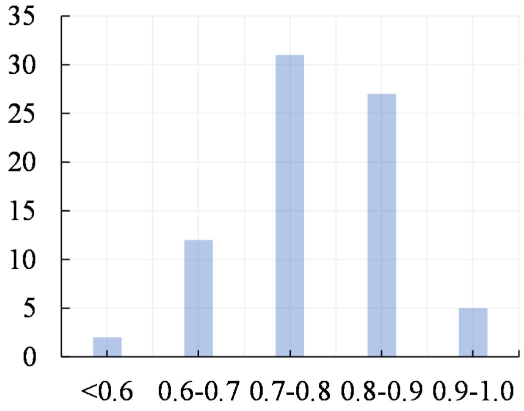

Figure 3. Test to decide metaknowledge association tolerance.

Figure 3 indicates that the cosine similarity between associated hierarchical vertices is basically in the range of $[0.7, 0.9]$; therefore, this work adopts $tolerance = 0.7$.

Meanwhile, we use S-MART (https://github.com/kkteru/r-gcn (accessed on 9 December 2021)) to obtain relevant entities to q, called seed entities $\mathcal{S}_q = \left\{ s_1, \cdots, s_{|\mathcal{S}_q|} \right\}$. Then, a retrieval starts in order to find the directly connected semantic vertices $V_{\mathcal{S}q}$ and hierarchical vertices V_{Hq}, and the latter extends to the top level hierarchical vertex (see also Knowledge Retriever in Figure 1). Then, we get the question subgraph $\mathcal{G}_q$:

$$\begin{aligned} &\mathcal{G}_q = \{\mathcal{V}_q, \mathcal{R}_q\}, \mathcal{V}_q = \{\mathcal{S}_q, V_{Sq}, V_{Hq}\}, \\ &s_i \xleftrightarrow{r_{Sij}} v_{Sj}, s_i^L \xleftrightarrow{r_{Hij}} v_{Hj}^L \xleftrightarrow{r_{Hjk}} v_{Hk}^{L-1} \leftarrow \cdots \rightarrow v_{H0}^0, s_i \in \mathcal{S}_q, v_{hj,k} \in V_{Hq}. \end{aligned} \tag{5}$$

3.3. Metaknowledge Encoding

In this work, we propose a Graph Encoder for Metaknowledge (GE4MK) to encode the text-described document metaknowledge. For document metaknowledge $M_i = \{V_{Hi} \cup V_{Si}, E_{Hi} \cup E_{Si}\}$, the features of each vertex $v_j \in V_{Hi} \cup V_{Si}$ could be divided into three parts: (1) the semantic features of v_j itself, including its textual content v_{cj} and its entity type v_{tj}; (2) the hierarchical features of v_j itself, including the semantic features v_{uj} of the upper hierarchical vertex that v_j belongs to, and the title's semantic features t_j; (3) the semantic features $r_{j1}, r_{j2}, \cdots, r_{jk}$ of relations between v_j and its k nearest 1-hop neighboring vertices.

Consequently, the vertex features $\mathbf{h}_j$ of v_j can be described as:

$$\mathbf{h}_j = \left[f_s(v_{cj}, v_{uj}, t_i) || f_t(v_{tj}) || f_r\Big(r_{j1}, r_{j2}, \cdots, r_{jk}\Big) \right], \tag{6}$$

where

$$v_{cj} = LM(text_{cj}), v_{uj} = LM(text_{uj}), t_i = LM(text_{title}), \tag{7}$$

The output of these PLMs is a λ-dimensional dense semantic vector. The f_s in Equation (3) indicates a 2-layer MLP, which transforms the concatenation of $[v_{cj}, v_{uj}, t_i]$ from $\mathbb{R}^{3\lambda}$ to $\mathbb{R}^{3D}$, D indicates the dimension of feature space which is manually set depending on the using PLM; for instance, in this work, we set $D = 1000$. $f_t : \mathbb{R}^{|\tau|} \rightarrow \mathbb{R}^{D/2}$ and $f_r : \mathbb{R}^{k|\mathcal{R}|} \rightarrow \mathbb{R}^{kD}$ are linear transformations, where $|\tau| = 9$ in Stanza and $|\mathcal{R}| = 80$ in OpenNRE$_{\text{Wiki80}}$.

For the convenience of calculation, we use matrices to describe all the vertices and edges (also the entities and relations) features in the document metaknowledge M_i, so the isolated vertices features (ignoring its neighbors) are:

$$\mathbf{V}_i = \begin{bmatrix} v_{ic1} & v_{iu1} & t_i \\ v_{ic2} & v_{iu2} & t_i \\ \vdots & \vdots & \vdots \\ v_{icn} & v_{iun} & t_i \end{bmatrix}, \tag{8}$$

and the type features of vertices are:

$$\mathbf{T}_i = \begin{bmatrix} v_{it1} & v_{it2} & \cdots & v_{itn} \end{bmatrix}^\top. \tag{9}$$

Meanwhile, the relation type features are denoted as:

$$\mathbf{r}_i = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix}, \tag{10}$$

where $r_{xy} = \#RelationType, x, y \in [1, n]$; for vertex (entity) x, if y is one of the k nearest one-hop neighbors, then $\#RelationType$ indicates the type number in $|\mathcal{R}|$ of the relation between vertices x and y; otherwise, $\#RelationType = 0$.

Therefore, for all the vertices $V_i = \{V_{Hi} \cup V_{Si}\}$ in M_i, their features are:

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_n \end{bmatrix}^\top = \text{concat}(f_s(\mathbf{V}_i), f_t(\mathbf{T}_i), f_r(\mathbf{r}_i)), \tag{11}$$

where concat$(\cdot)$ indicates concatenation by column, and f_r indicates a linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^{D/2}$ in Equation (8).

When considering the semantic information in relations, we define the Semantic Relation Matrix of M_i as:

$$\mathbf{R}_i = LM\left(\begin{bmatrix} text(r_{11}) \\ text(r_{12}) \\ \vdots \\ text(r_{nn}) \end{bmatrix}\right)_{n^2 \times D} . \tag{12}$$

Using $\mathbf{A}_i$ to indicate the adjacency matrix of M_i, then:

$$(\mathbf{A}_i)_{n\times n} \quad (\mathbf{H}_i)_{n\times 4D} \quad (\mathbf{R}_i)_{n^2\times D} = GENC(M_i), \tag{13}$$

where $GENC(\cdot)$ denotes GE4MK, and $\mathbf{R}_i$ only includes semantic relation, not hierarchical relations. Then, we use GE4MK to encode $\mathcal{G}_q$ from text-described data to matrices:

$$(\mathbf{A}_q)_{n\times n} \quad (\mathbf{H}_q)_{n\times 4D} \quad (\mathbf{R}_q)_{n^2\times D} = GENC(\mathcal{G}_q), \tag{14}$$

3.4. Graph Reasoning: MEGr-Net

Inspired by R-GCN[rgcn] and GAT[gat], according to the complex semantic and hierarchical relations, this work proposes a graph-attention-based model MEG-Net (Meta-knoledge Enhanced Graph reasoning Network) in order to perform reasoning on the question subgraph $\mathcal{G}_q$ (Figure 4).

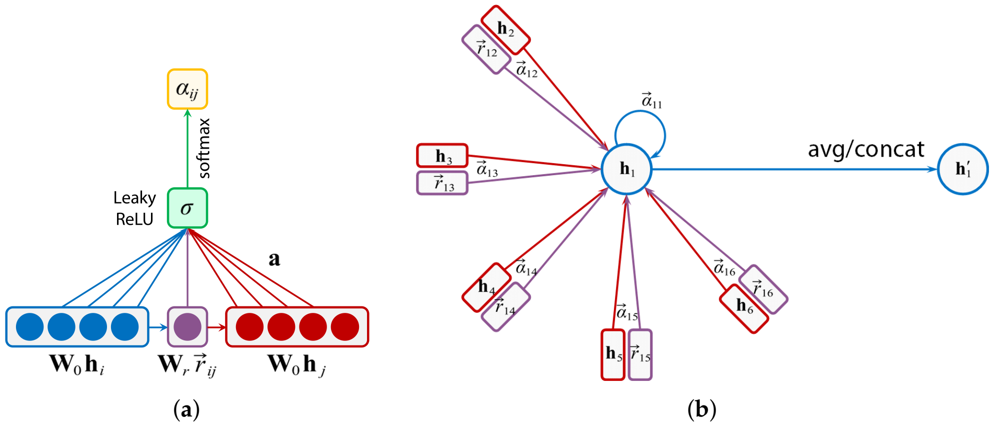

Figure 4. The attention mechanism of MEGr-Net. (**a**) Self-attention; (**b**) attention aggregation.

Relational Graph Attention Layer (R-GAL) is the basic part of MEGr-Net, and the output is the vertex state features under k-heads attention influence. We denote the total number of vertices as $N = |\mathcal{V}_q|$, the vertex features input to R-GAL as $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N\}, \mathbf{h}_i \in \mathbb{R}^F$ (F is the dimension of vertex state space), and the relations as $\mathbf{R} = [\vec{r}_{11}, \vec{r}_{12}, \cdots, \vec{r}_{1N}, \cdots, \vec{r}_{N1}, \cdots, \vec{r}_{NN}]_{N^2\times F_r}, \vec{r}_{ij} \in \mathbb{R}^{F_r}$ (F_r is the dimension of edge state space).

We firstly consider the interaction between vertex v_i and its $k-$neighbors (attention heads). The semantic relation matrix $\mathbf{R}_q$ from $GENC(\mathcal{G}_q)$ is transformed into relation features matrix $\mathbf{R}_k$:

$$\mathbf{R}_k = del\left(\begin{bmatrix} \vec{r}_{11} & \vec{r}_{12} & \cdots & \vec{r}_{1N} \\ \vec{r}_{21} & \vec{r}_{22} & \cdots & \vec{r}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{r}_{N1} & \vec{r}_{N2} & \cdots & \vec{r}_{NN} \end{bmatrix}_{N\times NF_r}\right) = \begin{bmatrix} \vec{r}_{1n_1^1} & \cdots & \vec{r}_{1n_1^k} \\ \vdots & \ddots & \vdots \\ \vec{r}_{Nn_N^1} & \cdots & \vec{r}_{Nn_N^k} \end{bmatrix} = \left(\vec{r}_{i\mathcal{K}_i}\right)_{N\times kF_r}, \tag{15}$$

where $del(\cdot)$ indicates deleting all the empty relations, $\mathcal{K}_i = \left\{n_i^1, \cdots, n_i^k\right\}$ indicate the $k-$neighbors of v_i.

The attention mechanism is denoted as $att : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$; then, we calculate the attention coefficients:

$$\hat{\alpha}_{ij} = att\big(\mathbf{W}_0\mathbf{h}_i, \mathbf{W}_0\mathbf{h}_j\big) + att\Big(\mathbf{W}_r\vec{r}_{i\mathcal{K}_i}, \mathbf{W}_r\vec{r}_{j\mathcal{K}_j}\Big), \tag{16}$$

where $\mathbf{W}_0 \in \mathbb{R}^{F\times F'}$ is the *vertex weight matrix*, and $\mathbf{W}_r \in \mathbb{R}^{F_r\times F'}$ is the *edge weight matrix*. These two matrices realize the parallel computation of linear transformation on each vertex. $\hat{\alpha}_{ij}$ indicates the interaction of the relation between vertex v_i and its neighbor v_j, as well as itself (self-attention). In MEGr-Net, the masked-attention mechanism is used to distribute the attention interaction to the $k-$neighbors $Ne(i)$ of v_i, so the masked-attention coefficient is:

$$\alpha_{ij} = \text{softmax}(\hat{\alpha}_{ij}) = \frac{\exp\,(\hat{\alpha}_{ij})}{\sum_{k\in Ne(i)} \exp\,(\hat{\alpha}_{ik})}. \tag{17}$$

The MEGr-Net sets the attention mechanism as a single-layer feed forward network (FFN) with parameters $\mathbf{a} \in \mathbb{R}^{2F'}$ and LeakyReLU activate function, then:

$$\begin{aligned}\alpha_{ij} &= \frac{\exp\,(\text{FFN}(\hat{\alpha}_{ij}))}{\sum_{k\in Ne(i)} \exp\,(\text{FFN}(\hat{\alpha}_{ik}))} \\ &= \frac{\exp\,(\text{LeakyReLU}(\mathbf{a}^\top \left[\mathbf{W}_0\mathbf{h}_i, \mathbf{W}_0\mathbf{h}_j\right] + \mathbf{a}^\top \Big[\mathbf{W}_r\vec{r}_{i\mathcal{K}_i}, \mathbf{W}_r\vec{r}_{j\mathcal{K}_j}\Big]))}{\sum_{k\in Ne(i)} \exp\,(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}_0\mathbf{h}_i, \mathbf{W}_0\mathbf{h}_k] + \mathbf{a}^\top\left[\mathbf{W}_r\vec{r}_{i\mathcal{K}_i}, \mathbf{W}_r\vec{r}_{k\mathcal{K}_k}\right]))}.\end{aligned} \tag{18}$$

Next, updating the vertex features of v_i:

$$\mathbf{h}_i' = \sigma\left(\sum_{j\in Ne\{i\}} \alpha_{ij}\big(\mathbf{W}_0\mathbf{h}_j + \mathbf{W}_r\vec{r}_{ij}\big)\right), \tag{19}$$

where $\sigma(\cdot)$is a nonlinear function, and we use the ELU in MEGr-Net.

When considering the multi-heads attention, we have:

$$\mathbf{h}_i' = \mathop{\Big|\Big|}_{k=1}^{K} \sigma\left(\sum_{j\in Ne\{i\}} \alpha_{ij}^k\Big(\mathbf{W}_0^k\mathbf{h}_j + \mathbf{W}_r^k\vec{r}_{ij}\Big)\right), \tag{20}$$

where $\mathop{\Big|\Big|}_{k=1}^{K}$ indicates the concatenation of k vertex features of v_i under its k-neighbors attention interaction.

In the last R-GAL, we calculate the average features instead of concatenating, and use the logistic sigmoid to normalize the output features into $[0,1]$ as the probability p_i that indicates vertex v_i is the answer entity:

$$\mathbf{h}_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in Ne\{i\}} \alpha_{ij}^k\Big(\mathbf{W}_0^k\mathbf{h}_j + \mathbf{W}_r^k\vec{r}_{ij}\Big)\right), \tag{21}$$

$$p_i = \text{sigmoid}\Big(\mathbf{h}_i'\Big). \tag{22}$$

For the efficiency of computation, we use matrices in MEGr-Net to describe the whole progress: First, the vertex features matrix $\mathbf{H}$ of question subgraph $\mathcal{G}_q$ is multiplied by the vertex weight matrix $\mathbf{W}_0$ for state space transformation. Then, an all-combination is used to concatenate $\mathbf{W}_0\mathbf{h}_i$ and $\mathbf{W}_0\mathbf{h}_j$ in Equation (16):

$$allc(\mathbf{W}_0\mathbf{H}) = (w_0 h_i)_{N^2 \times F} = \begin{bmatrix} w_0h_1 & \cdots & w_0h_1 & \cdots & w_0h_N & \cdots & w_0h_N \\ w_0h_1 & \cdots & w_0h_N & \cdots & w_0h_1 & \cdots & w_0h_N \end{bmatrix}^\top. \tag{23}$$

We do the same operation to $\mathbf{R}_k$:

$$\begin{aligned} allc(\mathbf{W}_r\mathbf{R}_k) &= \left(w_r\vec{r}_{i\mathcal{K}_i}\right)_{N^2 \times F} \\ &= \begin{bmatrix} w_r\vec{r}_{1\mathcal{K}_1} & \cdots & w_r\vec{r}_{1\mathcal{K}_1} & \cdots & w_r\vec{r}_{N\mathcal{K}_N} & \cdots & w_r\vec{r}_{N\mathcal{K}_N} \\ w_r\vec{r}_{1\mathcal{K}_1} & \cdots & w_r\vec{r}_{1\mathcal{K}_N} & \cdots & w_r\vec{r}_{1\mathcal{K}_1} & \cdots & w_r\vec{r}_{N\mathcal{K}_N} \end{bmatrix}^\top. \end{aligned} \tag{24}$$

Then, the attention coefficient vector:

$$\alpha = \operatorname{soft\,max}\Big(\mathrm{FFN}\Big(\mathbf{a}^\top(allc(\mathbf{W}_0\mathbf{H}) + allc(\mathbf{W}_r\mathbf{R}_k))\Big)\Big). \tag{25}$$

Updating the vertices' state:

$$\mathbf{H}^{'} = \mathop{\Big|\Big|}_{k=1}^{K} \sigma\Big(\alpha^k\Big(\mathbf{W}_0^k\mathbf{H} + \mathbf{W}_r^k\mathbf{R}_k\Big)\Big), \tag{26}$$

and aggregating:

$$\mathbf{H}^{'} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in Ne\{i\}} \alpha^k\Big(\mathbf{W}_0^k\mathbf{H} + \mathbf{W}_r^k\mathbf{R}_k\Big)\right). \tag{27}$$

Finally, the logistic sigmoid:

$$\mathbf{p} = \mathrm{sigmoid}(\mathbf{H}^{'}). \tag{28}$$

The vector $\mathbf{p}$ indicates the probability that each node in the question subgraph is the correct answer. In other words, the MEGr-Net turns question reasoning into a node classification task, and it picks the vertex whose probability is the highest as the most probable answer.

4. Experiments

To verify the effectiveness of metaknowledge network in open domain question answering, this section carries out experiments on a subset of WebQuestionsSP, analyzes the experimental variables including: (1) triplet-based knowledge and metaknowledge, (2) various graph reasoning models, and (3) several pre-trained language models.

4.1. Datasets and Set-Ups

This work uses the open domain natural language question answering dataset WebQuestionsSP [35] for experimental analysis, which includes 4737 questions in natural language. At present, there is no well-established large-scale metaknowledge base and metaknowledge network, so we have to build it from scratch by the approach designed in Sections 3.1 and 3.2. For the fact that the entities and relations extracted by open source NLP models naturally have quality disadvantages, the metaknowledge network we build in this work has an innate weakness when comparing with the finely-built large-scale knowledge bases such as FreeBase and WikiData. Consequently, to make hierarchical metaknowledge and non-hierarchical triplet-based knowledge comparable on the same track, considering the data quality limitation, we adopt the general approach in the construction of knowledge graph, that is, deleting all hierarchical nodes and relationships, retaining only semantic entities and relations in metaknowledge and integrating them to form a non-hierarchical triplet-based knowledge network.

Meanwhile, the process of extracting metaknowledge from Wiki documents, constructing a metaknowledge network, retrieving and encoding question subgraphs takes a big amount of time and computing resources. For example, in the previous experiment, it took an average of 2 h for 4 × 11 GB VRAM GPU and 2 × 12 Core, 24 Threads CPU to build a metaknowledge network from five Wiki documents relevant to a question and complete subgraph retrieval and its encoding. Considering the data quality and hardware, this section scaled down the dataset to 2.5% of WebQuestionsSP, that is, 250 questions in natural language. In addition, it was divided into 150 for the training set, 50 for the cross validation set and 50 for the test set. In this section, it is referred to as WebQuestions$_{MbQA}$. The training parameters of MEGr-Net are shown in Table 2.

Table 2. The training parameters of MEGr-Net.

Parameters	Values
Epochs	200
Learning Rate	5×10^{-3}
Attention Heads k	8
Dimension of Entity Features F'	1000
Dimension of Relation Features F'_r	500
Hidden Units	1000

The semantic encoder $LM(\cdot)$ is deployed on Server #1. The metaknowledge generation, metaknowledge network construction framework and MEGr-Net are deployed on Server #2 (see also Appendix A). A Tesla V100 GPU (with 32 GB VRAM) is used for training, which takes 13.5 d (325 h).

This work takes the average accuracy (avg. Acc.) as the evaluation index.

4.2. Experimental Control Groups

This section analyzes the impact of different experimental variables on MbQA from the following three aspects:

- **Hierarchical metaknowledge and non-hierarchical triplet-based knowledge.** This is the focus of this section, that is, what improvement hierarchical metaknowledge can make on open domain question answering compared with non-hierarchical triplet-based knowledge—in other words, whether metaknowledge and metaknowledge network have superiority in open domain QA tasks. As described in Section 3.1, considering the extraction quality of open domain entities and relationships by open source NLP models, this section uses the same data and extraction models to build a metaknowledge network (referred to as MK-Net in the experiment) and triplet knowledge base (referred to as Tri-KB) by the metaknowledge structure proposed in the beginning of Section 3 and the general triplet-based knowledge structure, respectively.
- **Graph reasoning model.** MEGr-Net, based on GAT, essentially achieves an improvement of graph data with complex relationships, like metaknowledge. Meanwhile, it partially adopts the relationship processing approach in R-GCN. Therefore, this section takes GAT and R-GCN as test baselines and compares them with MEGr-Net. To explain the impact of (meta)knowledge extraction quality on the results, this section introduces the results of DrQA [21] and GRAFT-Net [22] on the entire WebQuestionsSP as a reference.
- **Pre-trained language models (PLMs).** The input of MEGr-Net is the question subgraph $\mathcal{G}_q$ encoded by GE4MK, and its semantic features mainly come from the text embedding vector encoded by the PLM $LM(\cdot)$ in GE4MK. Therefore, different PLMs may exert different impact on the semantic feature richness of the problem subgraph. This section takes BERT$_{BASE}$ as the baseline and RoBERTa [33] and ALBERT [36] as the control groups.

4.3. Results and Analysis

The results on control group #1 on WebQuestions$_{MbQA}$ are shown in Table 3. The results show that, with the same data quality, the hierarchical metaknowledge achieves better results than non-hierarchical triplet knowledge in open domain question answering (+16.9% Tri-KB).

Table 3. Results on Control Group #1.

(Meta) Knowledge Network	IH-Acc #.
Tri-KB	0.483
MK-Net	**0.652**

# The WebQuestions$_{MbQA}$ dataset which we use in the experiment is part of the whole WebquestionsSP, so we use average In-House Accuracy (IH-Acc.) for avg. Acc.

The results on control group #2 are shown in Table 4. For GAT, the relationship matrix $\mathbf{R}_k$ in MEGr-Net and the relationship weight matrix $\mathbf{W}_r$ in R-GAL are removed in this section. Modifications have been made to R-GCN for the tasks in this section.

Table 4. Results on Control Group #2.

Graph Reasoning Models		Acc.
Baselines	GAT (MK-Net)	0.608 (IH †)
	R-GCN # (MK-Net)	0.601 (IH)
	MEGr-Net (MK-Net)	0.652 (IH)
	DrQA * (doc only)	0.215
	GRAFT-Net * (KB+doc)	0.687

# Model from https://github.com/kkteru/r-gcn (accessed on 9 December 2021). * The data are cited from [22], respectfully. † IH indicates that the experiments are based on the dataset WebQuestions$_{MbQA}$, and if not annotated that indicates that the experiments are based on the dataset WebQuestionsSP.

As can be seen from the results, MEGr-Net achieves better performance than the baselines in the reasoning of hierarchical graph data with complex semantic relationships, such as a metaknowledge network (+4.4%GAT, +5.1%R-GCN). Meanwhile, compared with GRAFT-Net, which uses the complete FreeBase as the knowledge base and integrates the document (doc) and KB features, MEGr-Net still lags behind, indicating that it still needs to be improved in MbQA, especially in the integration with MRC method (see also Section 4).

The results on control group #3 are shown in Table 5 (see Appendix B for the source of the pre-training parameter file of the pre-training language model). From the results, the PLMs (ALBERT$_{XXLAERGE}$, RoBERTa$_{LARGE}$) with large-scale parameters perform better, indicating that the larger the PLMS used by the graph encoder, the finer the fine tuning and the richer the semantic features of the question subgraph, the better performance will be achieved in MbQA.

Table 5. Results on Control Group #3.

MEGr-Net	PLMs	IH-Acc.
Baseline	+BERT$_{BASE}$	0.652
	+ALBERT$_{BASE}$	0.646
	+BERT$_{LARGE}$	0.670
	+RoBERTa$_{LARGE}$	0.692
	+ALBERT$_{XXLARGE}$	**0.708**

As shown in Figure 5, the combination of MEGr-Net and ALBERTXXLARGE achieved the best results (+5.6% MEGr-Net+BERT$_{BASE}$) and gained better performance than GRAFT-Net using LSTM [lstm] as a text encoder, which proves that PLMS based on Transformers [28] is better than LSTM in MbQA.

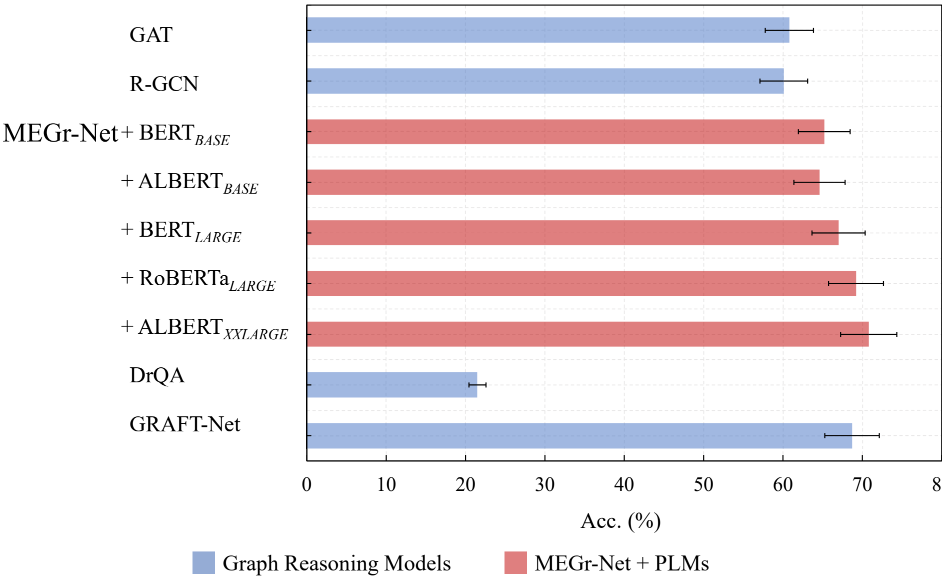

Figure 5. Results of graph reasoning models and PLMs in MbQA.

Generally, the metaknowledge network with document directory hierarchy can significantly improve the existing methods in KBQA, which is basically consistent with the view that titles play a positive role in question answering in [22]. Meanwhile, finer PLMs can improve the semantic feature representations of question subgraphs and achieve better results in question answering. This is also consistent with the view and experimental results in [37].

5. Discussion

From the overall results of this work, metaknowledge basically solves the problems of triplet-based knowledge with weak structural logic, and provides a new idea for the theoretical and practical research of knowledge engineering. Meanwhile, it must also be noted that, as a relatively new research field, there are still some urgent problems that need to be solved in the future work.

5.1. Metaknowledge and Metaknowledge Network Modeling

The metaknowledge and metaknowledge network modeled by the single dimension network in this work (Sections 3.2 and 3.3) is a compromising strategy to reduce the complexity of the model under the current realistic conditions of mainstream GNN models. In fact, according to the our concept, the metaknowledge network should be a multi-dimensional hyper-graph with hierarchical structure (Figure 6). The metaknowledge network expressed by that type of graph model includes two dimensions: hierarchical dimension and semantic dimension. The hierarchical dimension is in the outer layer, which includes all hierarchical nodes and relationships; the semantic dimension is in the inner layer, which includes all semantic nodes and relationships subordinate to the hierarchical nodes. Ref. [38] proposes an embedding framework MINES for multi-dimensional networks with hierarchical structure, which uses a hierarchical structure for multi-dimensional network embedding; Ref. [37] proposes an open domain question answering method based on hyper-edge fusion. These documents show the feasibility of graph reasoning on the metaknowledge network expressed by the hierarchical multi-dimensional hyper-graph. This metaknowledge modeling method needs to be further studied and explored.

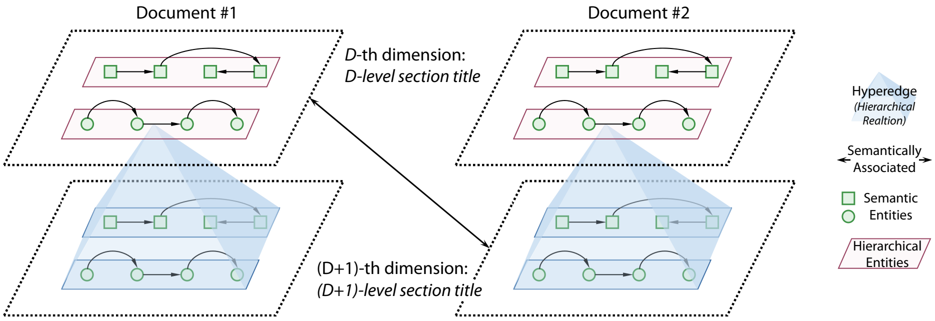

Figure 6. Metaknowledge that modeled by multi-dimensional hyper-graph with hierarchical structure.

5.2. MbQA and Graph Reasoning

Limited by the extraction effect of open-source NLP models, MbQA has insufficient advantages over KBQA. At least under the existing conditions, there is still a huge gap between the metaknowledge network and mutual large-scale knowledge bases such as freebase from the perspective of data quality. Therefore, MbQA may be more suitable for in-domain QA tasks (such as question answering on laws and regulations). The fine-tuned NLP models will significantly improve the extraction quality of metaknowledge semantic elements. At the same time, the structural logic of metaknowledge network makes it have the ability to deal with complex relationships. Therefore, the role of the metaknowledge network in multi-hop QA tasks is also a direction worthy of research. In terms of graph reasoning models for question answering tasks, MEGr-net relies on the hierarchical features contained in metaknowledge to supplement the short board relying only on semantic features (KBQA). On this basis, documents [22] and pre-trained language models [39] can continue to be integrated into graph reasoning to select the best from the best and enhance the effect of MbQA.

In general, the metaknowledge enhanced question answering is a brand-new method for solving the problem caused by triplet-based knowledge, and it improves the capability of current knowledge bases (which are also the knowledge engines of intelligent voice interaction devices). In the foreseeable future, this method could be the antidote to help intelligent voice devices get rid of the problems that they are not so good when answering complex questions asked by users, and make great progress in the interaction with human users.

6. Conclusions

Facing the problems in current open domain QA tasks caused by the loose knowledge association and weak structural logic of triplet-based knowledge, this work makes pivotal innovations on metaknowledge enhance question answering: (1) Metaknowledge extraction and metaknowledge network construction, where we present the approach of generating metaknowledge and building metaknowledge network from Wiki documents automatically. (2) Metaknowledge and metaknowledge network modeling, where we generally consider several different kinds of features from reasoning performance related aspects including semantic features such as textual content, entity type, relations, and along with hierarchical features. (3) MEGr-Net, which is proposed for question answering, which aggregates both relational and neighboring interactions compared with R-GCN and GAT. Experiments have proved the improvement of metaknowledge over main-stream triplet-based knowledge. We have found that the graph reasoning models and pre-trained language models also have influences on the metaknowledge enhanced question answering approaches.

Author Contributions: Conceptualization and methodology: S.L. and R.X.; data gathering and processing: R.X.; Experiments designing and anakysis: R.X.; research on the related work: S.L., R.X. and M.L.; writing (original draft): R.X., S.L. and Y.L.; writing (review and editing): S.L., R.X., Y.L. and L.D.; supervision: L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Experimental Environments of Hardware and Software

Table A1. Hardware and software environments of Server#1.

Server #1: Providing BERT Embedding Service		
Hard-ware Env.	CPU	2 × Intel Xeon E5-2678 v3 (48) @ 3.300 GHz
	RAM	32 GB
	GPU	4 × NVIDIA GV102 (11 GB VRAM)
Software Env.	OS	Ubuntu 18.04.5 LTS
	Python	Python 3.6.5: Anaconda
	PyTorch	1.6.0 (for GPU)
	TensorFlow	1.15.0 (for GPU)

Table A2. Hardware and software environments of Server#2.

Server #2: Main Experimental Environment		
Hard-ware Env.	CPU	2 × Intel Xeon Silver 4210R (40) @ 3.200 GHz
	RAM	256 GB
	GPU	4 × NVIDIA Tesla V100S (32 GB VRAM, using 1)
Software Env.	OS	Ubuntu 20.04.2 LTS
	Python	Python 3.7.7: Anaconda
	PyTorch	1.9.0 (for GPU)
	TensorFlow	1.15.0 (for GPU)

Appendix B. PLMs Used in This Work

1. BERT$_{BASE}$: https://huggingface.co/bert-base-uncased/tree/main (accessed on 9 December 2021).
2. BERT$_{LARGE}$: https:///huggingface.co/bert-large-uncased/tree/main (accessed on 9 December 2021).
3. RoBERTa$_{LARGE}$: https://huggingface.co/roberta-large/tree/main (accessed on 9 December 2021).
4. ALBERT$_{BASE}$: https://huggingface.co/albert-base-v2/tree/main (accessed on 9 December 2021).
5. ALBERT$_{XXLARGE}$: https://huggingface.co/albert-xxlarge-v2/tree/main (accessed on 9 December 2021).

References

1. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A nucleus for a web of open data. In Proceedings of the 6th International Semantic Web and 2nd Asian Semantic Web Conference, Busan, Korea, 11–15 November 2007; Volume 4825, pp. 722–735.
2. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
3. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
4. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J.R. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 21–26 August 2021; Volume 5, pp. 4483–4491.

5. Zhang, N.; Jia, Q.; Deng, S.; Chen, X.; Ye, H.; Chen, H.; Tou, H.; Huang, G.; Wang, Z.; Hua, N.; et al. AliCG: Fine-grained and Evolvable Conceptual Graph Construction for Semantic Search at Alibaba. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 3895–3905.
6. Li, Z.; Ding, X.; Liu, T. Constructing Narrative Event Evolutionary Graph for Script Event Prediction. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4201–4207.
7. Liu, S.K.; Xu, R.L.; Geng, B.Y.; Sun, Q.; Duan, L.; Liu, Y.M. Metaknowledge Extraction Based on Multi-Modal Documents. *IEEE Access* **2021**, *9*, 50050–50060. [CrossRef]
8. Evans, J.A.; Foster, J.G. Metaknowledge. *Science* **2011**, *331*, 721–725. [CrossRef] [PubMed]
9. Wu, P.; Zhang, X.; Feng, Z. A Survey of Question Answering over Knowledge Base. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Hangzhou, China, 24–27 August 2019; pp. 86–97.
10. Tunstall-Pedoe, W. True Knowledge: Open-Domain Question Answering Using Structured Knowledge and Inference. *AI Mag.* **2010**, *31*, 80–92. [CrossRef]
11. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1533–1544.
12. Yih, S.W.; Chang, M.W.; He, X.; Gao, J. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; Volume 1, pp. 1321–1331.
13. Dong, L.; Lapata, M. Language to Logical Form with Neural Attention. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Volume 1, pp. 33–43.
14. Xu, K.; Wu, L.; Wang, Z.; Yu, M.; Chen, L.; Sheinin, V. Exploiting Rich Syntactic Information for Semantic Parsing with Graph-to-Sequence Model. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 Octobe–4 November 2018; pp. 918–924.
15. Liang, C.; Berant, J.; Le, Q.V.; Forbus, K.D.; Lao, N. Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 23–33.
16. Qiu, Y.; Zhang, K.; Wang, Y.; Jin, X.; Bai, L.; Guan, S.; Cheng, X. Hierarchical Query Graph Generation for Complex Question Answering over Knowledge Graph. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1285–1294.
17. Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. Large-scale Simple Question Answering with Memory Networks. *arXiv* **2015**, arXiv:1506.02075.
18. Dong, L.; Wei, F.; Zhou, M.; Xu, K. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; Volume 1, pp. 260–269.
19. Jain, S. Question Answering over Knowledge Base using Factual Memory Networks. In Proceedings of the NAACL Student Research Workshop, San Diego, CA, USA, 12–17 June 2016; pp. 109–115.
20. Chen, Z.Y.; Chang, C.H.; Chen, Y.P.; Nayak, J.; Ku, L.W. UHop: An Unrestricted-Hop Relation Extraction Framework for Knowledge-Based Question Answering. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 345–356.
21. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1870–1879.
22. Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; Cohen, W.W. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 Octobe–4 November 2018; pp. 4231–4242.
23. Fang, Y.; Sun, S.; Gan, Z.; Pillai, R.; Wang, S.; Liu, J. Hierarchical Graph Network for Multi-hop Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual Event, 16–20 November 2020; pp. 8823–8838.
24. Li, Q.; Tang, X.; Jian, Y. Adversarial Learning with Bidirectional Attention for Visual Question Answering. *Sensors* **2021**, *21*, 7164. [CrossRef] [PubMed]
25. Guo, Z.; Han, D. Multi-Modal Explicit Sparse Attention Networks for Visual Question Answering. *Sensors* **2020**, *20*, 6758. [CrossRef] [PubMed]
26. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
27. Schlichtkrull, M.S.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the 15th International Conference on Extended Semantic Web Conference, ESWC 2018, 2018, pp. 593–607.

28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 5998–6008.
29. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
30. Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; Manning, C.D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Online Conference, 5–10 July 2020; pp. 101–108.
31. Han, X.; Gao, T.; Yao, Y.; Ye, D.; Liu, Z.; Sun, M. OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Hong Kong, China, 3–7 November 2019; pp. 169–174.
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K.N. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
33. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
34. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.G.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5753–5763.
35. tau Yih, W.; Richardson, M.; Meek, C.; Chang, M.W.; Suh, J. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; Volume 2, pp. 201–206.
36. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In Proceedings of the ICLR 2020: Eighth International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
37. Han, J.; Cheng, B.; Wang, X. Open Domain Question Answering based on Text Enhanced Knowledge Graph with Hyperedge Infusion. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16–20 November 2020; pp. 1475–1481.
38. Ma, Y.; Ren, Z.; Jiang, Z.; Tang, J.; Yin, D. Multi-Dimensional Network Embedding with Hierarchical Structure. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Los Angeles, CA, USA, 5–9 February 2018; pp. 387–395.
39. Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; Leskovec, J. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online Event, 6–11 June 2021; pp. 535–546.