Knowledge-Based Systems 221 (2021) 106954

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Knowledge Graph Question Answering with semantic oriented fusion model

Haobo Xiong [a,1], Shuting Wang [b,1], Mingrong Tang [a], Liping Wang [a,*], Xuemin Lin [a,c]

[a] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[b] School of Computer and Information Engineering, Zhejiang Gongshang University, Zhejiang, China
[c] The University of New South Wales, Sydney, Australia

ARTICLE INFO

Article history:
Received 5 January 2021
Received in revised form 4 March 2021
Accepted 10 March 2021
Available online 16 March 2021

Keywords:
Knowledge graph question answering
Topic entity recognition
Path generation
Model fusion

ABSTRACT

Knowledge Graph Question Answering (KGQA) is a major branch of question answering tasks, which can answer fact questions effectively by using the reasonable characteristics of the knowledge graph. Currently, lots of related works combined with a variety of deep learning models are presented for the KGQA task. However, there are still some challenges, such as topic entity recognition under ambiguity expression, semantic level representation of natural language, efficient construction of searching space for answers, etc. In this paper, we propose a comprehensive approach for complex question answering over KG. Firstly, during the stage of topic entity recognition, a deep transition model is constructed to extract topic entities, and an efficient entity linking strategy is presented, which combines character matching and entity disambiguation model. Secondly, for candidate path ranking, a dynamic candidate path generation algorithm is proposed to efficiently create the candidate answer set. And four dedicated similarity calculation models are designed to handle the intricate condition of complex questions with long sequence and diversity expression. Moreover, a fusion policy is proposed to make decision for the final correct answer. We evaluate our approach on CKBQA, a Chinese knowledge base question answering dataset, from CCKS2019 competition. Experimental results demonstrate that the improvements in each process are effective and our approach achieves better performance than the best team in CCKS2019 competition.

1. Introduction

With emergence of many high-quality open-domain and vertical-domain knowledge graphs, such as Freebase [1], Wikidata [2], YaGo [3], AliCoCo [4], etc., there are a growing number of researches on Knowledge Graph Question Answering (KGQA). The goal of KGQA is to match the satisfied nodes of KG as correct answers for related natural language question, which usually involves a series of intricate processes. On the one hand, KGQA enables users to access knowledge graph by natural language, greatly improving the friendliness of query and the usability of knowledge graph. On the other hand, by introducing the knowledge graph with rich semantic relations and reasonable characteristics, the question answering model can understand the question well, perform certain reasoning and return more correct answer.

So far, there are many works dedicating to the task of KGQA. Here, we summarize the existing methods into three categories including semantic parsing-based methods, information retrieval-based methods and deep learning-based methods. Semantic parsing-based methods [5,6] aim to transform questions into logical forms that can be reasoned and executed in related KG. To apply the logical forms to knowledge graph, specific query languages such as SPARQL [7], Cypher [8], etc. are introduced and query statements are generated according to the semantics. Information retrieval-based methods [9,10] aim to extract topic words from a nature language question and link them to the nodes in the KG. Here, we call these nodes topic entities. After that, the neighbor nodes of topic entities are selected as elements of candidate set, and the entities ranked with the highest score are returned as the final correct answers. With the rapid development of deep learning, more and more researchers introduce deep learning models into the first or the second methods, and name them with deep learning-based methods [11,12]. These methods make effort to learn the transformation from questions and candidate answers to related vector space, which convert the relatively complex KGQA task into a simple sequence generation task, similarity calculation task or classification task. The feature vectors of questions and candidate answers usually can be generated by different ways. At early stage, word embedding

* Corresponding author.
E-mail address: lipingwang@sei.ecnu.edu.cn (L. Wang).
1 Haobo Xiong and Shuting Wang are the joint first authors.

https://doi.org/10.1016/j.knosys.2021.106954
0950-7051/© 2021 Published by Elsevier B.V.

models such as Word2Vec [13] and GloVe [14] are often utilized. Recently, pre-trained language representation models such as GPT [15], BERT [16] and XLNet [17] can achieve better results. In general, the technology of representation learning has been successfully introduced into KGQA task. Recent experimental results show that these methods not only simplify the complexity of models for KGQA, but also improve the correctness. As a result, for simple questions, the current researches have reached an excellent level, but further researches are still needed for complex questions. Here we will summarize the main challenges for complex question answering.

Challenges. According to the research progress of KGQA task, we summarize three main challenges. The first challenge is how to accurately extract the topic entity from a natural language question, which is also a problem facing in natural language processing field. Here topic entity recognition includes two stages: sequence labeling and entity linking. For sequence labeling, the solutions based on the family of Recurrent Neural Network (RNN) dominate other methods and already achieve acceptable performance. However, these methods have an obvious shortcoming that the shallow connections between consecutive hidden states of RNN would limit the accuracy of these models. For entity linking, ambiguous expressions need to be handled. The second one is how to efficiently generate candidate answers. As we all known, the count of candidate answers increases exponentially with the number of hops. Previous works usually limit the hop of answer to a fixed number to control the count of candidates. In different scenarios, the number of hops for the correct answer is difficult to estimate. When the threshold is large, there are lots of redundant computation. Conversely, correct answers may be missed. Therefore, the key to an appreciate candidate answer set construction is to reduce the count of candidate answers without affecting the recall of models. The last one is related to the representation learning with complex questions and different candidates. In practice, we expect that models can not only deal with simple questions, but also have the ability to answer complex questions that often involve multiple entities and multi-hop paths. Compared with the sequential and explicit expression of simple questions, the expression of complex questions is often reversed and implicit, so that it is more difficult to effectively learn their features. Previous works mostly introduce as much answer information as possible, such as subject entity, answer path, answer type, answer context, etc., so as to express the answers and questions comprehensively and semantically. However, it usually does not work for the incomplete knowledge graph, such as missing node type and incomplete node context information.

For the above challenges, we propose the following solutions respectively. Firstly, inspired by the DTMT [18], we utilize the deep transition RNN [19] to build a topic entity recognition model based on BERT, which achieves richer syntactic features by deeply modeling the transition between hidden states. Secondly, inspired by the idea of beam search [20], we propose an algorithm to dynamically generate candidate answers. Different from the traditional work, we prune the irrelative candidates as early as possible, meanwhile can guarantee the recall of answers. Thirdly, four models incorporating BERT are designed to learn feature vectors of questions and candidate answers from multiple focuses. Moreover, we propose a suitable fusion strategy to choose the final answers, which can handle the condition of missing values and make a comprehensive decision from answers recommended by the above four models.

Contributions. We devise a complete architecture for Knowledge Graph Question Answering task. Our contributions involve different stages of KGQA.

(1) To enhance the accuracy of topic entity recognition under ambiguity expression, we utilize the deep transition RNN to build a topic entity recognition model based on BERT and present an effective strategy for entity linking task, which is composed of character matching and deep learning model.
(2) We propose an useful candidate path generation algorithm named DPG to efficiently compute the path similarity. DPG can dynamically expand the candidate path set with the increasing of hop number and prune the irrelevant paths as early as possible.
(3) To handle the intricate condition of complex questions with long sequence and diversity expression, we design four models considering global and local representation of candidate path, diversity of path, and implicit relations in path separately. As a result, the four models can learn feature vectors of Chinese question and path sequences from multiple focuses.
(4) We propose a fusion policy based on semantic similarity to return the final valid answers. It makes an evaluation to determine the most reasonable path from the candidate path sets recommended by above four models. Extensive experiments performed on the CKBQA dataset demonstrate the effectiveness of our approach.

This paper is an extension of our previous work [21]. We extend techniques and devise a complete architecture for KGQA task. The improved model for topic entity recognition is presented and comprehensive experimental results are evaluated. The rest of this paper is organized as follows. In Section 2, the related researches of KGQA are introduced. Section 3 presents the techniques for topic entity recognition and candidate path ranking. In Section 4, extensive experiments for each stage are depicted. Section 5 gives the conclusion of our work.

2. Related work

As an important part of the intelligent interactive system, the research on Knowledge Graph Question Answering is very valuable and attracts the attention of many researchers. Currently, there are lots of related works, which can be divided into three categories including semantic parsing-based methods, information retrieval-based methods and deep learning-based methods.

Semantic parsing-based methods belong to the traditional methods for KGQA. As stated above, the main idea for these methods is to transform questions into a reasonable logical form that can be understood by the knowledge graph through semantic analysis of natural language. Based on this idea, [22] proposes a new semantic representation form named Dependency-based Compositional Semantics (DCS), which can resolve a natural language question into a DSC tree. [22] achieves the highest accuracy on two standard semantic parsing benchmarks (GEO and JOBS) without time-consuming and laborious data annotation. However, the predefined rules need to be modified greatly when the language type changes. Similar to DCS, Category Compositional Grammar (CCG) [5] is another work of the basic semantic parsing grammars. [23] transforms the KGQA task into a subgraph matching task, which performs subgraph matching to obtain the final answer by parsing a question into a semantic query subgraph. In [23], authors propose two different frameworks to construct semantic query graphs. One is relation first, the other is node first, which can solve the problem of ambiguity in the query and matching stages.

The information retrieval-based methods also belong to the traditional methods. These methods obtain the correct answers by extracting the topic entity and performing effective candidate answer ranking. They assume the topic word of questions must correspond to a node named topic entity in the knowledge graph, and the answer of questions must be around the topic entity. [10] first obtains three question features including question type, question focus, and question subject, then extracts topic entity and candidate answer features. Lastly, a classifier combining the question features and answer features is trained. Compared with semantic parsing-based methods, the information retrieval-based methods do not involve too much basic linguistic knowledge, are easy to understand and are helpful to achieve an end-to-end model. But the correctness of these methods still need to be further improved.

With the rapid development and wide application of deep learning technology, researchers introduce it into the KGQA task frequently. The key to deep learning-based methods is learning feature vectors of question and answer effectively. The encoder–decoder model is one of effective ways to complete translation task. [24] develops a grammar-guided decoder to generate possible action sequences. By introducing representation learning technology, a KGQA task can be converted into a similarity computation task which returns a similarity score between a question vector and a candidate answer vector. In order to learn low-dimensional feature vectors of questions and knowledge graph triples in a space, an embedding model is trained in [11]. As a result, the distance between the representation vectors of the question and the answer can be used to evaluate their similarity. By using knowledge graph embedding technology, [25] proposes a model for answering simple questions. For each question, it aims to jointly recover the representations of head entity and tail entity in a vector space. Different from the above methods, [26] constructs graph-structured query statements by knowledge graph embedding technology which solves the mapping problem of entity and relation.

Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are also applied into question answering task. [27] utilizes Multi-column Convolutional Neural Networks (MCCNNs) and learns the distributed vectors of questions and answers from question context, answer type and answer path, which can achieve richer semantic feature vectors than other works. [28] proposes a new lattice-based Convolutional Neural Networks (LCNs). It can use multi-granularity information in the word lattice and can process noise information greatly. [29] proposes an end-to-end model based on a bidirectional Long Short-term Memory Network (Bi-LSTM). Different from previous work, this model can dynamically represents questions according to different aspects of candidate answers via cross-attention. Usually, the input of the LSTM model is a linear sequence. However, considering a question contains rich syntactic features that can be parsed as a tree structure, [30] proposes a long short-term memory network named Tree-LSTM, which can handle tree-type input and is helpful for the analysis of implicit intent.

Our work combines deep learning technology and information retrieval technology, which is a new deep learning-based method for KGQA task. Compared with existing work, we pay more attention to the intricate condition of complex questions with diversity expression and representation of semantic information. Moreover, during answer searching, prune technology is devised to improve the efficiency of method.

3. Our approach

Knowledge Graph (KG) is a structured representation of facts and can be used for interpretation and inference over facts. Considering the structure of KG, it can be represented as a directed graph with nodes as entities and edges as relations. For a given question q expressed by natural language, the Knowledge Graph Question Answering (KGQA) aims to use the KG for reasoning and query, and return an entity set A as correct answers. Fig. 1 gives the overall workflow of our approach for KGQA. The proposed solution involves two stages, topic entity recognition and candidate path ranking. The first stage is designed to complete the tasks of semantic parsing of question and the corresponding topic entity node finding in the KG. Specifically, the stage can be organized in two steps, sequence labeling and entity linking. The second stage consists of three steps which are candidate path generation, similarity calculation and result fusion. Considering the incompleteness of knowledge graph, such as missing node type and incomplete node context information, we use candidate paths to represent candidate answers in second stage. Then we select the correct answers by comparing the semantic similarity between questions and candidate paths.

In this paper, q is introduced as an input question, E_q means the topic entity set of q, and the related candidate path set is denoted by P_q. The answer for q is represented by A. More details about our improved models will be introduced in the following parts.

3.1. Topic entity recognition

In order to obtain the topic entity set E_q corresponding to question q, we first extract mentions in question through sequence labeling, and then link the mentions to some certain entity nodes in the knowledge graph. In Section 3.1.1, we devise an improved sequence labeling model based on BERT and deep transition RNN. In Section 3.1.2, we present an enhanced strategy for entity linking task which includes character matching and the design of binary classifier for contextual semantic parsing.

3.1.1. Sequence labeling

The performance of the current RNN-based sequence labeling model is usually restricted by the shallow connections between consecutive hidden states of RNN. Here, we utilize the bidirectional deep transition RNN to build a model, named BERT-BiDT-CRF and apply the pre-trained language model BERT to extract the character embedding for better performance.

As Fig. 2 shown, the deep transition (DT) cell is composed of one Linear Transformation Enhanced GRU (L-GRU) cell and several Transition GRU (T-GRU) cells [31], by which the transition of hidden states is modeled in depth. For each time step, the token of input embedding is encoded by the L-GRU cell firstly. Then the L-GRU cell passes its hidden state to the T-GRU cell chain, which are connected only through the hidden state. And the output of the chain is used as the "*state*" of the DT cell, which is also taken into the L-GRU of the next time step. At last, we use a Conditional Random Field (CRF) [32] layer to constrain the final output sequence label by learning the global feature information of the sequence.

3.1.2. Entity linking

For entity linking task, the main difficulty is how to link the mention to the corresponding entity node in knowledge graph efficiently after understanding the semantic of the question. Here, we first get candidate entity nodes by character matching [21] including exact matching and fuzzy matching, then we construct a deep learning model to further predicate the correctness of each entity. During exact matching, we use an entity dictionary to accurately match the mentions, and add the exactly matched entity nodes to the topic entity set E_q. The examples of entity dictionary definition are shown in Table 1. During fuzzy matching, we convert the fuzzy matching problem into a retrieval problem. Each mention is used to search in the Elastic Search index based on the entity node of KG. Finally, the satisfied entities are inserted into the topic entity set E_q.

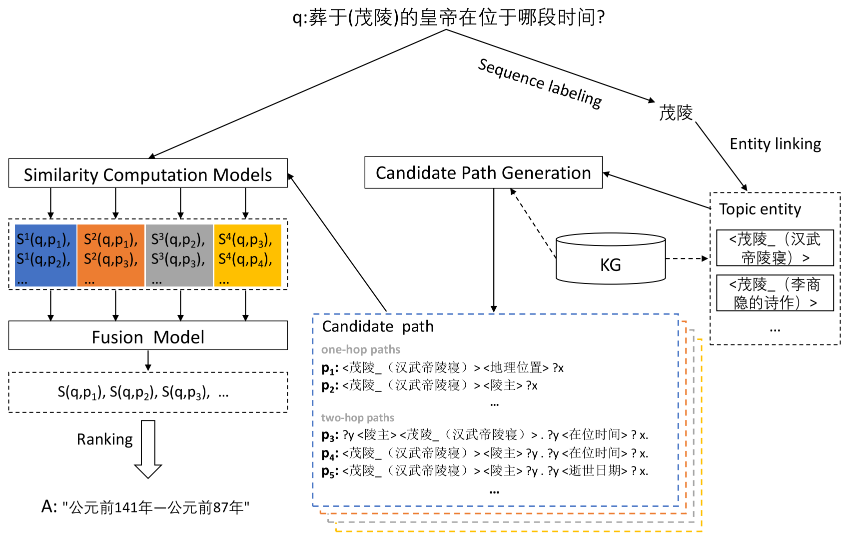


Fig. 1. The Overview of multi-focus fusion model for KGQA task.

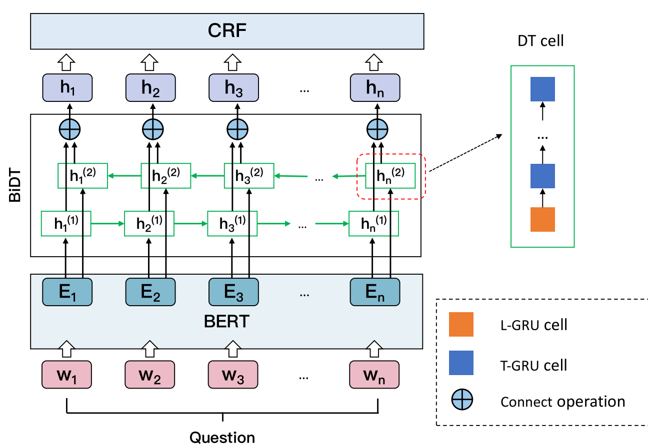


Fig. 2. The architecture of sequence labeling model BERT-BiDT-CRF.

Table 1
The examples of entity linking dictionary.

Mention	Topic entity	Related grade
茂陵	茂陵_（汉武帝陵寝）	1
茂陵	明茂陵	2
茂陵	茂陵_（李商隐的诗作）	3
佛罗伦萨	佛罗伦萨_(意大利托斯卡纳大区首府)	1
佛罗伦萨	佛罗伦萨足球俱乐部	2
佛罗伦萨	佛罗伦萨_(多肉植物名)	3
玫贵人	玫贵妇	1
玫贵人	白蕊姬	2

Since the above character matching does not consider the semantic information of the question context, the entities in set E_q generally include certain ambiguities. For example, according to the dictionary defined in Table 1, the topic entity set E_q obtained by character matching for the question "葬于茂陵的皇帝在位于哪段时间?"[2] is {"<茂陵_（汉武帝陵寝）>", "<茂陵_（李商隐的诗作）> ", "<明茂陵> "}.[3] The set includes obviously unrelated entities, such as "<茂陵_（李商隐的诗作）>".[4]

To solve the above problem, we design a BERT-based binary classifier which can consider the contextual semantics of mentions and perform entity disambiguation on the topic entity set E_q. The input of the classifier is the connection of question q and each entity in set E_q. The output of the classifier is represented by a binary label which means whether the entity is related to the question or not. Given a question q, the topic entity binary classifier optimizes the following likelihood function to identify its label:

$$p_\theta\left(E_q \mid q\right) = \prod_{i}^{L} \left(p_i\right)^{y_i} \left(1 - p_i\right)^{1-y_i} \tag{1}$$

Here L is the size of the set E_q. y_i is a binary label which means whether the ith topic entity of E_q is the corresponding topic entity of question q.

3.2. Candidate path ranking

In this section, we devise a dedicated candidate path ranking method to obtain the final path where the answer exists. This stage consists of three parts, candidate answer path generation, path similarity computation, and result fusion.

3.2.1. Candidate path generation

Based on the result of topic entity recognition, we regard the generation of candidate paths as a graph node search process. In order to reduce the count of elements in the candidate answer path set P_q, a dynamic candidate path generation (DPG) algorithm is proposed by constraining the search space in the knowledge graph.

The main idea of DPG is to filter the irrelevant paths as early as possible. As described in Algorithm 1, we construct one-hop candidate paths through the first-order neighbors of the topic entity in set E_q and add them to P_q. When constructing the next-hop candidate paths, the similarity between the candidate paths and the question q are been calculated, and the candidate paths are sorted according to the similarity. Some candidate paths with

2 "When did the emperor buried in Maoling rule?"

3 {"<Maoling_(mausoleum of emperor Wu of Han dynasty)> ", "<Maoling_(a poem by Li Shangyin)>. ", "<Ming Maoling> "}.

4 "<Maoling_(a poem by Li Shangyin)>".

Algorithm 1: Dynamic candidate path generation.

Require:
A question, q;
The topic entity set of question q, $E_q = \{e_1, e_2, \ldots, e_n\}$;
The max number of hop from topic entity node, max_hop;
The number of paths is selected for next hot, k;
The knowledge graph, G;
The similarity computation model, $M_{similarity}$;
Ensure:
The set of candidate paths for question q, $P_q = \{p_1, p_2, \ldots\}$;
$present\_hop = 1$;
$candidate\_paths \Leftarrow$ Generate one hop candidate paths form G with help of E_q;
$P_q \Leftarrow candidate\_paths$
while $present\_hop < max\_hop$ **and** $candidate\_paths$ is not Empty **do**
  $present\_hop \Leftarrow present\_hop + 1$
  for p_i in $candidate\_paths$ **do**
    $s_i \Leftarrow$ Calculate the similarity between q and p_i by $M_{similarity}$;
  end for
  $selected\_paths \Leftarrow$ Select the top k paths by similarity score from $candidate\_paths$;
  $candidate\_paths \Leftarrow$ Generate candidate paths form G with help of $selected\_paths$;
  $P_q \Leftarrow P_q \cup candidate\_paths$
end while
return P_q;

lower scores are cut out, and only the candidate paths with top k similarity scores are expanded to generate the next hop candidate paths, so as to reduce the space and time occupied during graph search. Repeat the above process, and the expansion process would be terminated until the hop count threshold is reached.

3.2.2. Similarity computation model

As discussed above, we can get the candidate paths by proposed DPG algorithm. It is observed that the diversity of language expression makes it hard for a single model to learn semantic features between questions and candidate paths completely. To handle the intricate condition, we design four semantic similarity models from different focuses to calculate the similarity scores of question and candidate path. The advanced language representation model BERT is utilized in the four models for different purposes. The main idea of each model is discussed as follows.

(1) Basic Global Model. This model is devised to match the correct answers of simply questions. In practice, there are some simple questions, which means their answers are within one-hop range of the corresponding topic entities, such as for the question "姚明的妻子是谁?",[5] the answer lies in the neighbor nodes of "<姚明>".[6] For these simple questions, we consider the global representation of question and candidate path, and design the Basic Global Model based on BERT. As shown in Fig. 3, we use [CLS], [SEP] to splice question and candidate path, and use it as the input of BERT block. Finally, we take the hidden representation vector corresponding to [CLS] into a fully connected layer to obtain the similarity score of question and candidate path.

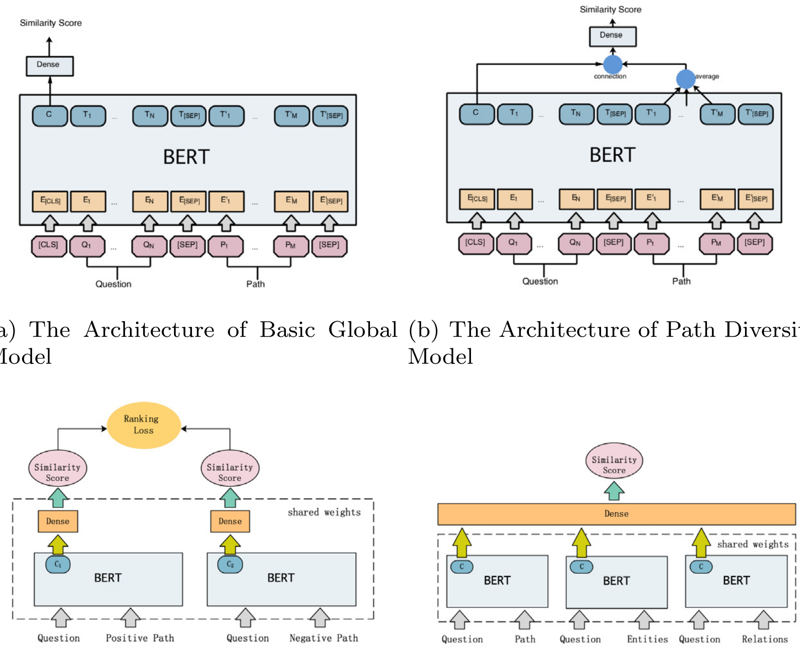
(2) Path Diversity Model. This model pays attention to the diverse semantic of path during complex question answering. For a question with extra semantic content, the expression of paths is usually diverse. In certain cases, the expression of answer paths is sequential. For example, the path of question "澳大利亚首都的邮编是多少?"[7] is "<澳大利亚> <首都> ?y . ?y <邮政区码> ?x."[8] While in some cases, the expression is reversed. For example, the path of question "葬于茂陵的皇帝在位于哪段时间?" is "?y <陵墓> < 茂陵_(汉武帝陵寝)>. ?y <在位时间> ?x .".[9] Usually, the first type of questions is easy to answer, but the second one is difficult. To overcome the above problem, we propose a model named Path Diversity Model, which focuses on the expression of paths. In this model, we calculate the average value of the hidden representation vectors of path tokens. Then we connect it with the hidden representation vector corresponding to the first input token ([CLS]) as the input of the next layer. The details are shown in Fig. 3(b).

(3) Implicit Relation Model. The model is primarily used for implicit relation mining of path. In KGQA task, it is a challenge to deal with the questions involving implicit relations. Here is an example. For the question "薛宝钗的哥哥外号叫什么?",[10] the correct answer path is "?y <妹妹> <薛宝钗>. ?y <别名> ?x ."[11], and the implicit relation is "<妹妹>"[12] For this kind of question, it is hard to get a satisfied result with a simple model. Inspired by Triplet Network [7], we design a new model named Implicit Relation Model. Fig. 3(c) shows the architecture of Implicit Relation Model, in which two BERT blocks share same parameters. To construct the positive candidate path and negative candidate path, we summarize 10 types of candidate path templates shown in Fig. 4 and apply them in KG. The path where the answer exists is called positive candidate path, which is constructed by the correct topic nodes and relations corresponding to the question. Meanwhile, the negative candidate paths are obtained by randomly replacing topic node or relation in the positive candidate path with its neighbors in KG. Then we train the model by using the triple loss function as shown below.

$$Loss = \max(0, \gamma + s(q, n) - s(q, p)) \tag{2}$$

Where $s(q, p)$ is the similarity score between a question and its positive candidate path, and $s(q, n)$ is the similarity score between the question and its negative candidate path. The $\gamma \in [0, 1]$ is an adjustable hyper-parameter, which means that the training

5 "*Who is Yao Ming's wife?*".
6 "*<Yao Ming>*".
7 "*What is the postcode of Australia's capital?*".
8 "*<Australia> <capital> ?y . ?y <postcode> ?x .*".
9 "*?y <mausoleum> < Maoling_(mausoleum of emperor Wu of Han dynasty)>. ?y <ruling time> ?x .*"
10 "*What's the nickname of Xue Baochai's brother?*".
11 "*?y <sister> <Xue Baochai>. ?y <nickname> ?x .*"
12 "*<sister>*".



(a) The Architecture of Basic Global Model

(b) The Architecture of Path Diversity Model

(c) The Architecture of Implicit Relation Model

(d) The Architecture of Path Decomposition Model

Fig. 3. The overview of similarity computation models. (a) is Basic Global Model which considers the global representation of question and path. (b) is Path Diversity Model which pays more attention on the expression of path. (c) is Implicit Relation Model which extracts the implicit relation. (d) is Path Decomposition Model which considers the difference in importance of entity and relation.

goal of the model is to make the similarity score of the positive candidate path be at least γ higher than that of the negative candidate path.

(4) Path Decomposition Model. The model is devised to mainly consider the local representation of path. Candidate paths are composed of entities and relations, which have different importance for different questions. Therefore, we propose Path Decomposition Model to automatically learn their importance weights. The model extracts the similarity features of question and path, question and entity, question and relation, and calculates the overall similarity score based on these three features. The architecture of model is shown in Fig. 3(d).

3.2.3. Result fusion

The above four models can deal with different type of questions, the final decision process is still needed to be designed. Here, a simple and effective fusion policy is proposed to combine the advantages of the four models. Ideally, we can use the sum of four scores computed by four models as the final score to evaluate a candidate path. However, this strategy does not apply to our work. The reason is that our candidate paths are dynamically generated based on similarity calculation, for the same question q, the candidate path sets obtained by different models are different. As a result, the score of a candidate path may be absent in certain models. To address the above issue, a novel strategy is proposed to make decision for the best path. Given a question q, we first get the top m candidate paths respectively from the four models to form the final candidate path set $P_q = \bigcup_i P_q^i$, where P_q^i is the candidate path set with the size of m obtained by similarity calculation model i. Then, we define Eq. (3) to calculate the filling score of model i for candidate paths which belong to P_q but not to P_q^i.

$$FScore_q^i = \max\left(0, k_1 * \min_{p_l \in P_q^i} s^i(q, p_l) - k_2\right) \tag{3}$$

For Eq. (3), $s^i(q, p_l)$ is the similarity score between question q and candidate path p_l which is calculated by model i. k_1 and k_2 are hyper-parameters. For each candidate path, we compute the total score from our models after the imputation of missing values. Then the candidate path with the largest total score is selected as our correct answer path.

4. Experiments

Our experiments are based on the Chinese Knowledge Base Question Answering (CKBQA) dataset, which is published by CCKS2019 (China Conference on Knowledge Graph and Semantic Computing). Among them, the Computer Technology Research Institute of Peking University provides about 3000 open-domain question–answer pairs, and Hundsun Technologies Inc. provides about 1000 question–answer pairs in the financial field. The data is split into training, validation and test sets. The training set contains 2298 samples, and both the validation set and test set contain 766 samples. We take PKUBASE constructed by the Institute of Computer Science of Peking University as the knowledge graph, and its online query terminal is http://pkubase.gstore.cn. This knowledge graph consists of 66,191,767 triples, 25,437,419 nodes and 408,261 relations.

In our experiments, macro precision, macro recall, average F1 score are used to evaluate the model and compare with others. Let Q be the question set, A_i is the answer set given by the model to the ith question, and G_i is the standard answer set of the ith question. Eq. (4)–(6) give the relevant calculation formulas. Moreover, the experimental results are analyzed in stages to evaluate the effectiveness of the improvements devised in this paper.

$$Macro\ Precision = \frac{1}{|Q|}\sum_{i=1}^{|Q|} P_i, \quad P_i = \frac{|A_i \cap G_i|}{|A_i|} \tag{4}$$

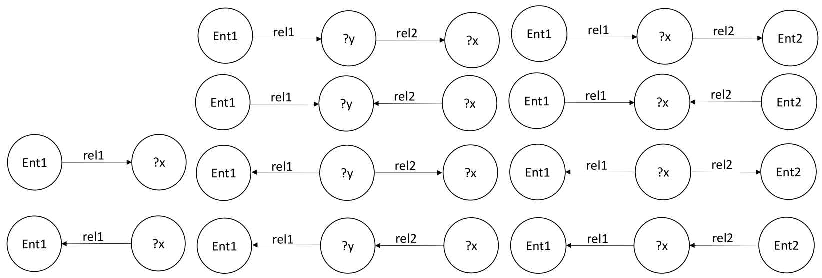


Fig. 4. Ten Types of candidate path templates.

Table 2
The evaluation of different sequence labeling models.

Model	Accuracy	Training epochs
BiLSTM-CRF model	84.72	44
BERT-CRF model	94.18	7
BERT-BiLSTM-CRF model	94.43	7
BERT-BiDT-CRF model	**95.10**	**2**

Table 3
The evaluation of different entity linking process.

Entity linking process	Recall	Avg count
Jinchang Luo (Rank 1)	90.73	2
Yiying Yang (Rank 3)	95.6	1
Mingyu Cao (Rank 4)	89.7	5
ESI-L	91.12	6
ESI-D	92.82	6
ESI-D-B	92.30	2

$$Macro\ Recall = \frac{1}{|Q|}\sum_{i=1}^{|Q|} R_i, \quad R_i = \frac{|A_i \cap G_i|}{|G_i|} \tag{5}$$

$$Average\ F1 = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{2P_iR_i}{P_i + R_i} \tag{6}$$

Our deep learning models are implemented in TensorFlow [33]. Adam is utilized as our optimizer, and an early stopping paradigm [34] is applied to retrieve the model with maximum validation F1 score. Moreover, our hyper-parameter settings largely follow our previous work [21], they are obtained by grid searching on the validation set. As a result, the parameter γ in Eq. (2) is set to 0.7 and the parameters m, k_1, k_2 mentioned in Section 3.2.3 are set to 3, 0.7, 0.2 respectively.

4.1. Experimental results of topic entity recognize

4.1.1. The effectiveness of sequence labeling model

We evaluate the performance of proposed BERT-BiDT-CRF model by accuracy and compare this model with other commonly used models. The comparison results are shown in Table 2. Here, the additional pre-trained model BERT is not involved in BiLSTM-CRF model, which gets the worst performance. BERT-BiLSTM-CRF model and BERT-BiDT-CRF model consider the order relationship between text sequences and memorize the contextual meaning, so their performances are better than BERT-CRF model. Moreover, BERT-BiDT-CRF model can perform more complex modeling for the sequential relationship between text sequences and reserve more semantic knowledge. Here, we measure the performance of models by accuracy, which is the ratio of correctly predicted entities to the total number of entities. We can see that BERT-BiDT-CRF model achieves higher performance, and its accuracy reaches 95.10%, meanwhile it is easier to converge than others.

Table 4
The evaluation of different candidate generation process.

Process	Parameter k	Avg count	Macro precision	Macro recall	Avg F1
FPG	–	4081	62.31	63.94	61.80
DPG	10	**1158**	**70.72**	**71.68**	**70.26**

Table 5
The influence of parameter k in DPG.

Parameter k	3	5	8	10	13
Avg count	429	621	885	1158	1293
Avg F1	67.95	69.25	69.32	**70.26**	69.54

4.1.2. The effectiveness of entity linking process

The correctness of topic entity recognition is directly related to the correctness of the final Question Answering task. Therefore, it is necessary to guarantee the recall of topic entity recognition and decrease the number of elements in entity set at the earliest. We use the recall and the average count of the topic entity set to measure the effectiveness of the entity linking process. The recall is defined as the Eq. (7). In this experiment, there are three entity linking methods, which are named as ESI-L, ESI-D and ESI-D-B separately. ESI-L means BERT-BiLSTM-CRF model is applied in topic entity recognition and the character matching is used for entity linking. ESI-D means that our improved topic entity recognition model BERT-BiDT-CRF is utilized and entity linking is the same as ESI-L. For ESI-D-B, its topic entity recognition model is the same as ESI-D, but our improved entity linking strategy is applied, which combines character matching and entity disambiguation classifier.

$$recall = \frac{|\{predicted\ entity\ nodes\} \cap \{correct\ entity\ nodes\}|}{|\{correct\ entity\ nodes\}|} \tag{7}$$

We compare our methods with the winning teams' solutions in the CCKS2109 KGQA competition [35]. The detail results are listed in Table 3. We can see that the proposed method ESI-D-B achieves better performance than Jinchang Luo team who is ranked at the first place in the competition. Compared with Yiying Yang who uses multiple pre-trained models, stop words, etc., our solution is slightly worse but it is more concise and efficient. Moreover, we conduct an ablation experiment to demonstrate the effectiveness of the entity disambiguation classifier. Specifically, the entity disambiguation classifier can decrease the average count of topic entities from 6 to 2 and guarantee the recall almost unchanged. Moreover, it can be found from Table 3 that the entity linking method ESI-D has better results than the ESI-L, which reflects the effectiveness of BERT-BiDT-CRF model devised in this paper.

4.2. Experimental results of candidate path ranking

4.2.1. The effectiveness of candidate generation process

Taking the usually used Fixed candidate Path Generation process (FPG) as the baseline, we design a related experiment to

Table 6
The evaluation of result fusion policy.

Model	Macro precision		Macro recall		Avg F1	
	ESI-L	ESI-D-B	ESI-L	ESI-D-B	ESI-L	ESI-D-B
Basic Global Model	68.07	70.72	69.11	71.68	67.65	70.26
Path Diversity Model	64.13	69.17	64.75	68.56	63.63	68.15
Implicit Relation Model	61.97	64.30	61.82	64.54	61.20	63.72
Path Decomposition Model	65.37	70.42	65.33	69.74	64.56	69.17
Basic Global Model*	60.02	61.51	60.49	62.00	59.56	61.04
Fusion model	75.21	**76.76**	76.13	**77.17**	74.63	**76.11**

demonstrate the effectiveness of the dynamic candidate generation process (DPG). FPG uses each candidate entity as the starting node to construct all paths within the hop threshold as candidate paths. Better than FPG, our proposed DPG can dynamically generate candidate paths. In experiment, according to the characteristics of the dataset, we set the hop threshold to 2. The experimental results of the two candidate path generation process are shown in Table 4. Note that FPG and DPG use the same topic entity recognition strategy and similarity calculation model, ESI-D-B and Basic Global Model respectively. We measure the computational cost by the average count of candidate paths. The experiment results demonstrate that the proposed DPG is effective. It can reduce the computational overhead by pruning the irrelevant paths, and improve the F1 score from 61.80% to 70.26%.

According to DPG, only the candidate paths with top k similarity scores are expanded to generate the next hop. We further introduce an experiment to evaluate the effect of parameter k. As shown in Table 5, when $k < 10$, the F1 score increases with the value of k. And the result shows that the number of missed answers caused by path pruning in early hops is gradually reduced. However, when $k > 10$, too much path expansion would lead to extra noise, which limits the performance of the model. Furthermore, it is obvious that there is a positive correlation between the amount of computation and parameter k. Considering the effectiveness of DPG, we set k to 10.

4.2.2. The effectiveness of fusion policy

Moreover, we also conduct an experiment to compare the performance of different similarity calculation models and demonstrate the effectiveness of the fusion policy. To make a complete comparison, the results of ESI-L and ESI-D-B in topic entity recognition are used for candidate path ranking models respectively. Moreover, To avoid that the cascade error in the stage of topic entity recognition affects subsequent similarity calculation model, the Basic Global Model* is introduced in experiment. Different from Basic Global Model, we directly search for questions on the ternary-based ES index to generate candidate paths in this model. As shown in Table 6, we use macro precision, macro recall and average F1 score to evaluate the performance of proposed methods.

When the candidate path ranking model is identical, the results cooperated with ESI-D-B totally achieve better performance than ESI-L. It shows that the introduce of BERT-BiDT-CRF model and entity disambiguation classifier is effective. For the fusion model, it combines the five result files according to the fusion policy presented in Section 3.2. We can see that precision, recall and F1 score are significantly enhanced by employing the fusion model, which verifies that our fusion policy can integrate the advantages from each model.

In addition, we analyze the average hit rates for top-m paths evaluated by models. Here the hit rates are quantified by F1 score. As shown in Fig. 5, on the one hand the hit rates of 6 models raise with the increase of variable m. On the other hand, the fusion model can always achieve the best hit rates under different condition. The experimental result verifies the advantages of our fusion policy again.

Table 7
The evaluation results on CKBQA dataset.

Team	Avg F1
Jinchang Luo (Rank 1)	73.54
Pengju Zhang (Rank 2)	73.08
Yiying Yang (Rank 3)	70.45
Mingyu Cao (Rank 4)	67.60
Our approach (ESI-L) [21]	74.63
Our approach (ESI-D-B)	**76.11**

4.2.3. The comparison of KGQA approaches

As mentioned above, the dataset used in our experiments comes from CCKS2019, and it is a competition dataset. Therefore, it is necessary to compare our results with the winning teams in this competition, which helps to show the effectiveness of proposed approach. The evaluation standard of the competition is average F1 score, and we can see the results from Table 7.

Similar to our model, the top four teams also use the pretrained model BERT. The difference is that their approaches require manual design of rules. For example, in the similarity calculation stage, Jinchang Luo designs and extracts 39 features and uses 10-fold cross-validation to reduce variance. Pengju Zhang uses BERT-based and rule-based methods to calculate similarity from the perspective of path and relation, and then adopts different methods for different types of questions. In contrast, our approach without manual design rules, is more concise and adaptable. And from the results of experiment, we can know that our approach achieves better performance. In addition, through the improved topic entity recognition ESI-D-B, the F1 score of our approach has been further improved, which reflects the effectiveness of the improvement in topic entity recognition stage.

5. Conclusion and future work

In this paper, we propose a comprehensive approach with multi-focus fusion model for the KGQA task, and achieve a significant improvement on CKBQA dataset compared with methods in CCKS2019 competition. The proposed approach has two stages, topic entity recognition and candidate path ranking. During the first stage, we construct a sequence labeling model which achieves richer syntactic features by deeply modeling the transition between hidden states. And an effective entity linking strategy is presented by combining character matching and entity disambiguation model. During the second stage, we first design a new process DPG to dynamically generate candidate paths, which can eliminate irrelevant candidate paths as early as possible and enhance recall of answers simultaneously. Then, we propose four parallel models for learning the feature vectors of questions and candidate paths from multiple focuses and computing similarity between them. Finally, we present an effective fusion policy to combine the above four similarity models, which can choose the final right answer paths after completing the missing path scores. A series of experiments demonstrate that our approach is effective from multiple stages and can achieve better performance than all methods published in CCKS2019 KGQA competition.

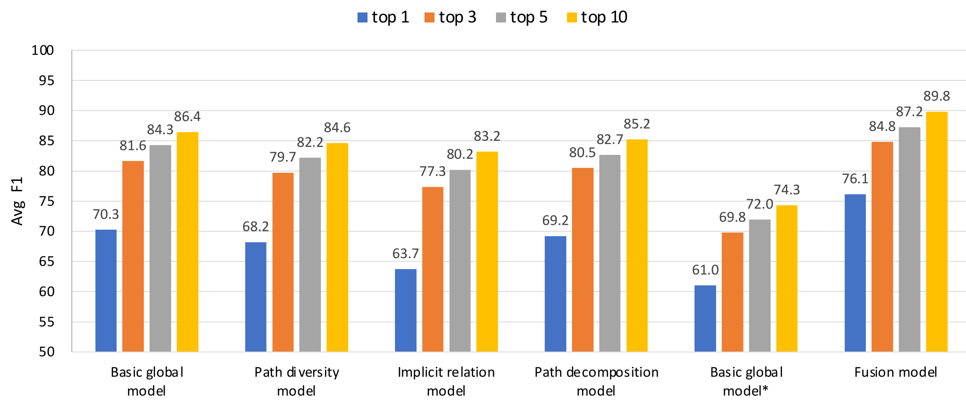


Fig. 5. The evaluation of model results with different matching scope.

Note that the proposed approach is designed for open-domain KGQA task, as the complexity of question set, the robustness of the model should be improved further. Although we have considered the diversity of language expression, the prediction algorithm for question class can be introduced and integrated into KGQA task in future work. Moreover, there need many valuable efforts to obtain an improved model suitable for special vertical-domain KGQA task.

CRediT authorship contribution statement

Haobo Xiong: Conceptualization, Methodology, Writing - original draft. **Shuting Wang:** Investigation, Conceptualization, Validation. **Mingrong Tang:** Methodology, Data curation, Software. **Liping Wang:** Conceptualization, Writing - review & editing, Project administration. **Xuemin Lin:** Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the National Key R&D Program of China under grant 2018AAA0102502 and the GuangDong Basic and Applied Basic Research Foundation 2019B1515120048.

References

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.

[2] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (10) (2014) 78–85.

[3] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, G. Weikum, Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames, in: International Semantic Web Conference, 2016, pp. 177–185.

[4] X. Luo, L. Liu, Y. Yang, L. Bo, Y. Cao, J. Wu, Q. Li, K. Yang, K.Q. Zhu, Alicoco: Alibaba E-commerce cognitive concept net, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020, pp. 313–327.

[5] Q. Cai, A. Yates, Large-scale semantic parsing via schema matching and lexicon extension, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013, pp. 423–433.

[6] S. Reddy, O. Täckström, M. Collins, T. Kwiatkowski, D. Das, M. Steedman, M. Lapata, Transforming dependency structures to logical forms for semantic parsing, Trans. Assoc. Comput. Linguist. 4 (2016) 127–140.

[7] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: International Workshop on Similarity-Based Pattern Recognition, pp. 84–92.

[8] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor, Cypher: An evolving query language for property graphs, in: Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 1433–1445.

[9] K. Xu, Y. Feng, S. Reddy, S. Huang, D. Zhao, Enhancing freebase question answering using textual evidence, 2016, arXiv preprint arXiv:1603.00957.

[10] X. Yao, B. Van Durme, Information extraction over structured data: Question answering with freebase, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 956–966.

[11] A. Bordes, J. Weston, N. Usunier, Open question answering with weakly supervised embedding models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2014, pp. 165–180.

[12] A. Bordes, S. Chopra, J. Weston, Question answering with subgraph embeddings, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 615–620.

[13] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.

[14] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, 2019, pp. 5753–5763.

[18] F. Meng, J. Zhang, Dtmt: A novel deep transition architecture for neural machine translation, in: The Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 224–231.

[19] R. Pascanu, C. Gulcehre, K. Cho, Y. Bengio, How to construct deep recurrent neural networks, in: 2nd International Conference on Learning Representations, 2014.

[20] S. Wiseman, A.M. Rush, Sequence-to-sequence learning as beam-search optimization, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1296–1306.

[21] M. Tang, H. Xiong, L. Wang, X. Lin, A dynamic answering path based fusion model for kgqa, in: International Conference on Knowledge Science, Engineering and Management, 2020, pp. 235–246.

[22] P. Liang, M.I. Jordan, D. Klein, Learning dependency-based compositional semantics, Comput. Linguist. 39 (2) (2013) 389–446.

[23] S. Hu, L. Zou, J.X. Yu, H. Wang, D. Zhao, Answering natural language questions by subgraph matching over knowledge graphs, IEEE Trans. Knowl. Data Eng. 30 (5) (2018) 824–837.

[24] D. Guo, D. Tang, N. Duan, M. Zhou, J. Yin, Dialog-to-action: Conversational question answering over a large-scale knowledge base, in: Advances in Neural Information Processing Systems, 2018, pp. 2942–2951.

[25] X. Huang, J. Zhang, D. Li, P. Li, Knowledge graph embedding based question answering, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 105–113.

[26] R. Wang, M. Wang, J. Liu, W. Chen, M. Cochez, S. Decker, Leveraging knowledge graph embeddings for natural language question answering, in: International Conference on Database Systems for Advanced Applications, 2019, pp. 659–675.

[27] L. Dong, F. Wei, M. Zhou, K. Xu, 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), 2015, pp. 260–269.

[28] Y. Lai, Y. Feng, X. Yu, Z. Wang, K. Xu, D. Zhao, Lattice cnns for matching based chinese question answering, in: The Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 6634–6641.

[29] Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, J. Zhao, An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 221–231.

[30] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory 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), 2015, pp. 1556–1566.

[31] Y. Liu, F. Meng, J. Zhang, J. Xu, Y. Chen, J. Zhou, Gcdt: A global context enhanced deep transition architecture for sequence labeling, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2431–2441.

[32] J. Lafferty, A. Mccallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 282–289.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {*USENIX*} Symposium on Operating Systems Design and Implementation ({*OSDI*} 16), 2016, pp. 265–283.

[34] Y. Yao, L. Rosasco, A. Caponnetto, On early stopping in gradient descent learning, Constr. Approx. 26 (2) (2007) 289–315.

[35] CCKS, China Conference on knowledge graph and semantic computing review papers, 2019, https://conference.bj.bcebos.com/ccks2019/eval/webpage/index.html.