SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Natural Language Processing and Social Computing

November 2017, Vol. 60 110103:1–110103:15
doi: 10.1007/s11432-017-9136-x

Topic enhanced deep structured semantic models for knowledge base question answering

Zhiwen XIE[1], Zhao ZENG[1], Guangyou ZHOU[1,2*] & Weijun WANG[2]

[1]*School of Computer, Central China Normal University, Wuhan* 430079*, China;*
[2]*Key Laboratory of Adolescent Cyberpsychology and Behavior, Ministry of Education, Central China Normal University, Wuhan* 430079*, China*

Received March 24, 2017; accepted June 20, 2017; published online September 21, 2017

Abstract Knowledge Base Question Answering (KBQA) is a hot research topic in natural language processing (NLP). The most challenging problem in KBQA is how to understand the semantic information of natural language questions and how to bridge the semantic gap between the natural language questions and the structured fact triples in knowledge base. This paper focuses on simple questions which can be answered by a single fact triple in knowledge base. We propose a topic enhanced deep structured semantic model for KBQA. The proposed method considers the task of KBQA as a matching problem between questions and the subjects and predicates in knowledge base. And the proposed model consists of two stages to match the subjects and predicates, respectively. In the first stage, we propose a Convolutional based Topic Entity Extraction Model (CTEEM) to extract topic entities mentioned in questions. With the extracted entities, we can retrieve the relevant candidate fact triples from knowledge base and obviously decrease the amount of noising candidates. In the second stage, we employ Deep Structured Semantic Models (DSSMs) to compute the semantic relevant score between questions and predicates in the candidates. And we combine the semantic level and the lexical level scores to rank the candidates. We evaluate the proposed method on KBQA dataset released by NLPCC-ICCPOL 2016. The experimental results show that our proposed method achieves the third place among the 21 submitted systems. Furthermore, we also extend the DSSM by using BiLSTM and integrate a convolutional structure on the top of BiLSTM layers. Our experimental results show that the extension models can further improve the performance.

Keywords question answering, deep learning, knowledge base, semantic matching, topic entity

Citation Xie Z W, Zeng Z, Zhou G Y, et al. Topic enhanced deep structured semantic models for knowledge base question answering. Sci China Inf Sci, 2017, 60(2): 110103, doi: 10.1007/s11432-017-9136-x

1 Introduction

Automatic question answering, which is aimed at directly generating the exact answers to natural language questions, is a typical and challenging task in natural language processing (NLP). Recently, with the rise of large-scale, structured knowledge bases, such as Freebase [1] and DBPedia [2], Knowledge Base Question Answering becomes a popular tendency of the research in question answering. In this paper, we focus on the shared KBQA task for Chinese language hosted by NLPCC-ICCPOL 2016. The knowledge base considered in this work is a collection of facts which are stored as (subject|||predicate|||object) triples.

* Corresponding author (email: gyzhou@mail.ccnu.edu.cn)

Due to the variability of the natural language expressions, the most challenging problem in KBQA is how to understand the semantic information of natural language questions and how to bridge the semantic gap between the questions and the structured fact triples in knowledge bases. To address these challenges, the studies in the literature can be divided into two groups. The first group studies attempt to train semantic parsers, which can transform the natural questions into logical forms or SPARQL [3,4]. This kind of methods need large-scale annotated logical forms of questions as supervision to train the parser. However, manually annotating these logical forms is a very expensive and time-consuming job. Moreover, the performance of most semantic parsers relies on predefined rules, hand-craft features and linguistic tools. The second group studies are based on information retrieval, which first retrieves a set of candidates and then extracts features to rank these candidates [5–8]. However, the information retrieval based methods use all possible n-grams of words among the questions to retrieve the candidates from the knowledge base, which may result in noise candidate triples.

In order to solve the above problems, we propose a topic enhanced deep structured semantic model for KBQA. The task of KBQA is viewed as a matching problem between the natural language questions and the subjects and predicates in knowledge base. The subjects in knowledge base are always the same with the topic entities mentioned in questions, but the predicates in knowledge base can be expressed in myriad ways. Due to the different features, we use two different deep learning strategies to match the subjects and the predicates, respectively. First, we propose a Convolutional based Topic Entity Extraction Model (CTEEM) to automatically extract topic entity in a question, and we use the extracted entities to retrieve the relevant candidate triples instead of using the n-grams of words in a question. Then, we employ Deep Structured Semantic Models (DSSMs) to compute the similarity between the question and the predicates in the semantic level without using any hand-crafted features.

Specially, we focus on simple questions, which can be answered by a single fact triple in knowledge base. For example, the question "谁知道孛儿只斤分布在哪些地区?" can be answered by the single fact triple (孛儿只斤|||分布地区|||中国蒙古国俄罗斯). We assume that each of these simple questions contains a single topic entity (that is, the main entity mentioned in the question) and can be linked to the subject of a triple in knowledge base. The topic entity in the example question is "孛儿只斤". The simple question answering can be addressed by linking the topic entity in the questions to the subjects of triples in knowledge base and matching the questions with the predicates of triples in knowledge base. Thus, we can deal with the KBQA problem in two stages: candidate retrieval stage which is applied to link the topic entities in questions to related candidate triples and answer selection stage which is used to match the question with the predicates in the candidate triples.

In candidate retrieval stage, we propose a Convolutional based Topic Entity Extraction Model (CTEEM) to recognize the topic entities in questions, without using any expensive feature engineering and additional linguist tools like part-of-speech tagging. We apply convolutional operation in the CTEEM to capture the abstract features of the input word sequence and generate a output tag sequence for the input. According to the output tag sequence, we can recognize the topic entity words in the input question. The extracted topic entity is fed into the information retrieval system to search the related candidate triples. In candidate retrieval stage, we can match the natural language questions with the subjects of the fact triples in knowledge base by utilizing the extracted topic entity.

In answer selection stage, we measure the similarity between the question and the predicate of each candidate triple. To bridge the semantic gap between the natural language question and the predicate in knowledge base, we develop several Deep Structured Semantic Models (DSSMs) to learn the semantic representation of the natural language questions, as well as the predicates in knowledge base. The recently proposed Convolutional Deep Structured Semantic Model (CDSSM) [9] has been used to learn the semantic representation of sentences. In this paper, we develop a BiLSTM based DSSM (BDSSM) which is able to deal with sequence information and capture global features of each word. Additionally, we also present a BiLSTM Convolutional based Deep Structured Semantic Model (BCDSSM), which apply a convolutional operation over the output of BiLSTM layer to capture richer semantic features. We use these DSSMs to translate the question and the predicate into semantic vector representations with the same dimension, so that we can use a distance function to measure the semantic similarity.

The DSSMs can be used to solve the mismatch problems between natural language question and the predicate in knowledge base by measuring the high level semantic similarity. However, sometimes the lexical similarity is also very import to rank the candidates because of the plenty of overlap words which occur both in questions and predicates. So we also consider lexical level similarity, which can capture the shallow level features of the questions and the predicates. We combine the semantic matching score and the lexical matching score as the final matching score to rank the candidates.

We evaluate our method on the Chinese Knowledge Base Question Answering dataset released by NLPCC-ICCPOL 2016. The experimental results show that our proposed method achieves the third place among the 21 submitted systems. We also show that the extension models can further improve the performance.

2 Related work

Automatic question answering is a typical and challenging task in natural language processing (NLP), which is aimed at automatically generating a direct and exact answer to a natural language question. Due to the advance of structured and large-scale knowledge base, Knowledge Base Question Answering (KBQA) has attached much attention both in industrial and academic fields. Generally, previous work in the literature can be divided into two groups: semantic parsing based methods and information retrieval based methods.

For semantic parsing based methods, the basic idea is to translate a natural language question into logical form which can be executed on knowledge base. Conventional semantic parsers require annotated logical forms as training signal [10, 11], which is very expensive especially when the scale is large. And some semantic parsers also suffered a limitation of scaling to large dataset. Recent studies attempt to leverage question-answer pairs to learn weak supervision semantic parses. Liang et al. [12] proposed a dependency-based compositional semantics (DCS), which used the question-answer pairs to infer the latent logical forms and learn the parameters in the model. To make the semantic parsers be able to scale to large dataset, Cai and Yates [13] proposed a method to automatically construct Combinatory Categorial Grammar (CCG) lexical entries for semantic parser by making it a prediction task. Berant et al. [3] developed a semantic parser that can both train without annotated logical forms and scale to large knowledge bases. Berant et al. [4] presented a novel approach to learning semantic parsers based on paraphrasing that can exploit large amounts of text not covered by the knowledge base. However, some of these methods still rely on hand-craft features and pre-defined rules.

For information retrieval based methods, they firstly retrieve a set of relevant candidates from knowledge base and then conduct further analysis to rank these candidates and select answers [5–8]. Yao and Durme [8] extracted question features by using rules and relied on dependency parse results. Some other studies [6, 7] used embedding-based models to learn low-dimensional vectors for question words and knowledge base constitutes, and used the sum of these vectors to represent questions and candidate answers, which ignore the word order information. Bordes et al. [5] used a memory networks framework to select answers. However, most of the information retrieval based methods use all possible n-grams of words of the question to retrieve the candidates from the knowledge base which can introduce lots of noise candidate triples. The system proposed by Lai et al. [14] used a SPE (subject predicate extraction) algorithm to extract subject-predicate pair from the question and translate it to a KB query to search the candidates and use a method based on word vector similarity and predicate attention to score the candidate predicates. The method proposed by Wang et al. [15] used a classifier to judge whether the predicate in the triple is what the question asked for. Yang et al. [16] used a Topic Phrase Detecting model based on phrase-entities dictionary to detect the topic phrase in the question and used several answer ranking models to rank the candidates. Xie et al. [17] proposed a topic entity extraction model based on convolutional neural network which can extract the topic entities in the question. However, the topic entity extraction model used in [17] used max pooling operation to capture the global features of the input question and then translated the global features into the predict label vector by using a full

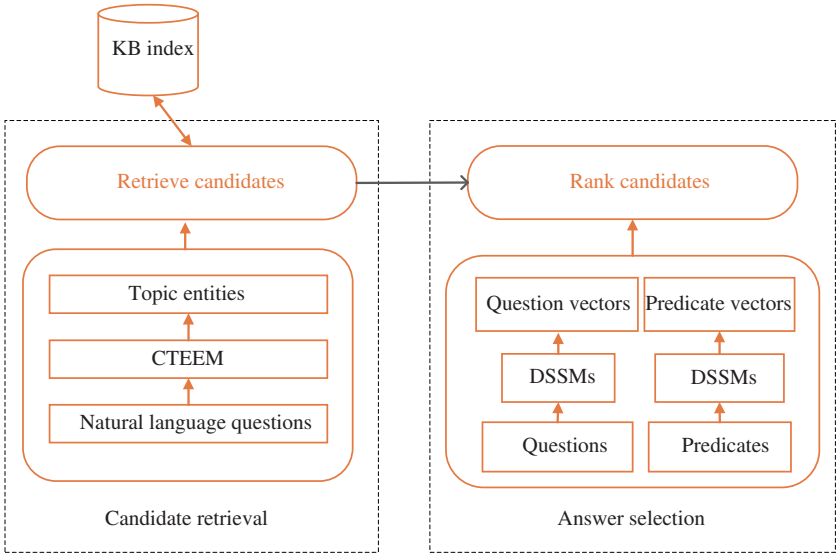

Figure 1 (Color online) The overview of the KBQA system framework.

connection layer, which failed to utilize the local contextual features of the words in questions and unable to process variable length input sequence. In this paper, we proposed a different Convolutional based Topic Entity Extraction Model (CTEEM) without using the max pooling operation and full connection layer. Thus, we can use the convolutional operation to capture the local contextual information. And the proposed CTEEM can deal with variable length sequence by using the convolutional layer to predict the output sequence instead of using a full connection layer.

Recently, deep learning models have been widely applied in natural language processing (NLP) and achieved impressive results, such as word vector representations [18–20] or question representations [21, 22], deep structured semantic models (DSSM) [9,23,24], machine translation [25] and text summarization [26]. With the successful application of deep learning in these research areas, some researchers also explore the deep learning for KBQA. Yih et al. [27,28] presented a Staged Query Graph Generation method to learn semantic parser by using a convolutional neural network model. Dong et al. [29] used multi-column convolutional neural networks to learn the representations of different aspects of the questions: answer path, answer context and answer type. Zhang et al. [30] proposed a neural attention-based model which can learn dynamically representation of the questions according to different aspects of various candidate answer. Jain [31] introduced a Factual Memory Network, which was used to answer questions by extracting and reasoning over relevant facts from the knowledge base. Dai et al. [32] proposed a Conditional Focused neural network-based approach to answering factoid questions with knowledge bases. These methods use deep learning methods in their system and successfully improve the performance of the question answering. In our paper, we view the task of KBQA as a matching problem between the natural language questions and the subjects and predicates in knowledge base. The subjects in knowledge base are always the same with the topic entity mentioned in questions, but the predicates in knowledge base can be expressed in myriad ways. Due to the different features, we use two different deep learning strategies to match the subjects and the predicates, respectively. The first one is a convolutional based topic extraction model, which is used to automatically recognize the topic entities in questions, the other one is a deep structured semantic model, which is used to match the predicates and the questions in semantic level.

3 Methods

Figure 1 illustrates the overview of the proposed approach. As shown in Figure 1, we deal with the task of KBQA in two stages: Candidate Retrieval and Answer Selection. In Candidate Retrieval stage, each fact

triple in knowledge base is viewed as a document with three fields (namely, subject, predicate and object) and stored in a inverted index. We apply a Convolutional based Topic Entity Extraction Model (CTEEM) to extract topic entities in the natural language questions. Then, the topic entities are used as keywords to retrieve candidate triples over the subject field in the KB index. In Answer Selection stage, we use Deep Structured Semantic Models (DSSMs) to map the natural language questions and the predicates into semantic representations with the same dimension. The semantic similarity between the semantic representation of the questions and the predicates can be computed by using some popular distance functions, such as cosine. Then, we combine the semantic matching score and the lexical matching score to rank the candidates. It seems more reasonable to compute the similarity between a question excluding the extracted topic entity and a predicate. However, since the topic entities extracted by the CTEEM are not always correct, if we discard the extracted topic entity in the question, we may loss some important information in the question. So we simply use the full question to match the candidate predicates.

In this paper, we mainly focus on the simple question which can be answered by a single fact in knowledge base. The simple questions can be answered by matching the subject and predicate of the fact triples in knowledge base. Most of the simple questions contain a single topic entity which can linked to the subject of the knowledge base. So the candidate retrieval stage can be viewed as the matching between a question and the subjects in knowledge base. The Answer Selection stage can be viewed as the matching between a question and the predicates of the candidate triples.

In this section, we first describe the Convolutional based Topic Entity Extraction Model (CTEEM) in Subsection 3.1, then we describe the Deep Structured Semantic Models (DSSMs) in Subsection 3.2. Finally, we present the details of the Candidate Retrieval and Answer Selection stages of the proposed method in Subsections 3.3 and 3.4, respectively.

3.1 Convolutional based topic entity extraction model

For a simple question, there is always a single topic entity mentioned in the question which can be linked to subjects of the relevant fact triples in knowledge base. Most of the previous studies use all possible n-grams of words of the question as candidate topic entities to retrieve the related candidate triples from knowledge base which can introduce lots of noise candidate triples. If we can extract the exact topic entity in the question, we can substantially improve the quality of the candidate triples by decreasing the amount of the noise candidates. So it is necessary to recognize the topic entities in the question directly.

Most conventional entity extraction methods rely on the linguistic tools such as Part-of-Speech tagging and hand-defined features, such as Conditional Random Fields (CRF) [33]. In this paper, we proposed a Convolutional based Topic Entity Extraction Model (CTEEM) to extract the topic entities directly. In contrast, we do not use any linguistic tools and hand-defined features. The architecture of the CTEEM is illustrated in Figure 2. Xie el al. [17] proposed a topic entity extraction model to extract topic entities. However, the model proposed in [17] used convolutional operation and max pooling operation to capture global features of the whole question and then translated the global features into the output label vector by using a full connection nerual network, which cannot capture the local contextual features for a speical word and unable to deal with vaiable length of questions. Different with the topic entity extraction model used in [17], without using the max pooling operation to capture the global features, the CTEEM proposed in this paper uses convolutional operations to capture the local contextual features for each word in the question in hidden layers. Then, we use another convolutional layer to map the contextual features into the output label for each word instead of using full connection layer, which make the model have the ability of handling the questions with variable length.

Following [17], we classified the words in the question into two classes: the words belong to the topic entity and the words not belong to the topic entity. Given a question $q = (w_1, w_2, \ldots, w_n)$, where w_i denotes the i-th word in the question. We define the label of the question as $y = (y_1, y_w, \ldots, y_n)$, where y_i is the category of the word w_i. If the word w_i belongs to the topic entity, we set $y_i = 1$, otherwise we set $y_i = 0$. For example, the topic entity of the question "商务星健身管理软件的经营范围是什么" is "商务星健身管理软件". So the label of the each word in the question is "商务/1 星/1 健身/1 管理/1 软件/1

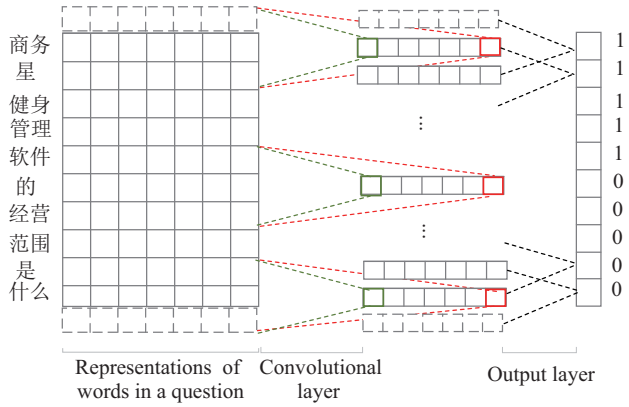

Figure 2 (Color online) The architecture of the CTEEM for an example question.

的/0 经营/0 范围/0 是/0 什么/0" and the label vector is $y = (1, 1, 1, 1, 1, 0, 0, 0, 0, 0)$.

In the CTEEM, we represent the words in questions as low dimensional vectors by using a pre-trained word embedding. Firstly, a lookup-table operation is used to map the word w_i into a low dimensional vector representation. Thus, we can obtain the embedding representation of the question, which is defined as $x = (x_1, x_2, \ldots, x_n)$, where $x_i \in \mathbb{R}^k$ is word vector of the word w_i in the question. During training time, we constraint the variable-length input sequences of questions to a fixed length n by using a padding operation, so that we can train the model in mini-batches.

Then, we capture the local contextual features of each word in a question by using convolutional operations which can be viewed as a local sliding window. The word vectors within a sliding window are encoded into a high level contextual feature by using the following convolution function:

$$c_i = \tanh(W_h[x_{i-s}^{\mathrm{T}} \cdots x_i^{\mathrm{T}} \cdots x_{i+s}^{\mathrm{T}}] + b_h), \tag{1}$$

where $2s + 1$ is the window size of the convolutional operation, c_i is the local contextual features of the word vector x_i, $W_h \in \mathbb{R}^{d \times (2s+1)k}$ (d is the dimension of the output vector c_i) is the weight matrix of the convolutional operation, $b_h \in \mathbb{R}^{d \times 1}$ is the bias vector, $[x_{i-s}^{\mathrm{T}} \cdots x_i^{\mathrm{T}} \cdots x_{i+s}^{\mathrm{T}}]$ refers to the concatenation of the vectors $x_{i-s}^{\mathrm{T}} \cdots x_i^{\mathrm{T}} \cdots x_{i+s}^{\mathrm{T}}$. As is shown in Figure 2, the borders of the input sequence is padded with zero vectors. Multiple similar hidden convolutional layers can be applied to form a deep topic entity extraction model which is able to obtain more abstract features.

Finally, we use a convolutional operation again at the output layer which can capture high level contextual features of the words in the input sequence. The neural network is trained to learn the contextual features and predict the label for each word according to the contextual information. The output word class label can be denoted as

$$z_i = \sigma(W_o[c_{i-s} \cdots c_i \cdots c_{i+s}] + b_o), \tag{2}$$

where $W_o \in \mathbb{R}^{1 \times (2s+1)d}$ (d is the dimension of the context vector c_i) is the weight matrix of the convolutional operation, $b_o \in \mathbb{R}^{1 \times 1}$ is the bias vector. σ is the sigmoid function which is defined as

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}. \tag{3}$$

The output vector is denoted as $z = (z_1, z_2, \ldots, z_n)$, where z_i is the predicted tag of the word w_i in the question. The goal of CTEEM is to minimize the error between the predicted result z and true label vector y. The objective function is computed by using a mean squared error function, which is defined as

$$\mathrm{MSE}(w, b) = \frac{1}{n} \sum_{i=1}^{n} (z_i - y_i)^2 + \lambda \|w\|_2^2, \tag{4}$$

where w and b are the parameters of the model. $\|w\|_2^2$ is the L2-regularization of the weight vectors w which is added to prevent overfitting. λ is a parameter that controls the degree of the penalty imposed on the weight parameters.

Table 1 Topic entities extracted by the CTEEM

Question	Topic entity
你|知道|雷锋|日记| – |拼音|版|的|副标题|是|什么|吗|?	雷锋日记–拼音版
你|知道|倭|叉|角|羚|这种|动物|是|什么|纲|的|吗|?	倭叉角羚
你|知道|拼搏|奥运|连连看|占|多|大|的|内存|吗|?	拼搏奥运连连看
奥图|码|pv|3225|这款|产品|是|干什么|用|的|啊|?	奥图码pv 3225
恒大|金碧|天下|是|什么|样子|的|房子|啊|?	恒大金碧天下
上海|假日|之|星|酒店|(|长宁店|)|在|哪|啊|?	上海假日之星酒店
你|知道|资产|预计|未来|现金流量|是|什么|意思|吗|?	资产预计未来现金流量

In our experiment, we obtain 14257 question-entity pairs and split this dataset into 10000 training data and 4257 testing data. We conduct experiments on the testing dataset to evaluate the proposed CTEEM and we evaluate the result by using word level F_1 score where a predicted word label is viewed as right if the word is labelled correctly and sentence level accuracy where a sentence is viewed as right if the topic entity is fully extracted correctly. Experimental results show that our proposed CTEEM achieves a F_1 score of 97.52% and achieves an accuracy of 91.02%. We also conduct comparison experiment by using CRF model which is trained with Stanford CRF-NER toolkit [1)]. And the CRF model achieves a F_1 score of 96.51% and an accuracy of 88.44%. Compared to the traditional CRF model which requires lots of hand-defined features, our CTEEM does not rely on any hand-defined features and can achieve better results than CRF model. With the extracted topic entities, the noise candidates can be substantially decreased and quality of the related candidate triples will be significantly improved. Table 1 shows some example results of CTEEM. The output labels of the CTEEM is not always continuous, to address this problem, we obtain the topic entities by simply extracting the word sequences between the first and last words whose predicted labels are 1.

3.2 Deep structured semantic models

Due to the informal and various expressions of the natural language questions, one of the most challenging problems in KBQA is to bridge the gap between the natural questions and the structured fact triples in knowledge base. For example, for the natural language question "你知道《门》多少钱可以买到吗?", the structured knowledge triple in knowledge base is "《门》(小说)|||价格|||20.00". By using conventional lexical level matching methods, we cannot measure the similarity between the question and the structured predicate "价格". Thanks to the rapid development of the deep learning in NLP, we can learn the latent semantic representation of the natural language sentences. Some previous studies have successfully applied the deep learning methods to learn the semantic representation, such as the Deep Structured Semantic Model (DSSM) proposed in [23], the improved Convolutional DSSM (CDSSM) presented in [24] and Long-Short-Term Memory DSSM (LSTM-DSSM) [34].

To handle the mismatch problem between the natural language questions and the structured fact triples in knowledge base, we apply several variants of the deep structured semantic models, including CDSSM which is based Convolutional Neural Network (CNN), BDSSM which is based on Bidirectional Long Short-Term Memory (BiLSTM), and BCDSSM which is based BiLSTM and CNN.

3.2.1 *CDSSM*

The CDSSM used in this paper is shown in Figure 3. We use the word embedding to represent the questions which can represent words as low dimensional and distributed vectors. Firstly, the input question and predicate sequences are represented as k-dimension word vector sequences by using a word embedding layer. Then, we separately apply a convolutional layer to deal with the word vector sequences of the questions and predicates. The convolutional operation can capture the local contextual information by applying a sliding window over the input sequence. Then a max pooling layer is followed to capture the most salient features from the output features of the convolutional layer and form a fixed-length

1) nlp.stanford.edu/software/.

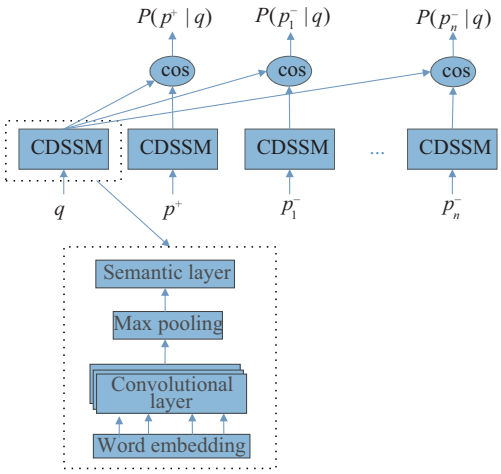

Figure 3 (Color online) The architecture of the CDSSM used in this paper.

global feature vector. The global feature vector is fed into a full connection semantic layer to obtain the semantic vector representation of the input sequence.

The semantic similarity between a question q and a predicate p can be computed by using cosine function, which is defined as

$$R(q,p) = \cos(y_q, y_p) = \frac{y_q^{\mathrm{T}} y_p}{\|y_q\|\|y_p\|}, \tag{5}$$

where y_q is the semantic vector representation of the question q and y_p is the semantic vector representation of the predicate p.

Following [23], by using a softmax function, we can compute the conditional probability of a predicate given the question q, which is defined as

$$P(p|q) = \frac{\exp(\lambda R(q,p))}{\sum_{p' \in \boldsymbol{P}} \exp(\lambda R(q,p'))}, \tag{6}$$

where λ is a smoothing factor in the function. $\boldsymbol{P}$ is a set of candidate predicates of the question q, including several negative predicate samples and a positive predicate sample. The semantic model is trained to maximize the likelihood of the positive predicate. Therefore, the objective function can be defined as

$$L(\Lambda) = -\log \prod_{r}^{R} P(p_r^+|q_r), \tag{7}$$

where Λ is the parameters in the semantic model, p_r^+ is the positive predicate of the r-th question out of R questions and $P(p_r^+|q_r)$ is the conditional probability of the positive predicate given the r-th question. All the deep semantic models used in this paper share the same loss function.

3.2.2 *BDSSM*

Long Short-Term Memory (LSTM) [35–37] is powerful to process sequence information and widely applied in natural language processing task. Compared to the simple recurrent neural network, LSTMs have the advantage of handing long-term dependencies and keeping the information from a long period of time in the memory. The architecture of the LSTM cell used in this paper similar to the one in [38]. Given an input sequence $x = (x_1, \ldots, x_T)$ the hidden state h_t at the time step t can be implemented as follows:

$$i_t = \sigma(W_{xi}x_t + U_{hi}h_{t-1} + b_i), \tag{8}$$

$$f_t = \sigma(W_{xf}x_t + U_{hf}h_{t-1} + b_f), \tag{9}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tanh(W_{xc}x_t + U_{hc}h_{t-1} + b_c), \tag{10}$$

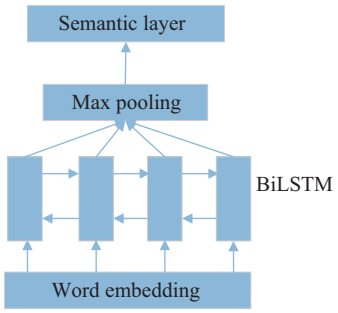

Figure 4 (Color online) The architecture of BDSSM.

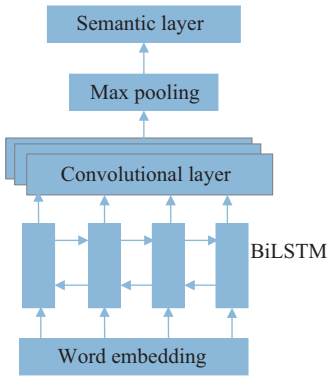

Figure 5 (Color online) The architecture of BCDSSM.

$$o_t = \sigma(W_{xo}x_t + U_{ho}h_{t-1} + b_o), \tag{11}$$

$$h_t = o_t \times \tanh(c_t), \tag{12}$$

where σ is the logistic sigmoid function, i,f,o and c are the input gate, forget gate, output gate and cell vectors. $W \in \mathbb{R}^{(H\times I)}$, $U \in \mathbb{R}^{(H\times H)}$ and $b \in \mathbb{R}^{(H\times 1)}$ are the parameters of the LSTM, where I is the dimension of the input vectors, H is the dimension of the hidden state vectors.

Though LSTMs have a powerful ability in sequence processing, single directional LSTMs can only remember the contextual information from the past tokens and ignore the information from the future tokens. In order to take the full advantage of the contextual informations coming from both the past and the future, we use bidirectional long short-time memory in our deep structure semantic model.

As shown in Figure 4, the BiLSTM Deep Structured Semantic Model (BDSSM) proposed in this paper use a bidirectional long-short term memory to model the input sequence in both forward and backward directions. At time step t, we can obtain a hidden state $\overrightarrow{h_t}$ which is from the forward direction, and a hidden state $\overleftarrow{h_t}$ which is from the backward direction. Then, the two vectors can be merged by using a concatenate operation, which is denoted as

$$h_t = \overrightarrow{h_t}||\overleftarrow{h_t}. \tag{13}$$

Similar to the CDSSM, we use a max pooling and a full connection layer to generate the semantic vector representation of the input sequence.

3.2.3 *BCDSSM*

The BDSSM is able to capture the global contextual features of a word from both past and future directions. However, the simple max pooling strategy may loss some import local features. Instead of using a simple max pooling operation to capture the global semantic features, in the BiLSTM Convolutional Deep Structured Semantic Model (BCDSSM), we apply a convolutional operation over the output sequence of the BiLSTM layer, which can utilize richer contextual information to generate the semantic representation of the input sequence, as shown in Figure 5. Then, similar to the BDSSM, we use a max pooling operation and a full connection layer to obtain the semantic representation of the input sequence.

3.3 Candidate retrieval

To make the retrieval more efficient, we store the knowledge base triples (subject, predicate, object) in an inverted index. The subject, predicate and object are indexed in separated fields. By using the CTEEM proposed in this paper, we have extracted the topic entities in questions. Then, we feed the extracted topic entities into the information retrial module to find out the related candidate fact triples in knowledge base. The topic entities are viewed as keywords to search over the subject filed of the KB index, which can link the topic entity to the subjects in knowledge base. Thus, we can obtain the

Table 2 The comparison of the candidates obtained by using the extracted topic entity and n-gram method

	CTEEM method	n-gram method
Total number	6	3000
Candidates	飞廉状风毛菊|||别名|||飞廉状风毛菊 飞廉状风毛菊|||中文名|||飞廉状风毛菊 飞廉状风毛菊|||门|||被子植物门 飞廉状风毛菊|||国内分布|||陕西省, 甘肃省, 四川省 飞廉状风毛菊|||生境|||疏林中 飞廉状风毛菊|||栽培|||非人工引种栽培	于中国|||别名|||于中国 于中国|||中文名|||于中国 于中国|||国籍|||中国 于中国|||职业|||高级工程师 于中国|||毕业院校|||吉林大学 飞廉状风毛菊|||别名|||飞廉状风毛菊 飞廉状风毛菊|||中文名|||飞廉状风毛菊 飞廉状风毛菊|||门|||被子植物门 飞廉状风毛菊|||国内分布|||陕西省, 甘肃省, 四川省 飞廉状风毛菊|||生境|||疏林中 飞廉状风毛菊|||栽培|||非人工引种栽培 分布(汉语词汇)|||别名|||分布 分布(汉语词汇)|||中文名|||分布 . . . . .

most relevant fact triples whose subject is most similar to the topic entity and obviously decrease the amount of noising candidates. And, we can substantially improve the matching performance by focus on fewer but more relevant candidates. Meanwhile, the calculation amount of the candidate ranking can be significantly reduced. For example, given the question "飞廉状风毛菊分布于中国的哪些省?", we can extract the topic entity "飞廉状风毛菊" by using the CTEEM. Then, this topic entity is used to retrieve the candidate fact triples over the subject field. Table 2 shows comparison of the retrieval results by using the extracted entity and conventional n-gram method. We limit the amount of candidates to top 3000 in the retrieval system. By using the n-gram method, we obtain about 3000 candidate fact triples most of which are irrelevant to the question. However, by using the proposed CTEEM method, we only obtained 6 relevant candidates. Since the extracted topic entities are not always correct, if we only use the extracted topic entities to retrieve the candidates, we cannot obtain the candidates of the questions whose extracted topic entities are wrong. To address this problem, we first apply a full match between the topic entity and the subject field. Then, if no results returned, we use n-gram method to obtain the candidates.

3.4 Answer selection

In answer selection stage, we attempt to rank the candidates generated from the candidate retrieval stage. Specially, we match the question with predicates of the candidates. In this paper, both the semantic and lexical level similarity are applied to improve the matching performance.

3.4.1 *Semantic matching score*

We use the deep structured semantic models describe in this paper to learn the semantic presentation of the questions and predicates. Then, we use a cosine similarity function to compute the semantic similarity between a question and each predicate of its candidates. The Semantic Matching Score (SMS) between a question q and a candidate predicate p is denoted as

$$\mathrm{SMS}(q,p) = \cos(y_q, y_p) = \frac{y_q^{\mathrm{T}} y_p}{\|y_q\|\|y_p\|}, \tag{14}$$

where y_q is the semantic vector of the question q and y_p is the semantic vector of the candidate predicate p.

By using different Deep Structured Semantic Models, we can obtain different semantic representation for the questions and the predicates. To take full advantage of all the semantic models, we use a combined

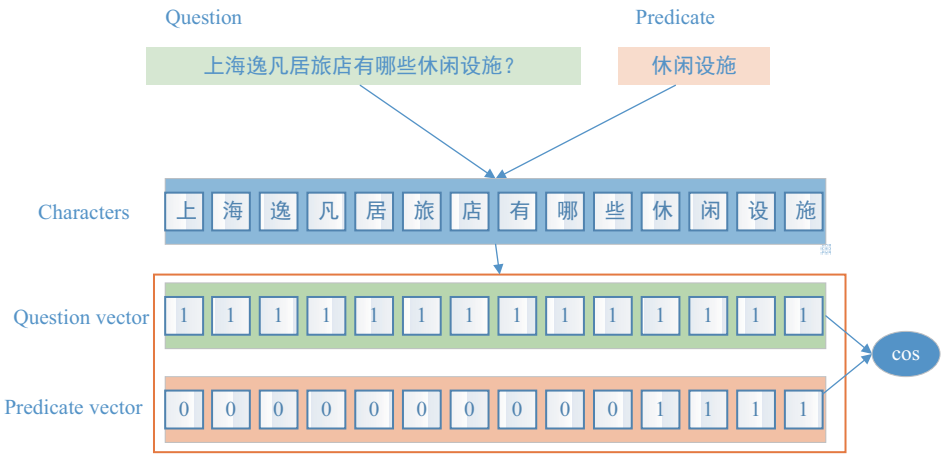

Figure 6 (Color online) An example of the lexical level matching.

semantic score which is defined as

$$\mathrm{SMS}_{\mathrm{combined}}(q,p) = \alpha \mathrm{SMS}_{\mathrm{b}}(q,p) + \beta \mathrm{SMS}_{\mathrm{bc}}(q,p) + \lambda \mathrm{SMS}_{\mathrm{c}}(q,p), \tag{15}$$

where $\mathrm{SMS}_{\mathrm{b}}$ is the semantic matching score of the BDSSM, $\mathrm{SMS}_{\mathrm{bc}}$ is the semantic matching score of the BCDSSM, $\mathrm{SMS}_{\mathrm{c}}$ is the semantic matching score of the CDSSM. α, β and λ are the weights of these three semantic matching scores and $\alpha + \beta + \lambda = 1$.

3.4.2 *Lexical matching score*

The DSSMs can be used to match the question and predicates in semantic level which can handle the mismatch problem between the question and predicate. However, the questions also may overlap with the predicates in knowledge base. For example, the predicate of the question "上海逸凡居旅店有哪些休闲设施?" in knowledge base is "休闲设施". Both the question and the predicate contain the word "休闲设施". So the lexical similarity is also very important for the answer selection. Following [17], we use the character level similarity to measure the lexical level matching score. As shown in Figure 6, we collect the characters in the question and predicate as a bag of characters, then the presentation of the question and predicate are built based on the bag of characters. The Lexical Matching Score (LMS) is computed as

$$\mathrm{LMS}(q,p) = \cos(c_q, c_p) = \frac{c_q^{\mathrm{T}} c_p}{\|c_q\| \|c_p\|}, \tag{16}$$

where c_q and c_p are the representation of the question and predicate, respectively.

We combine the semantic matching and the lexical matching scores into a final matching score, which is defined as follows:

$$\mathrm{score} = \mathrm{SMS}(q,p) + \omega \mathrm{LMS}(q,p), \tag{17}$$

where $\mathrm{SMS}(q,p)$ is the semantic matching score, $\mathrm{LMS}(q,p)$ is the lexical matching score, and ω is the coefficient parameters of $\mathrm{LMS}(q,p)$ which is used to control the weight of the lexical matching score.

4 Experiment

4.1 Dataset

The dataset used in this paper is released by the task of KBQA hosted by NLPCC-ICCPOL 2016, including a knowledge base which contains 43 M knowledge triples, 14609 question-answer pairs for training and 9870 questions to be answered. Each fact triple in knowledge base is formatted as: Subject ||| Predicate ||| Object. Following [17], we use the question-answer pairs to generate the training dataset

Table 3 The experimental results on the test dataset

System		Average F_1(%)
Compared methods	Baseline system (CDSSM)	52.47
	Lai et al., 2016 [14]	82.47
	Yang et al., 2016 [16]	81.59
	Wang et al., 2016 [15]	79.14
	Xie et al., 2016 [17]	79.57
Ours	CTEEM+LMS	74.62
	CTEEM+CDSSM	75.74
	CTEEM+BDSSM	74.22
	CTEEM+BCDSSM	76.18
	CTEEM+Combined-DSSM	77.89
	CTEEM+CDSSM+LMS	81.89
	CTEEM+BDSSM+LMS	81.63
	CTEEM+BCDSSM+LMS	82.21
	CTEEM+Combined-DSSM+LMS	**82.43**

for the CTEEM and the DSSMs. We use the Average F_1 score to measure the quality of the KBQA which is defined in [39].

4.2 Setup

By using word2vec [19], we use the knowledge base as the dataset to pre-train a word embedding of 200 dimension. The pre-trained word embedding is used as initial parameters of a embedding layer and will be updated during the training time. We use the Adam [40], which can automatically adjust the learning rate during training to optimize the objective function. The size of semantic representation generated by the DSSMs is set to 128. The smoothing factor λ in (6) is set to 5. For each positive predicate we randomly sampled 5 negative predicates to compute the objective function of DSSMs. We limit the length of the question to 20 and the predicate to 5 by using a padding operation (pad when less than this range and discard when out of this range), so as to train the model in mini-batches.

4.3 Experimental results

In this subsection, we analyse the experimental results, as shown in Table 3. We compare our results with the baseline system released by the NLPCC-ICCPOL 2016 KBQA task and some existing state-of-the-art systems, as shown in Table 3. The baseline system is based on CDSSM without using topic entity extraction models. Experimental results show that our proposed method [17] achieves the third place among the 21 systems on NLPCC-ICCPOL 2016 KBQA challenge task [39]. The method proposed in [14] achieves the first place and the method proposed in [16] achieves the second place. The improved method proposed in this paper achieves a much better result than most of the state-of-the-art methods.

We also compare the results of different matching methods described in this paper, including lexical matching method and different semantic matching methods. The Lexical Matching Score can be computed by using (16). The semantic matching score can be obtained by using different DSSMs described in the paper, including CDSSM, BDSSM and BCDSSM. The semantic matching score and lexical matching score are combined by using (17). The Combined-DSSM use the combined semantic matching score of the three DSSMs as the Semantic Matching Score, as defined in (15). In our experiment, we use 10000 data for training and 4609 data for validating to choose the value of hyper-parameters. Finally, the parameters in (15) are set to $\alpha = 0.1$, $\beta = 0.6$ and $\lambda = 0.3$. the weight of the LMS ω in (17) is set to $\omega = 1.2$. The experimental results are shown in Table 3.

Compared to the baseline system, which is based on CDSSM and without using any topic entity extraction method, our approach with using the proposed CTEEM outperforms the baseline system by a large margin. This illustrates that the topic entity extraction model plays a very important role in the question answering system.

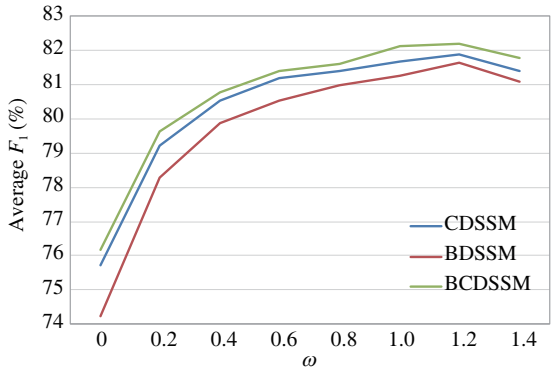

Figure 7 (Color online) The average F_1 vs. ω curves of the three semantic matching methods.

When we only using the simple lexical matching score (LMS) to rank the candidates, the Average F_1 score is 74.62%. This indicates that most of the questions in the dataset can be answered by using the lexical level matching. However, there are still lots of questions that cannot match with the predicates in knowledge base in lexical level. So we need to deal with the mismatch problem by using semantic level matching. Among the three semantic matching scores, the proposed BCDSSM can achieve an Average F_1 score of 76.18%, which is better than the CDSSM and the BDSSM. This means that the proposed BCDSSM can learn better semantic representation for the questions and the predicates in knowledge base than the CDSSM and BDSSM. The Combined-DSSM, which combine the different semantic matching scores, achieves the best result of all the models. And when we consider both the semantic matching score and the lexical matching score, the Average F_1 score can be obviously improved, which indicates that both the lexical semantic matching score and the semantic matching score are very important. For example, the Average F_1 score of the BCDSSM increases from 76.18% to 82.21% when integrate the lexical matching score. The best result of our methods can achieve an Average F_1 of 82.43%, which is comparable to the state-of-the-art results.

Figure 7 shows the Average F_1 vs. ω curves of the three semantic matching methods used in this paper. Here, ω is the parameter in (17) which is the weight of the lexical matching score. When the weight of the lexical matching score is increasing, the Average F_1 score can be substantially improved. This is mainly because that plenty of the questions in the dataset overlap with the predicates in knowledge base and the similarity between the question and the predicate can be measure the by using lexical similarity. And sometimes deep semantic level representation may loss some very important lexical features. So by combining the lexical level similarity with the semantic level similarity, we can match the question and the predicate better.

4.4 Error analysis

In order to better understand how our system works, we random sample 100 examples from the test data set to analyze the errors. The error types can be generally categorized into the following three classes.

Question ambiguity: Many questions contains a topic entity, which may be linked to many different subject entities in knowledge base. For example, for question "白沙镇的车牌代码是多少?", there are a lot of different entities "白沙镇" in knowledge base. In this situation, such question does not provide enough information to distinguish which subject is the one that the question asks for. Therefore, it is difficult to rank the candidates.

Candidate retrieval error: This type of errors occurs when the topic entity is extracted incorrectly or the topic entity in question cannot be successfully linked to the subject in knowledge base. Hence, we cannot obtain the desired candidate triples.

Predicate matching error: Even we use both the semantic level score and the lexical level score to match the questions and predicates in knowledge base, some errors are still caused by the mistakes of the predicate matching.

5 Conclusion

We propose a topic enhanced deep structured semantic model for KBQA. The proposed method consists of two stages to match questions with the fact triples in knowledge base, which consider the task of KBQA as a matching problem between questions and the subjects and predicates in knowledge base. In candiate retrieval stage, we present a novel convolutional based topic entity extraction model to automatically extract topic entities in questions. Then, we use the extracted topic entities to retrieve relevant candidates, which can be viewed as a matching between questions and the subjects in knowledge base. In answer selection stage, we employ the deep structured semantic model to measure the semantic similarity between questions and predicates. Moreover, we apply lexical matching score to improve the performance of the matching between questions and predicates. Experimental results show that our proposed method achieves the third place among the 21 systems on NLPCC-ICCPOL 2016 KBQA challenge task [39]. Furthermore, we also extend the DSSM by using BiLSTM and integrate a convolutional structure on the top of BiLSTM layers. Our experimental results show that the extension models can further improve the performance and achieve better result compared to most state-of-the-art methods.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61573163, 71571084), Fundamental Research Funds for the Central Universities (Grant No. CCNU16A02024), and Wuhan Youth Science and Technology Plan.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, 2008. 1247–1250
2 Auer S, Bizer C, Kobilarov G, et al. Dbpedia: a nucleus for a web of open data. In: Proceedings of the 6th International the Semantic Web and 2nd Asian Conference on Asian Semantic Web, Busan, 2007. 722–735
3 Berant J, Chou A, Frostig R, et al. Semantic parsing on freebase from question-answer pairs. Proc EMNLP, 2013, 2: 1533–1544
4 Berant J, Liang P. Semantic parsing via paraphrasing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Stroudsburg, 2014. 1415–1425
5 Bordes A, Usunier N, Chopra S, et al. Large-scale simple question answering with memory networks. Comput Sci, 2015, arXiv:1506.02075
6 Bordes A, Chopra S, Weston J. Question answering with subgraph embeddings. Comput Sci, 2014, arXiv:1406.3676
7 Bordes A, Weston J, Usunier N. Open question answering with weakly supervised embedding models. In: Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, 2014. 165–180
8 Yao X, Durme B V. Information extraction over structured data: question answering with freebase. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Stroudsburg, 2014. 956–966
9 Shen Y, He X, Gao J, et al. Learning semantic representations using convolutional neural networks for web search. In: Proceedings of the 23rd International Conference on World Wide Web, Seoul, 2014. 373–374
10 Zettlemoyer L S, Collins M. Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, Edinburgh, 2012
11 Kwiatkowski T, Zettlemoyer L, Goldwater S, et al. Inducing probabilistic CCG grammars from logical form with higher-order unification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, 2010. 1223–1233
12 Liang P, Jordan M I, Klein D. Learning dependency-based compositional semantics. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Stroudsburg, 2011. 590–599
13 Cai Q, Yates A. Large-scale semantic parsing via schema matching and lexicon extension. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, 2013. 423–433
14 Lai Y, Lin Y, Chen J, et al. Open domain question answering system based on knowledge base. In: Proceedings of the 24th International Conference on Computer Processing of Oriental Languages, Kunming, 2016. 722–733
15 Wang L, Zhang Y, Liu T. A deep learning approach for question answering over knowledge base. In: Proceedings of the 24th International Conference on Computer Processing of Oriental Languages, Kunming, 2016. 885–892
16 Yang F, Gan L, Li A, et al. Combining deep learning with information retrieval for question answering. In: Proceedings of the 24th International Conference on Computer Processing of Oriental Languages, Kunming, 2016. 917–925
17 Xie Z, Zeng Z, Zhou G, et al. Knowledge base question answering based on deep learning models. In: Proceedings of

the 24th International Conference on Computer Processing of Oriental Languages, Kunming, 2016. 300–311
18 Pennington J, Socher R, Manning C D. Glove: global vectors for word representation. Proc EMNLP, 2014, 14: 1532–1543
19 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. Adv Neural Inform Process Syst, 2013, 26: 3111–3119
20 Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. Comput Sci, 2013, arXiv:1301.3781
21 Zhou G Y, He T T, Zhao J, et al. Learning continuous word embedding with metadata for question retrieval in community question answering. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, 2015. 250–259
22 Zhou G Y, Huang X J. Modeling and learning distributed word representation with metadata for question retrieval. IEEE Trans Knowl Data Eng, 2017, 29: 1226–1239
23 Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, San Francisco, 2013. 2333–2338
24 Shen Y, He X, Gao J, et al. A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, 2014. 101–110
25 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. Comput Sci, 2014, arXiv:1409.0473
26 Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. Proc EMNLP, 2015
27 Yih W, Chang M W, He X, et al. 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, Beijing, 2015
28 Yih W T, He X, Meek C. Semantic parsing for single-relation question answering. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, 2014. 643–648
29 Dong L, Wei F, Zhou M, et al. Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, 2015. 260–269
30 Zhang Y, Liu K, He S, et al. Question answering over knowledge base with neural attention combining global knowledge information. Comput Sci, 2016, arXiv:1606.00979
31 Jain S. Question answering over knowledge base using factual memory networks. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), San Diego, 2016. 109–115
32 Dai Z H, Li L, Xu W. Cfo: conditional focused neural question answering with large-scale knowledge bases. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016
33 Lafferty J, McCallum A, Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning, San Francisco, 2001. 282–289
34 Palangi H, Deng L, Shen Y, et al. Semantic modelling with long-short-term memory for information retrieval. Comput Sci, 2014, arXiv:1412.6629
35 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780
36 Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput, 2000, 12: 2451–2471
37 Gers F A, Schraudolph N N, Schmidhuber J. Learning precise timing with LSTM recurrent networks. J Mach Learn Res, 2002, 3: 115–143
38 Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, 2013. 6645–6649
39 Duan N. Overview of the NLPCC-ICCPOL 2016 shared task: open domain chinese question answering. In: Proceedings of the 24th International Conference on Computer Processing of Oriental Languages, Kunming, 2016. 942–948
40 Kingma D, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2014