IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 3, 2019, accepted April 2, 2019, date of publication April 9, 2019, date of current version April 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909826

Knowledge Base Question Answering With Attentive Pooling for Question Representation

RUN-ZE WANG[1], ZHEN-HUA LING[1], (Senior Member, IEEE), AND YU HU [1,2,3]

[1]National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China
[2]iFLYTEK Research, Hefei 230088, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai 200031, China

Corresponding author: Zhen-Hua Ling (zhling@ustc.edu.cn)

ABSTRACT This paper presents a neural network model for a knowledge base (KB)-based single-relation question answering (SR-QA). This model is composed of two main modules, i.e., entity linking and relation detection. In each module, an embedding vector is computed from the input question sentence to calculate its similarity scores with entity candidates or relation candidates. This paper focuses on attention-based question representation in SR-QA. In the entity linking module, two attentive pooling methods, inner-sentence attention and structure attention, are employed to derive question embeddings, and their performances are compared in experiments. In the relation detection module, a new attentive pooling structure, named multilevel target attention (MLTA), is proposed to utilize the multilevel descriptions of relations. In this structure, the attention weights for aggregating the hidden states of question sentences are calculated using relation candidates as queries at the relation level, word level, and character level. Then, the similarity scores for relation detection are computed by matching questions to relation candidates at all three levels. The experimental results show that our proposed model achieves a state-of-the-art accuracy of 82.29% on the simple questions dataset. Furthermore, the results of ablation tests demonstrate the effectiveness of our proposed MLTA method for question representation.

INDEX TERMS Knowledge base question answering, entity linking, relation detection, attention mechanism.

I. INTRODUCTION

Question answering is a popular natural language processing (NLP) topic with a long research history. It has various forms, such as synthetic question answering [1], reading comprehension [2] and knowledge base-based question answering (KB-QA), according to what kinds of resources are utilized to generate or extract answers. This paper studies the KB-QA tasks that aim at answering factoid natural language questions using the triples in knowledge bases (KBs), such as Freebase [3] and DBpedia [4].

The KB-QA tasks can be further divided into two main categories, i.e., single-relation question answering (SR-QA) [5]–[8] and multi-relation question answering (MR-QA) [9]–[11], according to how many triples in KBs are necessary to answer each question. The facts in Freebase and other KBs are usually organized as subject-relation-object triples, such as (*Avatar*, */film/film/directed_by*, *James Cameron*), where both subject and object are usually entities. Each single-relation question can be answered using only one triple. For example, to answer the question "*Where was Barack Obama born*?", people only need to find the triple (*BarackObama*, */people/person/place_of_birth*, *the United States*) and then obtain the final answer "*the United States*". Single-relation questions are the most common form of questions found in search query logs and community question answering websites [5]. Different from SR-QA, MR-QA contends with more complicated questions that need more than one triple to obtain each answer.

This paper focuses on KB-based SR-QA tasks and presents an end-to-end neural network model for SR-QA. According to previous studies [6], [8], [12], the models for SR-QA usually involve two main procedures, i.e., entity linking and relation detection. Each question always corresponds to a topic entity, i.e., a subject of a triple in KBs that represents the topic of the question (e.g., a person, a place, a film, etc.). The entity linking procedure aims to find the topic entity from a set of candidates for each question. Previous methods for

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanyu Ma.

entity linking include parsing the input questions into logical forms [6], [13]–[15] or representing questions and entities with embedding vectors for similarity scoring [8], [16]. This paper follows the framework of vector space modeling and converts each question into an embedding vector to find its topic entity. Two attentive pooling methods, inner-sentence attention [17] and structure attention [18], are employed to derive question embeddings, and their performances are compared by experiments.

After entity linking, relation detection is concerned with identifying a KB relation that each question refers to. The main challenge of relation detection is that relations are usually expressed in diversified and implicit ways. In previous studies, the question embeddings for relation detection were usually computed by simple mean/max pooling after sequential encoding [12], [19], [20] or attentive pooling using the predicted topic entities as queries [8], [16]. This paper proposes a multilevel target attention (MLTA) method to improve the question representation for relation detection. Instead of using topic entities as queries for attentive pooling [8], [16], MLTA adopts the embeddings of relation candidates as queries to calculate attention weights. This approach leads to relation-dependent question embeddings and helps to specifically match the question to each relation candidate. Furthermore, the hierarchical information carried by each relation is utilized to generate relation embeddings at the character level, word level and relation level. Finally, the similarity score for relation detection is calculated between the multilevel embedding vectors of each relation candidate and the relation-dependent embedding vector of the question. In our proposed model, the entity linking procedure and the relation detection procedure share the same bidirectional LSTM (BiLSTM) layers for question encoding, and all model parameters are estimated simultaneously using training samples.

Our main contributions in this paper are threefold. First, an end-to-end neural network model for knowledge base-based single-relation question answering is designed. Second, a multilevel target attention method is proposed, which derives relation-dependent question embeddings utilizing the multilevel descriptions of relations. Third, a state-of-the-art accuracy of 82.29% on the SimpleQuestions dataset [7] is achieved using our proposed method.

In the next section, some related studies on Freebase, KB-QA and attention-based sentence embedding are briefly reviewed. Section III describes the proposed model architecture in detail. Section IV introduces the implementations of our proposed methods for training and inference. The experimental results and analysis are shown in Section V. Section VI draws conclusions and discusses our future work.

II. RELATED WORK

A. THE STRUCTURE OF FREEBASE

In recent years, many studies were conducted to utilize well-structured knowledge bases as external resources to support question answering. The knowledge bases can be built manually or by mining text corpora [21], [22]. Freebase is one of the most popular knowledge bases used in KB-QA tasks, which is a collaboratively created graph database for structuring human knowledge. It contains more than 125,000,000 tuples, more than 4,000 types and more than 7,000 properties. Because the HTTP-based graph-query API of Freebase has been closed, recent studies use Freebase through the data dump file[1] directly, which contains more than 3 billion facts. The facts in the Freebase dump file are represented by subject-relation-object triples, such as ''/m/0f819c, /location/country/capital, /m/05qtj''. The subject is an internationalized resource identifier (IRI) or a blank node. The relation is usually an IRI. The object is an IRI, a literal value or a blank node. Noticing that all the subjects and objects in triples are not represented as natural language, researchers need to translate them into natural language with the help of a common relation ''/type/object/name''. For example, the entity ''/m/0f819c'' can be translated into ''France'' using the triple ''/m/0f819c, /type/object/name, France''. The translated Freebase triples are widely used in many KB-related tasks, such as knowledge graph reasoning, link prediction and KB-QA.

The relations in Freebase are organized using a hierarchical structure as *domain* → *type* → *topic*. A domain or a type is usually represented by a single word. A topic can be further split into multiple words. For example, the relation with a name ''/people/person/place_of_birth'' belongs to the domain of ''people'' and the type of ''person''. Its topic is ''place_of_birth'' which contains three words, i.e., ''place'', ''of'' and ''birth''. In this paper, the relation-level, word-level and character-level descriptions of relations are utilized to design the MLTA method for relation detection.

B. KB-QA

The goal of KB-QA is to automatically extract answers from a given knowledge base for input questions. There are two conventional approaches to KB-QA, semantic parsing [23]–[27] and information retrieval [7], [10], [11], [14], [28], [29]. The first approach constructs a semantic parser to convert each natural language question into a structured expression, e.g., a logical form, to obtain the answer. The second approach searches answers from knowledge bases using the text information conveyed by questions.

With the development of deep learning in recent years, neural networks have been introduced to KB-QA [8]. A widely used dataset for KB-based single-relation question answering (SR-QA) research is SimpleQuestions, which was proposed by Bordes *et al.* [7]. They also set up a baseline model for this dataset using memory networks (MemNN). The MemNN model first parsed Freebase and stored it in memory. Then, the similarity scores between the input questions and the Freebase facts were calculated to obtain answers. He *et al.* [8] proposed a character-level encoder-decoder neural network

[1] `https://developers.google.com/freebase/data`.

with attentions for SR-QA. An attentive long short-term memory (LSTM) layer [30] was adopted to encode each input question, and another two-layer LSTM was built for decoding in order to determine the best topic entity and relation among all candidates. Yin *et al.* [19] improved He's work with an attentive max-pooling convolutional neural network (AMPCNN). The AMPCNN model also encoded each input question into a semantic vector that was further fed into the ranking process. Lukovniokov *et al.* [20] encoded each word in the input questions with gated recurrent units (GRU) [31] at both the character level and word level and then fed them into another GRU layer to generate the final semantic vector for each question. Mohammed *et al.* [15] tested different models for subject and relation encoding, and they finally chose bidirectional LSTM (BiLSTM) as their subject encoder and bidirectional GRU as their relation encoder. Yu *et al.* [12] focused on relation detection, and they encoded questions at the word level using a two-layer BiLSTM and encoded the relations at both the relation level and word level. Zhang *et al.* [32] improved Yu's work and still focused on relation detection. They encoded questions with a soft attention by treating the relation words as queries. However, they just considered word-level representations, and the attention results were fed into a comparison CNN layer for further feature extraction.

This paper follows the neural network-based approach to SR-QA and focuses on attention-based question representation. The details are introduced in the next subsection.

C. ATTENTION-BASED SENTENCE EMBEDDING

Attention mechanisms have been successfully applied to obtain sentence embeddings in many NLP tasks. With the help of an attention mechanism, the sentence embedding module can flexibly select informative parts of input sentences to derive sentence representations. The related studies can be divided into two categories: the attentive neural network (i.e., designing a neural network with attention to generate a sequence of hidden representations for an input sentence) and attentive pooling (i.e., employing attention as a pooling method to represent each sentence as a single vector).

Cheng *et al.* [33] proposed an attentive neural network called LSTMN that calculated the attention between a certain word and its previous words at each step of an LSTM. Vaswani *et al.* [34] processed a self-attention network which used multihead scaled dot product attention to represent each word in sentences. Shen *et al.* [35] integrated the attention mechanism into reinforcement learning and proposed a reinforcement self-attention network.

Regarding attentive pooling, Liu *et al.* [17] proposed an inner-sentence attention method that calculated scalar attention weights between every word in a sentence and their mean-pooling vector. Lin *et al.* [18] proposed a structure attention method for sentence embedding that utilized only LSTM states as inputs and calculated attention weights using a multilayer perceptron (MLP). Chen *et al.* [36] proposed a generalized pooling method, which included vector-based multihead self-attention and some penalization terms.

In previous studies on SR-QA, the single vector representations of input questions were usually obtained by applying mean pooling or max pooling to the output of sequential question encoders [12], [19]. There were very few studies that adopted attentive pooling to build the question encoders for SR-QA. One example is that He and Gohub [8] used a zero vector and the predicated topic entity as attention queries for attentive pooling in entity linking and relation detection, respectively. Another related study is the cross-attention approach [9], which was proposed for MR-QA and utilized the characteristics of target answers as queries for the attentive pooling of input questions.

This paper investigates the effectiveness of applying attentive pooling methods to question representation in SR-QA. In the entity linking module, inner-sentence attention [17] and structure attention [18] are employed to derive the embedding vectors of questions, and their performances are compared in experiments. In the relation detection module, we propose a new attention structure, named multilevel target attention (MLTA), to make better use of the hierarchical information carried by relations. In this structure, the attention weights for aggregating the hidden states of question sentences are calculated using each relation candidate as a query at the relation level, word level and character level, and relation-dependent question embeddings are obtained. A similar idea of utilizing the multitype data of entities was developed for obtaining entity embeddings in previous work [37]. Compared with the method proposed by He and Gohub [8], the MLTA method adopts relation candidates instead of topic entities as queries for the attentive pooling of questions. Compared with the cross-attention approach [9], this paper focuses on SR-QA instead of MR-QA, and our MLTA method further utilizes the multilevel descriptions of relations for question representation.

III. MODEL ARCHITECTURE

A. OVERVIEW

The overall flowchart of our proposed SR-QA method is shown in Fig. 1. The entity linking module first determines the optimal subject (i.e., topic entity) by calculating similarity scores between the embedding vector of the input question and the embedding vectors of all topic entity candidates. Then, the relations that link with the optimal subject are extracted from KB as relation candidates and are represented as embedding vectors at the relation level, word level and character level. These multilevel representations of relation candidates are used to derive relation-dependent question embeddings and to calculate the similarity scores between each question and its relation candidates. The relation candidate with the highest similarity score is selected to determine a triple in the KB that can answer the input question.

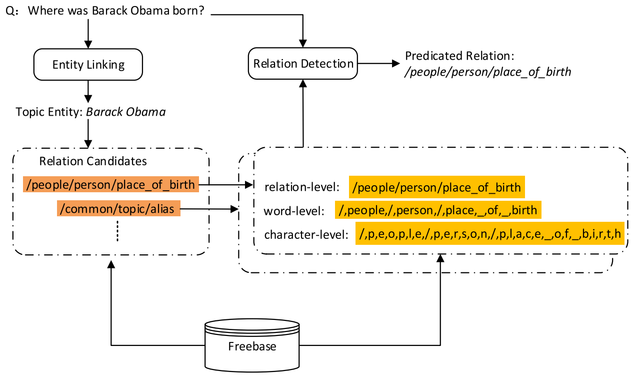

Q: Where was Barack Obama born?
Entity Linking
Relation Detection
Predicated Relation:
/people/person/place_of_birth
Topic Entity: Barack Obama
Relation Candidates
/people/person/place_of_birth
/common/topic/alias
relation-level: /people/person/place_of_birth
word-level: /,people,/,person,/,place,_,of,_,birth
character-level: /,p,e,o,p,l,e,/,p,e,r,s,o,n,/,p,l,a,c,e,_,o,f,_,b,i,r,t,h
Freebase

FIGURE 1. The overall flowchart of our proposed SR-QA method. After entity linking and relation detection, a triple in Freebase is determined that can answer the input question.

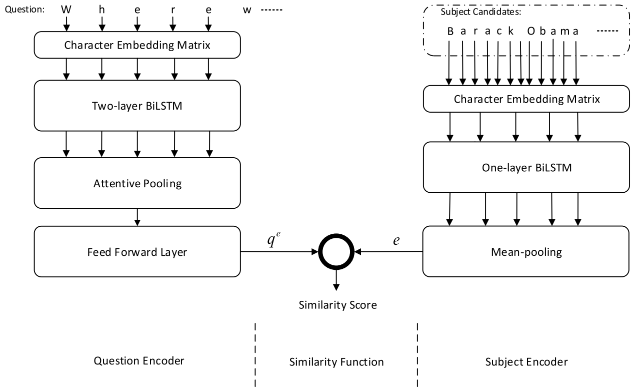


FIGURE 2. The model architecture of the entity linking module.

B. ENTITY LINKING

The entity linking module is composed of three main parts, i.e., question encoder for generating question embeddings, subject encoder for generating subject embeddings and a similarity function for calculating the similarity scores between each question and its subject candidates. Its detailed architecture is shown in Fig. 2.

1) QUESTION ENCODER

To address the out-of-vocabulary problem, each question is written as a character sequence that forms the input of the question encoder. A character embedding matrix $E_c \in \mathbb{R}^{v_c \times d}$ is employed to obtain the embedding vector for each character. Here, d means the dimension of character embeddings, and v_c is the vocabulary size. E_c is randomly initialized and updated during training. Then, the character embeddings are fed into a two-layer BiLSTM [38]. The output $\boldsymbol{q}_t$ of the second-layer of BiLSTM at the t-th step is treated as the representation of the t-th character in the question, which can be written as

$$\boldsymbol{q}_t = [\boldsymbol{q}_t^f; \boldsymbol{q}_t^b] = [\varphi_t^1, \cdots, \varphi_t^{\frac{d}{2}}, \varphi_t^{\frac{d+1}{2}}, \cdots, \varphi_t^d], \tag{1}$$

where $\boldsymbol{q}_t^f$ and $\boldsymbol{q}_t^b$ stand for the outputs of forward and backward units in BiLSTM, respectively, and the dimension d of BiLSTM outputs is identical to the dimension of character embeddings.

After obtaining $\boldsymbol{q}_t$, a pooling process is necessary to aggregate all character embedding vectors into a single question embedding vector. Two attentive pooling methods are compared in this part as follows.

- **Inner-sentence attention** The inner-sentence attention method was proposed by Liu *et al.* [17]. When applying it to the question encoder for entity linking, we first calculate the average vector of all characters in the question as

$$\bar{\boldsymbol{q}} = \frac{\sum_{t=1}^{N_q} \boldsymbol{q}_t}{N_q}, \tag{2}$$

where N_q is the number of characters in the question. Then, the attention weights $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_{N_q}]^\top$ are calculated using a perceptron between the average vector and each character vector and are further used to obtain the single vector representation $\tilde{\boldsymbol{q}}^e$ of the input question. The calculations can be formulated as

$$\boldsymbol{\alpha} = softmax(\boldsymbol{v}_a^\top tanh(\boldsymbol{W}_a \boldsymbol{q}_t + \boldsymbol{U}_a \bar{\boldsymbol{q}})), \tag{3}$$

$$\tilde{\boldsymbol{q}}^e = \sum_{t=1}^{N_q} \alpha_t \boldsymbol{q}_t, \tag{4}$$

where $\{\boldsymbol{W}_a, \boldsymbol{U}_a\} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{v}_a \in \mathbb{R}^{m \times 1}$.

- **Structure attention** The structure attention method was proposed by Lin *et al.* [18]. In our implementation, the weights of structure attention with multiple heads are calculated as

$$\boldsymbol{A} = softmax(\boldsymbol{V}_a^\top tanh(\boldsymbol{W}_a \boldsymbol{q}_t)), \tag{5}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{m \times d}$, $\boldsymbol{V}_a \in \mathbb{R}^{m \times K}$, $\boldsymbol{A} \in \mathbb{R}^{K \times N_q}$, and K denotes the number of heads. Then, the vector representation of the input question is calculated as

$$\tilde{\boldsymbol{q}}^e = reshape(\boldsymbol{A}[\boldsymbol{q}_1, \cdots, \boldsymbol{q}_{N_q}]), \tag{6}$$

where the function $reshape()$ converts a matrix of K rows and d columns into a vector of Kd elements.

After attentive pooling using either inner-sentence attention or structure attention, a feedforward layer is applied to $\tilde{\boldsymbol{q}}^e$ to obtain the final question representation $\boldsymbol{q}^e \in \mathbb{R}^d$ for entity linking. The feedforward layer is designed following the one used in He's work [8].

2) SUBJECT ENCODER

Because the entity names in KBs are usually simple words of phrases, a character-level one-layer BiLSIM together with a mean-pooling layer are adopted to extract the embedding vector $\boldsymbol{e}_i \in \mathbb{R}^d$ for the i-the subject in the candidate set, as shown in Fig. 2.

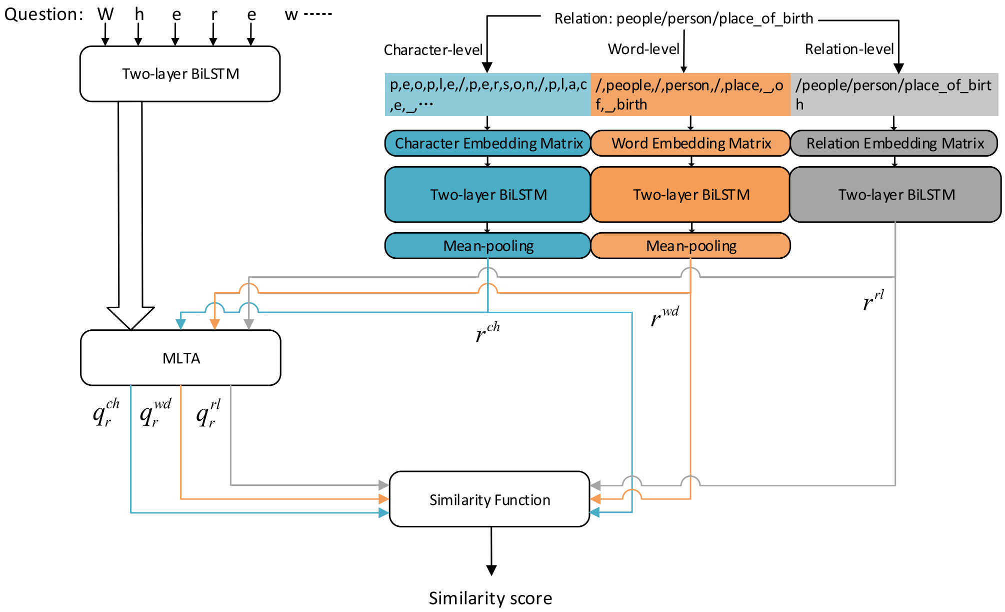


FIGURE 3. The model architecture of the relation detection module with our proposed multilevel target attention (MLTA).

3) SIMILARITY FUNCTION

Finally, we calculate the similarity score SE_i between the input question and the i-th subject candidate using a cosine similarity function together with a Softmax function, which can be written as

$$SE_i = \frac{\exp(\cos(\boldsymbol{q}^e, \boldsymbol{e}_i))}{\sum_{k=1}^{M} \exp(\cos(\boldsymbol{q}^e, \boldsymbol{e}_k))}, \tag{7}$$

where M is the number of subject candidates. The subject candidate with the highest similarity score is determined as the topic entity of the input question.

C. RELATION DETECTION

The detailed architecture of the relation detection module is shown in Fig. 3. Similar to entity linking, this module is also composed of three main parts, i.e., a multilevel relation encoder, a question encoder with multilevel target attention (MLTA), and a similarity function for relation prediction.

1) MULTILEVEL RELATION ENCODER

As shown in Fig. 3, each relation is encoded at three different levels including relation level, word level and character level. The relation level considers each relation as a whole, the word level describes each relation name as a sequence of words separated by some symbols such as "/" and "_", and the character level converts each relation name into a sequence of characters. For example, the word-level description of the relation with name "/people/person/place_of_birth" is a word sequence $\{/, people, /, person, /, place, \_, of, \_, birth\}$ and the character-level description is a character sequence $\{p, e, o, p, l, e, /, p, e, r, s, o, n, /, p, l, a, c, e, \_, o, f, \_, b, i, r, t, h\}$.

For each relation, we first apply the embedding matrices of different levels to its three-level descriptions. These embedding matrices are all initialized randomly and are updated during model training. Then, three one-layer BiLSTMs are built to encode the relation at different levels.[2] The hidden states of the forward and backward LSTMs at each step are concatenated as

$$\boldsymbol{h}_{(x)_j} = [\boldsymbol{h}^f_{(x)_j}; \boldsymbol{h}^b_{(x)_j}] = [h_j^1, h_j^2, \cdots, h_j^{\frac{d}{2}}, h_j^{\frac{d}{2}+1}, \cdots, h_j^d], \tag{8}$$

where $(x) \in \{rl, wd, ch\}$, $j \in \{1, \cdots, N_{(x)}\}$, $N_{(x)}$ stands for the sequence lengths at level (x), and N_{rl} is always 1. Further, the hidden state sequences at different levels can be written as

$$\boldsymbol{H}^r_{(x)} = [\boldsymbol{h}_{(x)_1}{}^\top, \cdots, \boldsymbol{h}_{(x)_{N_{(x)}}}{}^\top], \tag{9}$$

with $\boldsymbol{H}^r_{(x)} \in \mathbb{R}^{d \times N_{(x)}}$.

After that, the hidden state sequences are sent into a mean-pooling layer to obtain the embedding vectors $\{\boldsymbol{r}^{rl}, \boldsymbol{r}^{wd}, \boldsymbol{r}^{ch}\}$ of each relation candidate at the relation level, word level, and character level, respectively, where $\boldsymbol{r}^{(x)} \in \mathbb{R}^d$ and $(x) \in \{rl, wd, ch\}$. Using the word level as an example, the mean-pooling mechanism can be formulated as

$$r^i = \frac{\sum_{n=1}^{N_{wd}} h_n^i}{N_{wd}}, \tag{10}$$

$$\boldsymbol{r}^{wd} = [r^1, r^2, \cdots, r^d], \tag{11}$$

where N_{wd} is the word number in the candidate relation, and d is the dimension of BiLSTM outputs.

[2] To apply the unified BiLSTM structure to all three levels, the embedded relation-level description is treated as a sequence with length 1.

2) QUESTION ENCODER WITH MLTA

The question encoder for relation detection shares the character embedding matrix and the two-layer character-level BiLSTM used in the question encoder for entity linking. In other words, the sequences of $\boldsymbol{q}_t$ in Eq. (1) are also used here to derive the single-vector representations of questions for relation detection.

In previous studies on SR-QA, mean pooling or max pooling was popularly employed to obtain the embedding vector of each question for relation detection [12], [19]. To describe the similarity between the question and each relation candidate in a more specific way, a multilevel target attention (MLTA) structure is proposed in this paper. In this structure, the multilevel representations of relation candidates are employed as queries to aggregate the sequence of $\boldsymbol{q}_t$ into a single vector. Thus, relation-dependent question embeddings are obtained.

For each relation candidate, its embedding vectors $\{\boldsymbol{r}^{rl}, \boldsymbol{r}^{wd}, \boldsymbol{r}^{ch}\}$, given by the multilevel relation encoder, are adopted as the query vectors to calculate the attention weights at the relation level, word level, and character level, respectively. The calculation of attention weights can be written as

$$\boldsymbol{\alpha}^{(x)} = softmax(\boldsymbol{v}_a^{(x)\top} tanh(\boldsymbol{W}_a^{(x)}\boldsymbol{q}_t + \boldsymbol{U}_a^{(x)}\boldsymbol{r}^{(x)})), \tag{12}$$

where $(x) \in \{rl, wd, ch\}$, $\{\boldsymbol{W}_a^{(x)}, \boldsymbol{U}_a^{(x)}\} \in \mathbb{R}^{m\times d}$, and $\boldsymbol{v}_a^{(x)} \in \mathbb{R}^{m\times 1}$. Subsequently, these attention weights are employed to derive the embedding vectors of the question as follows:

$$\boldsymbol{q}_r^{(x)} = \sum_{t=1}^{N_q} \alpha_t^{(x)} \boldsymbol{q}_t, \tag{13}$$

where $\boldsymbol{q}_r^{(x)} \in \{\boldsymbol{q}_r^{rl}, \boldsymbol{q}_r^{wd}, \boldsymbol{q}_r^{ch}\}$ stand for the embedding vectors of the question for matching the relation candidate at the relation level, word level, and character level, respectively.

3) SIMILARITY FUNCTION FOR RELATION PREDICTION

For each relation candidate, its similarity scores with the input question are first calculated at different levels using $\{\boldsymbol{q}_r^{rl}, \boldsymbol{q}_r^{wd}, \boldsymbol{q}_r^{ch}\}$ and $\{\boldsymbol{r}^{rl}, \boldsymbol{r}^{wd}, \boldsymbol{r}^{ch}\}$. The similarity function used here is the same as the one for entity linking, i.e.,

$$SR_j^{(x)} = \frac{\exp(\cos(\boldsymbol{q}_r^{(x)}, \boldsymbol{r}_j^{(x)}))}{\sum_{k=1}^{N} \exp(\cos(\boldsymbol{q}_r^{(x)}, \boldsymbol{r}_k^{(x)}))}, \tag{14}$$

where j is the index of relation candidates and N is the total number of relation candidates. Then, the similarity scores at three levels are averaged to generate the final similarity score for each relation candidate as

$$SR_j = \frac{\sum\limits_x SR_j^{(x)}}{3}. \tag{15}$$

The relation candidate with the highest similarity score is selected and combined with the topic entity to determine a triple in KB, whose object is the answer of the input question.

IV. TRAINING AND INFERENCE

A. TRAINING

The entity linking module and the relation detection module are jointly trained in an end-to-end manner. The training criterion is to minimize the negative log likelihood (NLL) of training samples. For each training sample $\{q, s_g, r_g\}$, where q is the question, s_g and r_g are the indexes of the golden subject and the golden relation labeled for the question, and its NLL is calculated as

$$L_{\{q,s_g,r_g\}} = -log(SE_{s_g} \cdot SR_{r_g}), \tag{16}$$

where SE_{s_g} and SR_{r_g} are calculated using Eqs. (7) and (15).

B. INFERENCE

At the testing stage, for each question q, we first generate a candidate set $\{s\}$ for entity linking. Then, a candidate set $\{r\}$ for relation detection is constructed by merging the relations attached to each subject candidate in the KB. The details of candidate generation are introduced in the next section. The indexes of the optimal subject-relation pair are predicted as

$$(i^*, j^*) = argmax_{i,j}(SE_i \cdot SR_j). \tag{17}$$

In our implementation, Eq. (17) is solved by greedy search, i.e., determining the optimal subject first and then discarding other subject candidates when building $\{r\}$. We also tried the beam search strategy by keeping more than one subject for relation detection. However, preliminary experimental results showed that there was no significant benefit of beam search.

V. EXPERIMENTS

A. DATASET AND EXPERIMENTAL SETTINGS

1) DATASET

The SimpleQuestions dataset [7] was adopted for evaluation in this paper. The original SimpleQuestions dataset consisted of 108,442 single-relation questions and their corresponding triples $(subject, relation, answer)$ in Freebase. Conventionally, this dataset was split into a training set of 75,910 question-triple pairs, a validation set of 10,845 pairs, and a test set of 21,687 pairs. Two subsets of Freebase, Freebase2M (FB2M) and Freebase5M (FB5M), were used for answer extraction. These two KBs contained approximately 2 million and 5 million entities, respectively.

2) CANDIDATE GENERATION

The original subjects and answers in the SimpleQuestions dataset were in the format of machine IDs (MIDs),[3] which were not suitable for natural language processing. Thus, we converted all MIDs into natural language forms for candidate generation. The main steps of generating candidate sets are described as follows.[4]

[3] MID is one of the IRI forms mentioned in Section II.A.

[4] There were no publicly available candidate sets for SimpleQuestions released by previous studies. Our candidate sets are available at https://drive.google.com/open?id=1i9ARCcvVX3PhdrR8lAnMk_uHFrS_a6DE.

TABLE 1. Some statistics on the datasets used for experiments and the results of candidate generation.

	FB2M			FB5M		
	Training	Validation	Test	Training	Validation	Test
Original number of questions	75,910	10,845	21,687	75,910	10,845	21,687
Number of questions used for experiments	75,831	10,822	21,687	75,821	10,822	21,687
Number of questions with missing subjects	0	520	1,226	0	521	1,229
Number of questions with missing relations	0	0	0	0	0	0
Average number of subject candidates	3.09	3.06	3.05	3.15	3.11	3.11
Average number of relation candidates	16.43	16.4	16.83	18.86	18.81	19.3

TABLE 2. The test set accuracies (%) of different methods.

Methods	FB2M	FB5M
MemNN-Ensemble [7]	62.9	63.9
Character Attention [8]	70.9	70.3
GRU [20]	71.2	-
BiLSTM+BiGRU [15]	74.9	-
STAGG [13]	76.4	-
AMPCNN [19]	76.4	75.9
Multi-Detectors [12]	78.7	-
our method	**82.1**	**82.29**

TABLE 3. The test set accuracies (%) of using different pooling strategies to build the question encoder for entity linking.

	Joint Acc.	Entity Acc.	Relation Acc.
Mean Pooling	81.85	92.21	82.00
Inner-Sentence Attention	82.19	92.12	82.35
Structure Attention (K=1)	**82.29**	92.15	**82.44**
Structure Attention (K=3)	81.82	**92.23**	81.96
Structure Attention (K=5)	81.99	92.12	82.14

- A MID-to-name mapping table was first built for all the subjects that appeared in FB2M or FB5M using the relation ‘‘/type/object/name’’.
- For a question in the SimpleQuestions dataset, each name in the MID-to-name mapping table that was a substring of the question was considered as its subject candidate. To reduce redundancy, the name that was a substring of another name in the candidate set was removed.
- For each subject candidate, all the relations directly linking with it in FB2M or FB5M were treated as relation candidates.

Some statistics on the datasets used for the experiments and the results of candidate generation are shown in Table 1. The number of training and validation samples were less than the original ones for two reasons. First, there were some questions that had missing golden subjects in the MID-to-name mapping table. Second, there were questions whose golden subjects or golden relations were missing in the candidate sets. Although the test set also contained such questions, we retained them and labeled them as incorrect ones during the test for fair comparisons with other studies.

In Table 1, the questions with missing subjects means those questions whose golden subjects cannot be found in the subject candidates of these questions. The questions with missing relations means those questions whose golden subjects are in candidate sets but whose golden relations cannot be found in relation candidates. The average number of subjects means the average number of subject candidates for all questions in each set. The average number of relations means the average number of relation candidates linking with the golden subjects for all questions in each set. Since the scale of FB5M was larger than that of FB2M, it is reasonable that FB5M yielded more subject candidates and relation candidates than FB2M.

3) EXPERIMENTAL SETTINGS

The hidden units of both forward and backward LSTMs in all BiLSTMs were 100, and the output dimensions of all BiLSTMs were 200. For a multilevel relation encoder, we counted the frequencies of all relations, words, and characters in the training set and chose the top 8,000 relations, 5,000 words, and 150 characters to form the dictionaries of embedding matrices. The dimensions of the embedding matrices were 200. All the hyperparameters of the built network were determined according to the performance of some preliminary experiments on the validation set. All model parameters were initialized by sampling from a standard Gaussian distribution. Adadelta was adopted for optimization, and minibatches were utilized. The max epoch was set as 50, and an early stop strategy was adopted with a patience number of 3. The model training cost two days on a GTX1080 GPU.[5]

B. RESULTS

1) THE EFFECTIVENESS OF THE PROPOSED METHOD

To evaluate the effectiveness of our proposed method, we compared our method with previous studies using the SimpleQuestions dataset, and the question answering accuracies on the test set are shown in Table 2. In this table, the MemNN-Ensemble model [7] provided the first results on the SimpleQuestions dataset, which used a memory network to remember Freebase facts and simply calculated the similarity scores between questions and Freebase facts.

[5] Our code and data are available at https://github.com/runzewang/target-att.

TABLE 4. The test set accuracies (%) of ablation analysis on the question encoder for relation detection.

Pooling	Settings	Joint Acc.	Entity Acc.	Relation Acc.
MLTA	Relation+Word+Character	**82.29**	92.15	**82.44**
	Relation+Word	81.92	**92.16**	82.04
	Relation+Character	82.07	92.12	82.23
	Word+Character	81.85	92.09	82.01
	Relation	80.52	91.92	80.67
	Word	81.68	92.09	81.79
	Character	81.83	92.14	81.99
Others	Mean Pooling	81.74	91.98	81.94
	Inner-Sentence Attention	81.4	91.94	81.62
	Structure Attention (K=1)	81.29	92.04	81.47
	Structure Attention (K=3)	81.61	92.02	81.75
	Structure Attention (K=5)	81.74	92.05	81.88

The character attention method [8] adopted a character-level encoder-decoder model and utilized the attention mechanism at the decoder layer. The GRU method [20] encoded each word in the input question with a GRU at both the character level and word level and then fed them into another GRU for generating the final question representations. The BiLSTM+BiGRU method [15] built a BiLSTM as the subject encoder and a BiGRU as the relation encoder. The STAGG method [13] followed the semantic parsing approach and converted each question into a query graph. The AMPCNN method [19] used an attentive convolutional neural network that implanted the attention mechanism into the CNN mechanism for question encoding. The multidetector method [12] only focused on relation detection. It first encoded questions at two levels, i.e., the relation level and word level, and then simply calculated the similarity scores between each question and its relation candidates. Our proposed method adopted structure attention with $K = 1$ in the question encoder for entity linking and MLTA in the question encoder for relation detection.

From Table 2, we can see that our method outperformed all previous methods and improved the state-of-the art test set accuracy of the SimpleQuestions dataset from 78.7% to 82.29%. Since our method achieved slightly better performance on FB5M than on FB2M, the following analytical experiments were conducted only on FB5M.

2) COMPARISON OF QUESTION ENCODERS FOR ENTITY LINKING

The performances of using different pooling strategies to build the question encoder for entity linking were compared with experiments. The results are shown in Table 3. In this table, entity accuracy and relation accuracy mean the percentages of test samples whose topic entities and relations were correctly predicted, respectively. In addition, joint accuracy means the percentage that both of them were correctly predicted, i.e., the accuracy shown in Table 2.

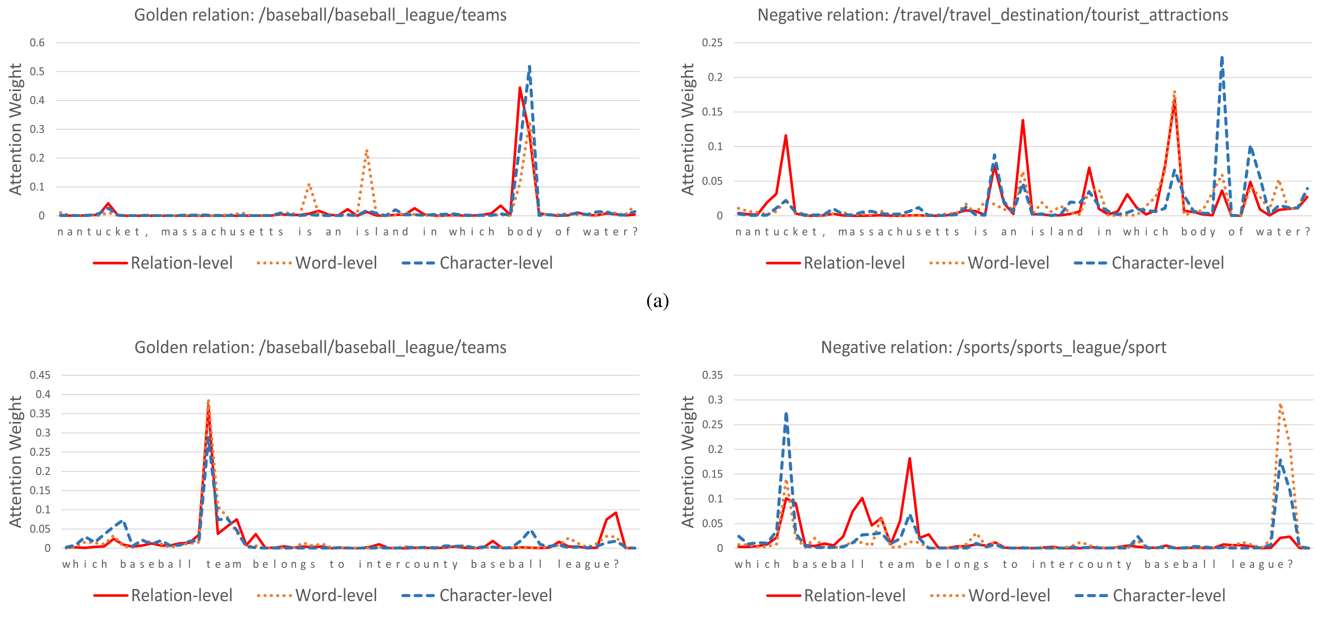
From this table, we can see that the structure attention with $K = 3$ achieved the best performance on entity linking, while the structure attention with $K = 1$ achieved the highest joint accuracy. One reason for this inconsistency was that the question embedding matrix and the two-layer BiLSTMs were shared by both of the question encoders for entity linking and relation detection, and all model parameters were estimated jointly in an end-to-end manner. Thus, changing the pooling strategy in the question encoder for entity linking may also influence the accuracy of relation detection. Finally, we adopted the structure attention with $K = 1$ in the question encoder for entity linking because it achieved the best joint accuracy among all mean-pooling and attentive-pooling strategies.

3) ABLATION ANALYSIS ON RELATION DETECTION

To evaluate the effectiveness of the MLTA method proposed in this paper, ablation analysis was conducted by removing different levels in MLTA and replacing MLTA with other pooling strategies in the question encoder for relation detection. The results are shown in Table 4.

From this table, we can see that when using single-level representation in MLTA, the character level and word level achieved a similar performance, while the relation level achieved worse performance. When using two levels, combining the relation-level and character-level representations achieved the best performance. We can also find that using word-level and character-level representations jointly did not improve the performance of using a single one of them. Adding relation-level representation to character-level or word-level representation achieved significant improvement. These results implied that word-level and character-level representations may contain similar information for relation detection, while the relation-level representation can provide some additional information to them. The relation accuracy and the joint accuracy achieved the best performance when using all of the three levels jointly in MLTA.

We sampled one question in the test set that was answered correctly when using the MLTA model with all three levels but was answered incorrectly when using MLTA with only one or two levels. The results are shown in Table 5. For the question *what do they speak on dance your ass off?*, the topic



(a)

(b)

FIGURE 4. Attention weights on the character sequences of two test questions calculated by MLTA for relation detection. For each question, the results of the golden relation and a negative relation are shown. (a) Nantucket, Massachusetts, is an island in which body of water?. (b) Which baseball team belongs to an intercounty baseball league?.

TABLE 5. The predicted relations for the question "what do they speak on dance your ass off?" using MLTA models with different levels. For this question, all models predicted the topic entity *dance your ass off* correctly. The golden relation is */tv/tv_program/languages*, which is predicted correctly only when the MLTA model considers all three levels.

Settings	Predicted Relation
Relation+Word+Character	/tv/tv_program/languages
Relation+Word	/tv/tv_program/genre
Relation+Character	/music/release/format
Word+Character	/tv/tv_program/genre
Relation	/music/release/track_list
Word	/music/release/region
Character	/music/release/region

entity *dance your ass off* was linked with many relations in the KB. Some relations were about television programs, while some were music-related. The most important phrase in the question for relation detection was *speak on*, which supported the golden relation *tv/tv_program/languages* in an implicit way. When using only single-level representation in MLTA, all models incorrectly predicted the relation as a music-related one, as shown in the last three rows of Table 5. When using relation-level and word-level representations or word-level and character-level representations in MLTA, the models can predict the relation as a TV program-related one but still failed to link the phrase *speak on* to the concept of *language*. For the MLTA model considering all relations, word and character levels, the golden relation was predicted correctly, which demonstrated the advantages of utilizing comprehensive representations in MLTA.

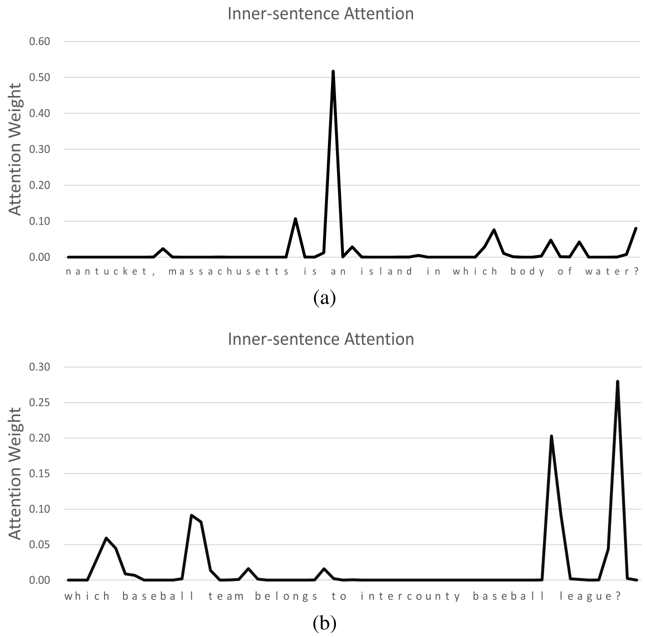
Table 4 also compares MLTA with other pooling strategies. We can see that MLTA with three levels achieved better performance than other pooling strategies. As we have discussed, the advantage of our proposed target attention strategy is that it can take relation candidates into consideration when encoding the input question. Thus, the relation-dependent question embedding vectors can be beneficial to the matching between questions and relation candidates for relation detection.

We also exported the attention weights calculated by MLTA on the character sequences of two questions in the test set. For each question, its golden relation and a randomly sampled negative relation were used. The results are shown in Fig. 4. For question (a), the left picture shows that the relation-level and character-level attentions focused on *body of water*, while the word-level attention also assigned high weights on *island*. The right picture was produced by using a negative relation as a query for attentive pooling. Since there were no correlative descriptions between this relation and the question, the attention weights distributed randomly without focuses. For question (b), the left picture also showed reasonable attention weights. Different from question (a), the negative relation of question (b) shared a word, *league*, with the question, and the word *sport* in the relation name also related to the word *baseball* in the question. Thus, MLTA assigned higher word-level weights to *league* and character-level weights to *baseball*. However, the consistency among the three-level attention weights calculated using the negative relation was much worse than the one using the golden relation. This result helped to discriminate them during relation detection.

For comparison, the attention weights on the characters of these two questions calculated by inner-sentence

TABLE 6. Some sampled questions for error analysis.

Error Type	Index	Question	Golden	Predicted
Subject	1	which instrument does amapola cabase play?	maria cabase	play
	2	what is the release type of justin?	a tian	justin
	3	what is the name of buddy holly's album?	album	buddy holly
	4	is popeye a human or an alien?	popeye	human
Relation	5	who is the author of warrior queen?	/book/written_work/author	/music/recording/artist
	6	what is theta andromedae?	/astronomy/celestial_object/category	/common/topic/notable_types
	7	which tracks were a part of the release catalog 1/14 - 5/12?	/music/release/track_list	/music/release/track
	8	what province is gabon in?	/location/location/contains	/location/location/contained by



(a)

(b)

FIGURE 5. Attention weights on the character sequences of two test questions calculated by inner-sentence attention for relation detection. Right subjects and wrong relations were predicted for both questions when using inner-sentence attention for relation detection. (a) Nantucket, Massachusetts, is an island in which body of water?. (b) Which baseball team belongs to an intercounty baseball league?.

attention for relation detection were also exported, and the results are shown in Fig. 5. Correct topic entities and wrong relations were predicted for both questions when using inner-sentence attention for relation detection. We can see that for question (a), inner-sentence attention focused on the word *island*, while it ignored an important segment *body of water* for relation detection. For question (b), inner-sentence attention assigned high weights to the word *league*, while it ignored the word *team*. The main reason was that the inner-sentence attention strategy only utilized the question itself for calculating attention weights. Thus, it neglected the key information carried by relation candidates. In contrast, the MLTA method proposed in this paper utilized the multilevel embedding vectors of relation candidates as queries, and thus, the computed relation-dependent question representations are more informative for relation detection.

4) ERROR ANALYSIS

We evaluated some statistics on the evaluation results of our proposed method on FB5M. There were 44 test questions whose golden subject could not find a corresponding natural language name in the MID-to-name mapping table. These questions were removed before candidate generation but were still labeled as incorrect ones when calculating the test set accuracy. There were 3,779 test questions that were answered incorrectly with our method, and 43.42% of them were caused by a wrong entity-linking result, while the rest of them were answered with a correct subject but a wrong relation. We randomly sampled 20 test questions with incorrect subjects and 20 questions with incorrect relations to analyze the reasons that caused there errors.

The reasons for entity linking errors are summarized as follows, and some examples are shown in Table 6.

- There were 14 questions whose golden subjects could not be found in the candidate sets. One reason was that the golden subject was a synonym for a substring in the question, e.g., Question #1 in Table 6. As introduced in Section V.A.2, we generated subject candidates only by extracting the entities that were substrings of input questions. Another reason was that the MID-to-name table generated a wrong name for the MID of the golden subject, e.g., Question #2.
- There were 5 errors caused by the incorrect subject labels in the original SimpleQuestions dataset. However, our method made correct predictions, e.g., Question #3.
- The last question, i.e., Question #4, was incorrectly answered because the entity linking procedure made a wrong prediction among subject candidates.

The reasons for relation detection errors are summarized as follows, and some examples are also shown in Table 6.

- Our model cannot distinguish if an entity is a music recording or a book. For Question #5, *warrior queen* was treated as a music recording incorrectly. There were 4 such errors in total in the 20 samples.
- The question was too simple to provide sufficient information for relation detection, e.g., Question #6. There were 10 such errors in total in the 20 samples.
- Freebase may express the relationship between a subject and an object with different relations. For Question #7, although both the golden relation and the predicted

relation can lead to the same answer, we still labeled it as a mistake. There was 1 such error in the 20 samples.

- Our method failed to distinguish subtle meaning differences among some relations with similar textual representation, e.g., the meanings of *contains* and *contained by* in Question #8. There were 5 such errors in the 20 samples.

VI. CONCLUSION

In this paper, we have presented an end-to-end neural network model for single-relation question answering and evaluated it on the SimpleQuestions dataset. We compared different attentive pooling methods for entity linking and finally chose the single-head structure attention strategy. For relation detection, a multilevel target attention method was proposed to utilize the multilevel descriptions of relations for deriving relation-dependent question representations. The entity linking and the relation detection modules were jointly trained using training samples. Finally, we achieved the state-of-the-art accuracy of 82.29% on the SimpleQuestions dataset. To integrate external knowledge, such as synonyms, into our model and to further improve the discrimination ability of the relation detection module will be our future work.

REFERENCES

[1] J. Weston *et al.* (2015). "Towards AI-complete question answering: A set of prerequisite toy tasks." [Online]. Available: https://arxiv.org/abs/1502.05698

[2] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data," in *The Semantic Web*. Berlin, Germany: Springer, 2007, pp. 722–735.

[5] A. Fader, L. Zettlemoyer, and O. Etzioni, "Paraphrase-driven learning for open question answering," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1608–1618.

[6] W.-T. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2014, pp. 643–648.

[7] A. Bordes, N. Usunier, S. Chopra, and J. Weston. (2015). "Large-scale simple question answering with memory networks." [Online]. Available: https://arxiv.org/abs/1506.02075

[8] X. He and D. Golub, "Character-level question answering with attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1598–1607.

[9] Y. Hao *et al.*, "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 221–231.

[10] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 615–620.

[11] A. Bordes, J. Weston, and N. Usunier, "Open question answering with weakly supervised embedding models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2014, pp. 165–180.

[12] M. Yu, W. Yin, K. S. Hasan, C. dos Santos, B. Xiang, and B. Zhou, "Improved neural relation detection for knowledge base question answering," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 571–581.

[13] S. W.-T. Yih, M.-W. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Language Process.*, vol. 1, 2015, pp. 1321–1331.

[14] X. Yao and B. Van Durme, "Information extraction over structured data: Question answering with freebase," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2014, pp. 956–966.

[15] S. Mohammed, P. Shi, and J. Lin, "Strong baselines for simple question answering over knowledge graphs with and without neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 291–296.

[16] R.-Z. Wang, C.-D. Zhan, and Z.-H. Ling, "Question answering with character-level LSTM encoders and model-based data augmentation," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Cham, Switzerland: Springer, 2017, pp. 295–305.

[17] Y. Liu, C. Sun, L. Lin, and X. Wang. (2016). "Learning natural language inference using bidirectional LSTM model and inner-attention." [Online]. Available: https://arxiv.org/abs/1605.09090

[18] Z. Lin *et al.* (2017). "A structured self-attentive sentence embedding." [Online]. Available: https://arxiv.org/abs/1703.03130

[19] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schütze, "Simple question answering by attentive convolutional neural network," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 1746–1756.

[20] D. Lukovnikov, A. Fischer, J. Lehmann, and S. Auer, "Neural network-based question answering over knowledge graphs on word and character level," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1211–1220.

[21] C. Zhang, Y. Zhang, W. Xu, Z. Ma, Y. Leng, and J. Guo, "Mining activation force defined dependency patterns for relation extraction," *Knowl.-Based Syst.*, vol. 86, pp. 278–287, Sep. 2015.

[22] C. Zhang, W. Xu, Z. Ma, S. Gao, Q. Li, and J. Guo, "Construction of semantic bootstrapping models for relation extraction," *Knowl.-Based Syst.*, vol. 83, pp. 128–137, Jul. 2015.

[23] Q. Cai and A. Yates, "Large-scale semantic parsing via schema matching and lexicon extension," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 423–433.

[24] L. S. Zettlemoyer and M. Collins, "Learning context-dependent mappings from sentences to logical form," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (AFNLP)*, vol. 2, 2009, pp. 976–984.

[25] T. Kwiatkowski, E. Choi, Y. Artzi, and L. Zettlemoyer, "Scaling semantic parsers with on-the-fly ontology matching," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1545–1556.

[26] L. S. Zettlemoyer and M. Collins, "Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars," in *Proc. 21st Conf. Uncertainty Artif. Intell.*, 2005, pp. 658–666.

[27] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2013, pp. 1533–1544.

[28] Y. Feng, S. Huang, and D. Zhao, "Hybrid question answering over knowledge base and free text," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 2397–2407.

[29] K. Xu, S. Reddy, Y. Feng, S. Huang, and D. Zhao, "Question answering on freebase via relation extraction and textual evidence," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 2326–2336.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Workshop Deep Learn.*, Dec. 2014.

[32] H. Zhang, G. Xu, X. Liang, T. Huang, and K. Fu. (2018). "An attention-based word-level interaction model: Relation detection for knowledge base question answering." [Online]. Available: https://arxiv.org/abs/1801.09893

[33] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 551–561.

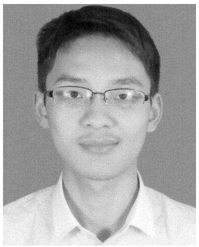
[34] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[35] T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, and C. Zhang. (2018). "Reinforced self-attention network: A hybrid of hard and soft attention for sequence modeling." [Online]. Available: https://arxiv.org/abs/1801.10296

[36] Q. Chen, Z.-H. Ling, and X. Zhu, "Enhancing sentence embedding with generalized pooling," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1815–1826.

[37] H. Sun, J. Liao, J. Wang, and Q. Qi, "MTDE: Multi-typed data embedding in heterogeneous networks," *Neurocomputing*, vol. 278, pp. 119–125, Feb. 2018.

[38] D. Bahdanau, K. Cho, and Y. Bengio. (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473



RUN-ZE WANG received the B.E. degree in electronic information engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2016, where he is currently pursuing the Ph.D. degree in signal and information processing. His research interests include natural language processing and deep learning.

ZHEN-HUA LING (M'10–SM'19) received the B.E. degree in electronic information engineering, the M.S. and Ph.D. degrees in signal and information processing from the University of Science and Technology of China, Hefei, China, in 2002, 2005, and 2008, respectively. From 2007 to 2008, he was a Marie Curie Fellow with the Centre for Speech Technology Research (CSTR), University of Edinburgh, Edinburgh, U.K. From 2008 to 2011, he was a joint Postdoctoral Researcher with the University of Science and Technology of China and iFLYTEK Co., Ltd., China. He also worked with the University of Washington, Seattle, WA, USA, as a Visiting Scholar, from 2012 to 2013. He is currently an Associate Professor with the University of Science and Technology of China. His research interests include speech processing, speech synthesis, voice conversion, and natural language processing. He was a recipient of the IEEE Signal Processing Society Young Author Best Paper Award, in 2010. He is currently an Associate Editor of the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

YU HU received the B.E., M.E., and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2000, 2003, and 2009, respectively, all in electrical engineering. In 1999, he became a Research Engineer with iFlytek, Ltd., as a Cofounder, working on Mandarin speech synthesis and speech prosody analysis. He was one of the researchers who built the first few generations of iFlytek Mandarin speech synthesis engines. Since 2004, his research interest has changed to robust speech recognition, and began to work on the iFlytek Mandarin speech recognition systems. He is currently the Director of iFlytek Research and working on speech recognition over mobile internet.