A BERT-based Approach with Relation-aware Attention for Knowledge Base Question Answering

1st Da Luo
School of Computer Science and Engineering
South China University of Technology
Guangzhou, China
csluoda@mail.scut.edu.cn

2nd Jindian Su*
School of Computer Science and Engineering
South China University of Technology
Guangzhou, China
sujd@scut.edu.cn

3rd Shanshan Yu*
School of Medical Information Engineering
Guangdong Pharmaceutical University
Guangzhou, China
susyu@139.com

***Abstract*—Knowledge Base Question Answering (KBQA), which uses the facts in the knowledge base (KB) to answer natural language questions, has received extensive attention in recent years. The existing works mainly focus on the modeling method and neglect the relations between questions and KB facts, which might restrict the further improvements of the performance. To address this problem, this paper proposes a BERT-based approach for single-relation question answering (SR-QA), which consists of two models, entity linking and relation detection. For entity linking, we adopt pre-trained BERT and a heuristic algorithm to reduce the noise in the candidate facts. For relation detection, the existing approaches usually model the question and the candidate fact respectively before calculate their semantic similarity, which might lose part of the original interaction information between them. To work around this problem, a BERT-based model with relation-aware attention is proposed. We construct the question-fact pair as the input of pre-trained BERT to preserves the original interactive information. To bridge the semantic gap between the questions and the KB facts, we also use a relation-aware attention network to enhance the representation of candidates. The experimental results show that our entity linking model achieves new state-of-the-art results and our complete approach also achieves state-of-the-art accuracy of 80.9% on the SimpleQuestions dataset.**

***Index Terms*—Deep learning, knowledge base question answering, BERT, attention mechanism**

I. Introduction

In recent years, with the emergence of large-scale knowledge bases (KB) such as Freebase [1], YAGO [2] and DBpedia [3], Knowledge Based Question Answering(KBQA) has attracted a lot of attention [4]–[6] and become a heated research hotspot. KBQA aims to automatically answer the natural language question with the facts from a KB. The facts in KB are usually organized as triples, i.e., (head entity, relation, tail entity). According to the number of KB triples in the answer to the question, the KBQA task can be devided into two subtasks: single-relation question answering (SR-QA) and muti-relation question answering (MR-QA). This paper focuses on SR-QA task, where the questions can be answered by a single fact from the KB. Most of the questions queried on the web are single-relation questions, and the SR-QA task is still far from being solved because of the semantic gap between the natural language questions and the structured facts in KB.

The current researches on KBQA can be divided into two categories, semantic parsing-based method [7]–[10] and neural network-based method [11]–[14]. The former uses semantic parsing to translate a natural language question into a KB-specific structured form, which can be used to query the answer from the KB directly. However, such methods might lack of flexibility and versatility. The latter tries to encode both questions and the candidate KB facts as vectors, such that the question and the correct KB facts are the nearest neighbors in the vector space, and finally gets the answers that match the questions best. In general, the KB-based SR-QA usually consists of two steps: (1) entity linking. It finds out the head entities related to the question and then gets the candidate facts from the KB. (2) relation detection. It predicts the candidate triple that can best answer the question.

For entity linking, most of the existing methods first use a sequence labeling model to find out the words in the question that are related to the KB. Then, they query the KB head entities whose names contain these words and get a set of candidate KB facts linked to the entities [12], [13], [15]. Although existing neural network-based methods have achieved good results, the complex expression of the question still brings a great challenge. For example, the word "Apple" in the questions may refer to "Apple Corporation" or the fruit "apple." Since the pre-trained language models [16], [17] proved to have impressive results in the sequence labeling task, we employ the pre-trained BERT [18] with a heuristic algorithm to improve the performance of entity linking.

For relation detection, many researchers map the questions and the candidate facts to low-dimensional vectors. Then, they encode the interaction information between the questions and the facts and afterwards calculate their similarity to obtain

This work was supported in part by the National Natural Science Foundation of China under Grant 61936003, in part by Research and Development Program in Key Areas of Guangdong Province under Grant 2018B010109004, in part by the Applied Scientific and Technology Special Project of Department of Science and Technology of Guangdong Province under Grant 20168010124010, in part by Natural Science Foundation of Guangdong Province under Grant 2015A030310318, in part by Medical Scientific Research Foundation of Guangdong Province under Grant A2015065, and in part by Innovation Project of Guangdong Pharmaceutical University under Grant 52159433.

Corresponding author: Jindian Su, Shanshan Yu (e-mail: sujd@scut.edu.cn, susyu@139.com)

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the answers [19]–[21]. However, all these methods treat the modeling of the questions and the candidate facts as two separate processes, i.e. using the LSTM [22] or CNN [23] to encode them respectively, which may easily lose the original interaction information as a result. To tackle this issue, we construct the question-fact pair as the input of the BERT-based model and employ the Bi-diretional LSTM (Bi-LSTM) to extract the high-level interaction information. Another challenge of relation detection is the semantic gap between the natural language questions and the structured KB facts. Some studies use the type of entities as additional information to enhance the representation of the candidate facts [12], [24], but the type attribute of an entity in the KB is often missed. Since the intended answer of SR-QA is the tail entity of the correct triple, we enhance the representation of candidates by the relations linked to their tail entities. And we use the attention machanism to increase the weight of relations which is more relevant to the question.

This paper proposes a novel approach for KB-based SR-QA, which consists of two models: entity linking model and relation detection model. The constriubtions of this paper are summarized as follows:

1) For entity linking, we propose a BERT-based sequence labeling model to improve the recognition performance of entity words mentioned in the question. And we also present a heuristic algorithm to reduce the impact of labeling errors.
2) For relation detection, to preserve the original interaction information between the question and the candidate, we use a pre-trained BERT model for encoding the question-fact pair and extracting high-level matching features via Bi-LSTM. We also make use of a relation-aware attention network to enhance the representation of candidates.
3) The experimental results on the SimpleQuestions dataset show that our overall method achieves state-of-the-art accuracy of 80.9%.

The rest of this paper is organized as follows: Section II briefly summarizes some typical related studies about KBQA and BERT. Section III describe our proposed approach in detail. Section IV presents the experiments and their results. The error analysis is shown in section V. Finally, the conclusions and directions for future work are given in section VI.

II. Related Work

A. Neural Network-based KBQA

The goal of Neural Network-based KBQA is to measure the semantic similarity between the question and the candidates. The core of KBQA consists of two parts: entity linking, which determines the quality of the candidates, and relation detection, which determines the accuracy of the answer selection.

Early entity linking studies often use n-gram of the question words to search for the KB entities whose name contains the text and generate a set of candidate KB facts [11], [20], [25]. However, this approach requires lots of KB queries which is time-consuming and introduces too many noisy candidates. More recently, many works use a sequence labeling model to narrow down the n-gram search range, Dai *et al.* [12] and Yin *et al.* [15] employ a BiLSTM-CRF sequence labeling model to identify the entity mention of the question, and then query the candidate entity in KB using the entity mention. Qu *et al.* [13] not only adopt a sequence labeling model to identify the entity mention, but also propose a heuristic algorithm to reduce the impact of the error from the sequence labeling model. As we can see, the performance of the sequence labeling model determines the quality of the entity linking results. To better encode the words in the question according to their context, we employ the pre-trained BERT with a heuristic algorithm to improve the performance of entity linking.

Relation detection methods for KBQA are originally based on the hand-craft features [26], [27]. With the development of deep learning, many researchers apply deep learning architectures to relation detection task. Golub *et al.* [20] propose a character-level encoder-decoder framework, and adopt the attention mechanism to better handle longer sequences. Yu *et al.* [19] use both word-level and relation-level relation representations, and propose the Hierarchical Residual Bi-LSTM model for relation detection. To utilizing the mutilevel feature of relations, Wang *et al.* [14] propose a multilevel target attention method and achieve great success. Qu *et al.* [13] present the AR-SMCNN model, which is able to capture comprehensive hierarchical information utilizing the advantages of both RNN and CNN. Lan *et al.* [28] make use of a matching-aggregation model and question-specific contextual relations for relation detection. The main difference between the models mentioned above and ours is that we construct the question-fact pair as the input of our model to preserve the original interaction information between the question and the candidate. We also use the attention mechanism to make better use of the relations linked to the candidates, which enhances the representation of candidate answers.

B. BERT

BERT is a language representation model which stands for Bidirectional Encoder Representations from Transformers [18]. The proposal of BERT has achieved great results on many natural language processing tasks, such as named entity recognition(NER), question answering(QA), and machine reading comprehension(MRC). Goldberg *et al.* [29] demonstrates that the BERT model extracts the syntactic information of subject-verb agreement well. The research from Jawahar *et al.* [30] shows that BERT extracts an extensive hierarchical linguistic information, with surface features in lower layers, syntactic features in middle layers and semantic features in higher layers. This work investigates the use of pre-trained BERT for the single-relation question answering(SR-QA) task. In the entity linking model, we employ BERT to improve the accuracy of the sequence labeling model. In the relation detection model, we take advantage of the transformers of BERT to extract the original interaction information between the question and the answer.

III. APPROACH

In this section, we will introduce our approach to SR-QA. As illustrated in Fig.1, our approach contains two models:

1) **Entity linking**: It recognizes the entity words mentioned in the question and then gets the candidate facts and the relations linked to candidates by querying to the KB.
2) **Relation detection**: It gets the best match facts by calculating the similarity between all candidate facts and the question.

A. Entity linking

The main task of the entity linking model is to identify the entity mention, which are several consecutive words in the question. Therefore, we can regard the task as a binary classification problem that predicts whether each word in the question belongs to the entity mention. Given a question Q, we feed it into the WordPiece [31] tokenizer and fine-tune BERT as the NER method mentioned in [18]. Then we get the words that are labeled as positive by the model. Since these words may not be consecutive, we adopt a heuristic algorithm as follows:

1) If a word is predicted as negative, but its neighbors(both on left and right) are positive, it is very likely to be wrong. So we change the labels of all these words to positive and then combine the adjacent positive words to get the substring S. If there is more than one S, we keep the last one, which is based on the observation that the entity mention is usually close to the end of the question.
2) Query out all the entities in Freebase whose name or alias is exactly the same as S and form a set of candidate entities E. If none of the entities in Freebase match S, then continue to the next step.
3) Combine all the words in S as a window and slide the window with the distance of [+1, -1, +2, -2]('+' means move to right and '-' means move to left). Each time we slide, we will get a new substring S' and then use S' to find the corresponding entities as step 2). Once a match is found, the sliding stops, and we get the entity mention S' and the candidate entities E

Algorithm 1 describes the heuristic algorithm that we adopt in detail.

For each candidate entity e in E, we can get the candidate answers as (e, r, t) by query out all the relations linked to e from KB. For each candidate answer, we also query out the relations R linked to t as the additional information of the answer. Finally, we exclude the tail entity t in the original candidate answer as other studies do, and get a set of candidate answers F for each question.

B. Relation detection

Given a question Q, The target of relation detection is to calculate a matching score $S(Q, f_i)$ for each fact f_i in the candidate answers set F. The final predition $\tilde{f}_i$ is as follow:

$$\tilde{f}_i = \arg\max_{f_i \in F} S(Q, f_i) \tag{1}$$

The architecture of our relation detection model is shown in Fig.2. First, we take the question-fact pair as the input of BERT, and we take the relations linked to the candidate as the input of the relation embedding matrix. Next, a pre-trained BERT is employed to encode the question-fact pair. Then we use Bi-LSTM and relation-aware attention network to calculation the matching scores z_1 and z_2. Finally, we combine the two scores and calculate the final score between the question and each corresponding candidate answer. The details are shown in the following sections.

1) INPUT LAYER: Given a question Q and a candidate fact f, we construct the question-fact pair as the input of the BERT-based encoder. Firstly we concatenate them into a sequence with a separator token [SEP]. Then we add a classification token [CLS] before the sequence and a token [SEP] as the end. Finally the WordPiece tokenization is perform and we get the input sequence as $T = [t_1, t_2, ..., t_k]$, where k is the number of tokens in the input sequence.

For each candidate fact f, since the intended answer is its tail entity, we also take the relations connected to the tail entity of f as its additional information. Then the relations R are mapped as $R' = [r_1, r_2, ..., r_l]$ by a relation embedding matrix $E_r \in \mathbb{R}^{v_r \times d_r}$, which is randomly initialized. Here l is the number of relations linked to the candidate fact, d_r means the dimension of the relation embedding and v_r is the size of the relation vocabulary.

2) BERT-BASED ENCODER: The purpose of the BERT-based encoder is to extract the primary interactive information between the question and the candidate fact. BERT consists of multi-layer bidirectional Transformers, which relies entirely on an attention mechanism to extract global dependencies from all the inputs. The benefit of this construction is that the BERT-based encoder can preserve the original interactive information from the question-fact pair and remove the long-term memory bottleneck faced by RNN based models.

The input of BERT-based encoder is the tokenized question-fact pair $T = [t_1, t_2, ..., t_k]$ from input layer. Unlike the sentence pair classification method in [18] that utilizes only the hidden state of the [CLS] token, we also use all of the representations to measure the matching information between the question and the candidate. Thus, the output of the encoding layer are a sequence of vectors:

$$X = [x_1, x_2, ..., x_k] \quad x_i \in \mathbb{R}^{d_{bert}}, \forall i \in \{1, 2..., k\} \tag{2}$$

where d_{bert} is the hidden size of the BERT model and k is the number of the input tokens.

3) SIMILARITY FROM Bi-LSTM: Given the contextual representation X from the BERT-based encoder, we employ Bi-LSTM to extract the high-level interaction information between the question and the candidate fact. Then the calculation of an LSTM unit is as follow:

$$f_t = \sigma(W_f[x_t; h_{t-1}] + b_f) \tag{3}$$

$$i_t = \sigma(W_i[x_t; h_{t-1}] + b_i) \tag{4}$$

$$o_t = \sigma(W_o[x_t; h_{t-1}] + b_o) \tag{5}$$

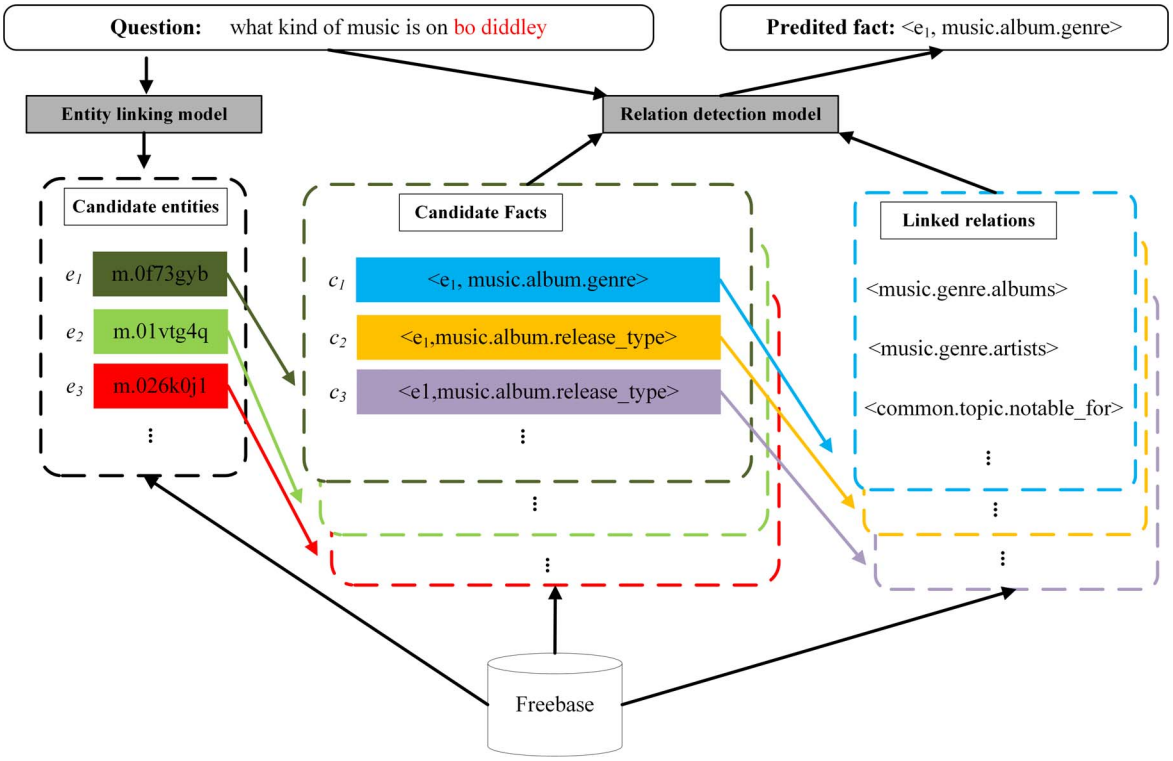

Fig. 1. The process of our KBQA approach

Algorithm 1 The heuristic algorithm in entity linking

Require: Question tokens: $T = [t_1, t_2, ..., t_k]$, labels of question tokens: $L = l_1, l_2, ..., l_k$
Ensure: Candidate entities E

```
S = [t_i, t_{i+1}, ..., t_n] ∈ T ← Combine the adjacent words whose labels are positive and keep the last substring
for x in [0, +1, −1, +2, −2] do
    S' = [t_{i+x}, t_{i+1+x}..., t_{n+x}]
    if there is no entity named S' in the KB then
        continue
    else
        E(S') ← Take all the entities named S' as candidate entites
        break
return Candidate entities E ← E(S')
```

$$\tilde{c}_t = \tanh(W_c[x_t; h_{t-1}] + b_c) \tag{6}$$

$$c_t = \sigma(f_t \odot c_{t-1} + i_t \odot \tilde{c}_t) \tag{7}$$

$$h_t = o_t \odot \tanh(c_t) \tag{8}$$

where $\sigma(\cdot)$ is the activation functionn and $\odot$ denotes element-wise production; x_t is the input vector of step t and h_{t-1} is the hidden state of last step; W_f, W_i, W_o, W_c, b_f, b_i, b_o and b_c are the parameters of the LSTM. For each word at step t, the output of Bi-LSTM is $v_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$, where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the hidden state from forward LSTM and backward LSTM. And $[\cdot;\cdot]$ is the concatenation operator.

Then we employ a linear layer to get the similarity score z_1 between the question and the candidate:

$$z_1 = W^{\mathrm{T}}[v_1; v_2...; v_k] + b \tag{9}$$

where V is the output of Bi-LSTM, and $[v_1; v_2...; v_k]$ is the concatenation of V.

4) SIMILARITY FROM RELATION-AWARE ATTENTION NETWORK: To enhance the representation of the candidate fact, we apply an attention network between the relevant relations $R' = [r_1, r_2, ..., r_l]$ from the input layer and the hidden state of [CLS] token x_1 from the BERT-based encoder, and get another similarity score z_2:

$$z_2 = x_1' \otimes e \tag{10}$$

The operation $\otimes$ here is the dot product of two vectors. And x_1' and e are calculated as:

$$x_1' = W_t^{\mathrm{T}} x_1 \tag{11}$$

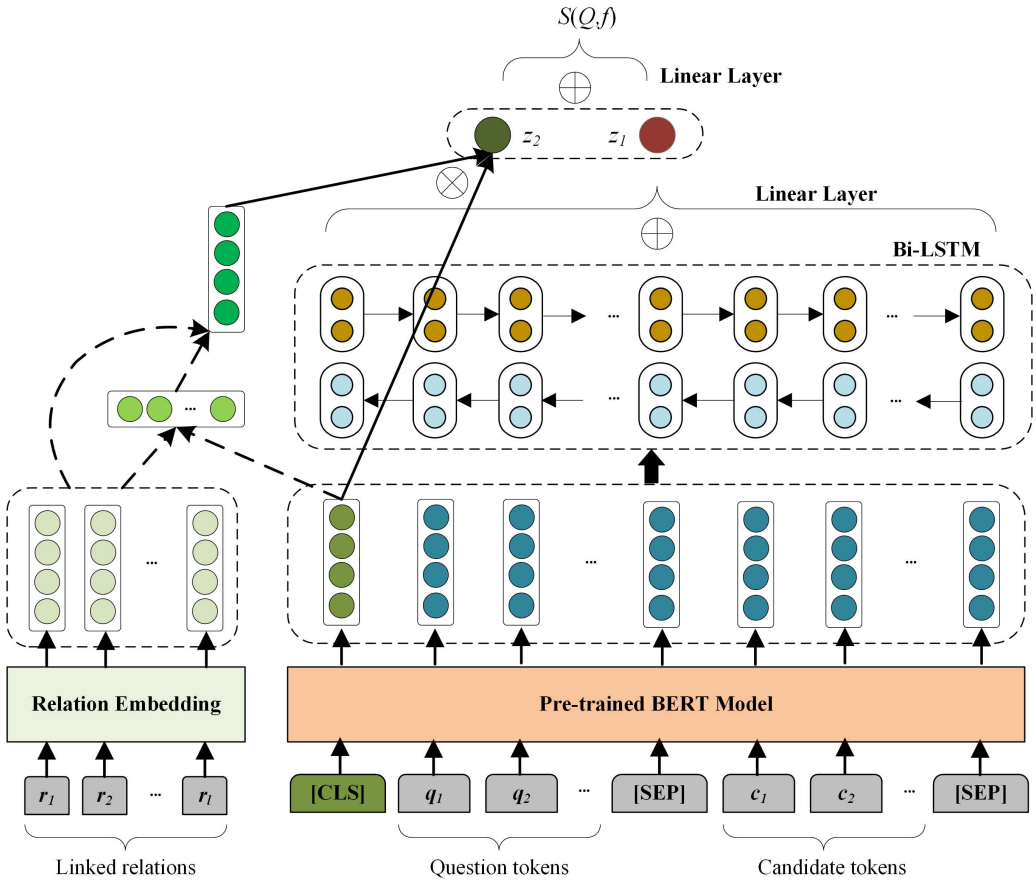

Fig. 2. The architecture of our relation detection model

$$e = \sum_{i=1}^{l} \alpha_i r_i \tag{12}$$

$$\alpha_i = softmax(v_\alpha^{\mathrm{T}} \tanh(W_\alpha^{\mathrm{T}}[r_i; x_1'])) \tag{13}$$

where $W_t \in \mathbb{R}^{d_r \times d_{bert}}$, $v_\alpha \in \mathbb{R}^{1\times c}$ and $W_\alpha \in \mathbb{R}^{c\times(2*d_r)}$ (c is a hyperparameter).

5) COMBINATION: After two granularities of matching, we get two similarity scores (z_1, z_2). Then we utilize a linear layer to learn their respective contribution for the final matching score:

$$S(Q, f_i) = W^{\mathrm{T}}[z_1; z_2] + b \tag{14}$$

where $[\cdot;\cdot]$ is the concatenation operator.

Our model is trained with the Margin Ranking Loss to maximizing the margin between the golden fact f^+ and the negative fact f^-:

$$loss(Q, f^+, f^-) = [\lambda + S(Q, f^-) - S(Q, f^+)]_+ \tag{15}$$

where λ is a hyper-parameter.

IV. Experiments

A. DATASET

We conduct experiments on the SimpleQuestions dataset [11]. SimpleQuestions provides 108,442 single-relation questions and their answer facts, which are triples in Freebase.

TABLE I
STATISTICS OF FB2M AND FB5M

	FB2M	FB5M
Entities	2,150,604	4,904,397
Relations	6,701	7,523
Triples	14,180,937	22,441,880

The dataset is split into a training set, a validation set, and a test set, with 75,910, 10,845, and 21,687 question-fact pairs, respectively. The benchmark also provides two subsets of Freebase: FB2M and FB5M, whose statistics are shown in Table.I. For this dataset, the standard evaluation metric is accuracy.

B. EXPERIMENTAL SETTINGS

In both entity linking and relation detection models, we use the pre-trained *bert-base-uncased* model from *pytorch-transformers*[1], where the number of Transformer blocks is 12, the hidden size is 768, and the number of self-attention heads is 12. For optimization, parameters are trained using Adam optimizer [32] with an initial learning rate of 5e-5 in both two

[1]https://github.com/huggingface/transformers

TABLE II
THE TEST SET ACCURACY(%) OF DIFFERENT MODELS.

Models	FB2M	FB5M
Bordes *et al.*(2015) [11]	63.9	62.7
Golub *et al.*(2016) [20]	70.9	70.3
Dai *et al.*(2016) [12]	-	75.7
Lukovnikov *et al.*(2017) [24]	71.2	-
Qu *et al.*(2018) [13]	77.9	76.8
Hao *et al.*(2018) [21]	80.2	-
Lan *et al.*(2019) [28]	**80.9**	-
Lan *et al.*(w/o enhanced)	79.2	-
Ours	**80.9**	**80.7**
Ours *et al.*(w/o enhanced)	79.9	-

models except the relation embedding whose learning rate is 1e-3 in relation detection model. And the batch size is set to 64, 3200 in entity linking and relation detection.

For relation detection, the hidden size of Bi-LSTM is 128. To prevent overfitting, we also use dropout with a dropout rate of 0.1. During training, all relation embeddings are randomly initialized with 200 dimensions. The margin λ in loss function is set to 1.0 and the negative sampling sizes is 100.

C. EXPERIMENTAL RESULTS

1) OVERALL RESULTS: The overall result in this paper is compared with other works that have performed well in the SimpleQuestions dataset in recent years. These works include the Memory Network based model of Bordes *et al.* [11], the character-level attention-based model of Golub *et al.* [20], the CFO model of Dai *et al.* [12], the multi-level GRU based model of Lukovnikov *et al.* [24], the AR-SMCNN model of Qu *et al.* [13], the pattern-revising joint select model of Hao *et al.* [21], and the enhanced matching-aggregation model of Lan *et al.* [28].

We adopt the same metrics as Bordes *et al.* [11] do, which compare the (head entity, relation) pair to the ground truth. If the head entity and relation we select both match the ground truth, the answer to the question is correct. From the results in Table.II, we can see that:

1) Our approach achieves an accuracy of 80.7% on FB5M setting, outperforming the previous state-of-art results by 3.9%. And we got an accuracy of 80.9% on FB2M setting, which is the current state-of-the-art results.
2) Lan *et al.* [28] also achieve 80.9% on FB2M setting, they employ a matching-aggregation model for sequence matching. Besides, they propose to use the question-specific contextual relations to enhance the candidate sequence. When neither uses external knowledge, the accuracy of our BERT-BiLSTM model is 0.7% higher than the matching-aggregation model, which shows that our model is better at extracting text features between questions and candidates.

To further evaluate our approach, we also follow [14], [15] in matching the question and the candidate answer on the text level rather than the KB facts level. This means that a question is counted as correct if the selected entity and the true entity have the same name attribute and the predicted relation is correct. Since there are many entities with the same name in Freebase and we do not need to distinguish them on the text level metric, the accuracy will be higher than the results shown in Table.II. We can observe from Table.III that our approach improves the state-of-the-art accuracy of the SimpleQuestions dataset from 82.29% to 89.1% on the text level metric. It means that our model can effectively capture the text matching information between the question and the answer fact.

TABLE III
THE ACCURACY(%) ON THE TEXT LEVEL METRIC

Models	FB2M	FB5M
Yin *et al.*(2016) [15]	76.4	75.9
Wang *et al.*(2019) [14]	82.1	82.29
Ours	**88.9**	**89.1**

Overall, all these results indicate that our approach is very competitive compared with other models on SR-QA task. And our approach also performs well on a relaxed evaluation metric.

2) EFFECT OF ENTITY LINKING MODEL: To evaluate the effectiveness of our proposed entity linking method, we compare it with several baselines, including focused pruning method from Dai *et al.* [12], active entity linker from Yin *et al.* [15], candidate generation method from Lukovnikov *et al.* [24], and heuristic extension method from Qu *et al.* [13] To compare with the early approaches, we sort the candidate entities by their out-degrees, which is the same as Qu *et al.* [13]. The recall is computed as the percentage of questions whose correct entity is in the top K of the candidate entities.

Table.IV shows the recall of top K candidate entities. As we can see, our model achieved the best recall from top1 to top400. It proves that our entity linking model can effectively reduce the noise of the candidate entities.

As Table.V shown, we can see that:

1) Our entity linking method gets 98.1% overall recall, which performs 1.4% higher than the state-of-the-art result. The model without the heuristic algorithm achieves can already outperform the state-of-the-art recall, which shows that BERT can effectively improve the performance of the sequence labeling models in entity linking. When augmented with the heuristic algorithm, the recall of our model increases 0.8%, indicating that our heuristic algorithm does lift the quality of the candidate entities.
2) Yin *et al.* [15] achieve higher recall but Qu *et al.* [13] get a smaller candidate set. Compared with them, our approach can not only improve the overall performance but also reduce the noise in the candidate set.

In theory, we can use all possible n-gram fragments for question traversal to achieve 100% recall. However, it is time-consuming and would introduce a lot of irrelevant entities. Unlike the previous works that use a model to narrow down the search range, we get the entity mention in question by

TABLE IV
THE RECALL(%) OF TOP-K CANDIDATE ENTITIES

Top K	Yin *et al.* [15]	Lukovnikov *et al.* [24]	Qu *et al.* [13]	Ours
1	73.6	71.2	74.5	**77.1**
5	85.0	85.4	86.0	**88.9**
10	87.4	88.4	88.5	**91.2**
20	88.8	-	90.2	**92.7**
50	90.4	-	91.9	**94.2**
100	91.6	-	93.1	**95.3**
400	-	93.7	95.1	**97.1**

TABLE V
THE OVERALL RECALL(%) AND AVERAGE SIZE OF CANDIDATE ENTITIES SET

Methods	Recall	Avg size
Dai *et al.*(2016) [12]	94.9	-
Yin *et al.*(2016) [15]	96.7	162
Qu *et al.*(2018) [13]	95.8	**57**
Ours(w/o heuristic algorithm)	97.3	-
Ours	**98.1**	61

the heuristic algorithm mentioned in section III-A. Since our method does not use the substring traversal method, it is more efficient than previous works. Besides, most of our candidate entities have the same name, so our approach avoids the noise entities who have a common substring with the question.

3) ABLATION STUDY ON RELATION DETECTION: To investigate the effect of different modules in our relation detection model, we conduct experiments for our approach with different relation detection models:

- **BERT base**: This is our relation detection model without the Bi-LSTM layer and the relation-aware attention model. As the sentence pair classification method mentioned in [18], we take the hidden state of the [CLS] token in the last hidden layer of BERT as the matching feature and then get the score via a fully connected layer.
- **BERT + Bi-LSTM** This is our relation detection model that removes the relation-aware attention network. (Use only the z_1 score)
- **BERT + Relation-aware attention** This is our relation detection model that removes the Bi-LSTM layer. (Use only the z_2 score)
- **BERT + Linear + Relation-aware attention** This is our relation detection model that replaces the Bi-LSTM layer with a linear layer, whose input size and output size are the same as the Bi-LSTM layer.
- **BERT + BiLSTM + Relation-agv** This is our relation detection model that replaces the attention operate in the Relation-aware attention network with taking the average of all the relation embeddings.

Notice that the ablation experiments are conducted only on FB2M. As shown in Table.VI, we can see that:

1) The accuracy of our original approach is 79.5%. When we add Bi-LSTM, relation-aware attention network and both of them to the original model, the accuracy increases 0.5%, 0.7%, and 1.4%, indicating that either module is important for our approach.
2) When we replace Bi-LSTM with a linear layer to extract the high-level feature, the accuracy dropped by 0.4%, mainly because Bi-LSTM can extract the dependent information of the input sequence, while the linear layer can only model the input sequence independently. At the same time, it was 0.3% higher than the model without the extractor. We conclude that the Bi-LSTM in our model can effectively extract high-level interaction features between questions and candidates.
3) When we use the relevant relations, even we just simply take the average of all the relation embeddings, the performance of our model is improved. And with the attention mechanisms, the performance can be further improved. It proves that the relevant relations can enhance the representation of candidate facts, and the model with attention mechanism can make better use of these relations.

TABLE VI
THE ACCURACY(%) OF USING DIFFERENT FEATURE EXTRACTION METHOD FOR RELATION DETECTION

Methods	Accuracy
Our approach(fully integrated)	**80.9**
BERT base	79.5
BERT + Bi-LSTM (w/o z_2)	80.0
BERT + Relation-aware attention (w/o z_1)	80.2
BERT + Linear + Relation-aware attention	80.5
BERT + BiLSTM + Relation-agv	80.4

V. ERROR ANALYSIS

We also conduct the error analysis on our approach and examine the reasons for the mistakes. There are 3951 questions answered incorrectly with our approach. We sample 50 wrongly answered questions to analyze the reasons for the mistakes. The errors can be divided into the following categories:

- **Question ambiguity (52%)**: This category of errors are caused by ambiguous question descriptions. In this case, there may be multiple answers to a question, and although our method chooses one of the potential answers, we still label it as a prediction error. For example, for the question "which artist recorded the track Jungle Fever?", the golden answer is "The Tornados", who recorded the track "Jungle Fever". And our predicted answer is "Stevie Wonder", who also recorded a track with the same name.
- **Model error(26%)**: There are some questions that our approach can not answer correctly. For example, for the question "how was 2001 released", the groud truth is "album" but the answer to our method is "Stereopoly", the name of the record.
- **Dataset Noise(16%)**: We also meet the error caused by the incorrect labels in the original SimpleQuestions

dataset as previous research found [14], [21]. For example, for the question "what is Thomas Tallis 's career ?", the golden relation is "common.topic.notable_types" but our predicted relation is "people.person.profession", though both of them can lead to the same answer "composer", we consider it to be a prediction error. In this case, we think that the relation we predicted is more precise.

- **KB redundancy(6%)**: There are some redundant entities in the KB which have exactly the same attributes but different ids. For example, for the question "how is the book Eric Brighteyes bound?", both "m.04vg19f" and "m.04vg194", are in the candidate entities set, and they actually refer to the same entity. Since the entity we randomly choose that is different from the target entity, we label it as a mistake.

VI. Conclusion

In this paper, we propose a novel approach for KB-based single-relation question answering. In order to reduce the noise in the candidates, we present a BERT-based model and a heuristic algorithm in the entity linking process. To preserve the original matching information between questions and candidate facts, we construct the question-fact pair as input to our relation detection model. And we also use a relation-aware attention network to enhance the representation of candidate KB facts. The experimental results on the SimpleQuestions dataset show that our complete approach achieves state-of-the-art accuracies of 80.9% and 80.7% based on FB2M and FB5M.

In future work, we plan to focus on the utilization of the external KB information. Furthermore, we would like to extend our approach to deal with multi-relation problems.

References

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, AcM, 2008.

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, ACM, 2007.

[3] 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*, pp. 722–735, Springer, 2007.

[4] W. Zhao, T. Chung, A. Goyal, and A. Metallinou, "Simple question answering with subgraph ranking and joint-scoring," *arXiv preprint arXiv:1904.04049*, 2019.

[5] H. Jin, Y. Luo, C. Gao, X. Tang, and P. Yuan, "Comqa: Question answering over knowledge base via semantic matching," *IEEE Access*, 2019.

[6] M. Wei and Y. Zhang, "Natural answer generation with attention over instances," *IEEE Access*, vol. 7, pp. 61008–61017, 2019.

[7] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, 2013.

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

[9] J. Berant and P. Liang, "Semantic parsing via paraphrasing," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1415–1425, 2014.

[10] W.-t. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 643–648, 2014.

[11] A. Bordes, N. Usunier, S. Chopra, and J. Weston, "Large-scale simple question answering with memory networks," *arXiv preprint arXiv:1506.02075*, 2015.

[12] Z. Dai, L. Li, and W. Xu, "Cfo: Conditional focused neural question answering with large-scale knowledge bases," *arXiv preprint arXiv:1606.01994*, 2016.

[13] Y. Qu, J. Liu, L. Kang, Q. Shi, and D. Ye, "Question answering over freebase via attentive rnn with similarity matrix based cnn," *arXiv preprint arXiv:1804.03317*, vol. 38, 2018.

[14] R.-Z. Wang, Z.-H. Ling, and Y. Hu, "Knowledge base question answering with attentive pooling for question representation," *IEEE Access*, vol. 7, pp. 46773–46784, 2019.

[15] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schütze, "Simple question answering by attentive convolutional neural network," *arXiv preprint arXiv:1606.03391*, 2016.

[16] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[17] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/language-unsupervised/language_understanding_paper. pdf*, 2018.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] M. Yu, W. Yin, K. S. Hasan, C. d. Santos, B. Xiang, and B. Zhou, "Improved neural relation detection for knowledge base question answering," *arXiv preprint arXiv:1704.06194*, 2017.

[20] D. Golub and X. He, "Character-level question answering with attention," *arXiv preprint arXiv:1604.00727*, 2016.

[21] Y. Hao, H. Liu, S. He, K. Liu, and J. Zhao, "Pattern-revising enhanced simple question answering over knowledge bases," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3272–3282, 2018.

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

[23] Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, 1989.

[24] D. Lukovnikov, A. Fischer, J. Lehmann, and S. Auer, "Neural network-based question answering over knowledge graphs on word and character level," in *Proceedings of the 26th international conference on World Wide Web*, pp. 1211–1220, International World Wide Web Conferences Steering Committee, 2017.

[25] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," *arXiv preprint arXiv:1406.3676*, 2014.

[26] H. Bast and E. Haussmann, "More accurate question answering on freebase," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1431–1440, ACM, 2015.

[27] X. Yao, J. Berant, and B. Van Durme, "Freebase qa: Information extraction or semantic parsing?," in *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pp. 82–86, 2014.

[28] Y. Lan, S. Wang, and J. Jiang, "Knowledge base question answering with a matching-aggregation model and question-specific contextual relations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1629–1638, 2019.

[29] Y. Goldberg, "Assessing bert's syntactic abilities," *arXiv preprint arXiv:1901.05287*, 2019.

[30] G. Jawahar, B. Sagot, D. Seddah, S. Unicomb, G. Iñiguez, M. Karsai, Y. Léo, M. Karsai, C. Sarraute, É. Fleury, *et al.*, "What does bert learn about the structure of language?," in *57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy*, 2019.

[31] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.