Question Answering over Knowledge Base with Neural Attention Combining Global Knowledge Information

Yuanzhe Zhang*, **Kang Liu***, **Shizhu He***, **Guoliang Ji***, **Zhanyi Liu**†, **Hua Wu**†, **Jun Zhao***
* Institute of Automation, Chinese Academy of Sciences
{yzzhang, kliu, shizhu.he, guoliang.ji, jzhao}@nlpr.ia.ac.cn
† Baidu Inc.
{liuzhanyi, wu_hua}@baidu.com

Abstract

With the rapid growth of knowledge bases (KBs) on the web, how to take full advantage of them becomes increasingly important. Knowledge base-based question answering (KB-QA) is one of the most promising approaches to access the substantial knowledge. Meantime, as the neural network-based (NN-based) methods develop, NN-based KB-QA has already achieved impressive results. However, previous work did not put emphasis on question representation, and the question is converted into a fixed vector regardless of its candidate answers. This simple representation strategy is unable to express the proper information of the question. Hence, we present a neural attention-based model to represent the questions dynamically according to the different focuses of various candidate answer aspects. In addition, we leverage the global knowledge inside the underlying KB, aiming at integrating the rich KB information into the representation of the answers. And it also alleviates the out of vocabulary (OOV) problem, which helps the attention model to represent the question more precisely. The experimental results on WEBQUESTIONS demonstrate the effectiveness of the proposed approach.

1 Introduction

As the amount of the knowledge bases (KBs) grows, people are paying more attention to seeking effective methods for accessing these precious intellectual resources. There are several tailor-made languages designed for querying KBs, such as SPARQL [PrudHommeaux *et al.*, 2008]. However, to handle such query languages, users are required to not only be familiar with the particular language grammars, but also be aware of the vocabularies of the KBs. By contrast, knowledge base-based question answering (KB-QA) [Unger *et al.*, 2014], which takes natural language as query language, is a more user-friendly solution, and has become a research focus in recent years.

The goal of KB-QA is to automatically return answers from the KB given natural language questions. There are two mainstream research directions for this task, i.e., semantic parsing-based (SP-based) [Zettlemoyer and Collins, 2005; Zettlemoyer and Collins, 2009; Kwiatkowski *et al.*, 2013; Cai and Yates, 2013; Berant *et al.*, 2013; Yih *et al.*, 2015] and information retrieve-based (IR-based) [Yao and Van Durme, 2014; Bordes *et al.*, 2014b; Bordes *et al.*, 2014a; Dong *et al.*, 2015; Bordes *et al.*, 2015] methods. SP-based methods usually focus on constructing a semantic parser that could convert natural language questions into structured expressions like logical forms. IR-based methods are more like to search answers from the KB based on the information conveyed in the questions. Here, ranking techniques are often adopted to make correct selections from candidate answers. In general, IR-based methods are easier and more flexible to implement. [Dong *et al.*, 2015; Bordes *et al.*, 2015] have proven that IR-based methods could acquire competitive performance compared with SP-based methods through the experiments conducted over Freebase [Bollacker *et al.*, 2008].

Recently, with the progress of deep learning, neural network-based (NN-based) methods have been introduced to the KB-QA task [Bordes *et al.*, 2014b]. They belong to IR-based methods. Different from previous methods, NN-based methods represent both the questions and the answers as semantic vectors. Then the complex process of KB-QA could be converted into a similarity matching process between an input question and its candidate answers in a semantic space. The candidates with the highest similarity score will be considered as the final answers. Because they are adaptive and robust, NN-based methods have attracted more and more attention, and this paper also focus on using neural networks to answer questions over knowledge base.

In NN-based methods, the crucial step is to compute the similarity score between a question and a candidate answer, where the key is to learn their representations. Previous methods put more emphasis on learning representations of the answer end. For example, [Bordes *et al.*, 2014a] considers the importance of the subgraph of the candidate answers. [Dong *et al.*, 2015] makes use of the context and the type of the answers. By contrast, the representation methods of the question end are oligotrophic. Existing approaches often represent a question into a single vector using a simple bag-of-words (BOW) model [Bordes *et al.*, 2014b; Bordes *et al.*, 2014a], whereas its relatedness to the answer end is neglected. We argue that a question should be represented differently according to the different focuses of various answer aspects[1].

Take question “Who is the president of France?” and one of its candidate answers “Francois Hollande” as an example. When dealing with the answer entity Francois Holland, “president” and “France” in the question is more focused, and the question representation should bias towards the two words. While facing the answer type /business/board_member, “Who” should be the most prominent word. Obviously, this is an attention mechanism, which reflects how the focus of answer aspects could influence the representation of the question.

When learning the representations of the questions, we should make proper use of each word in the question according to different attention of each aspect of the candidate answer, instead of simply compressing them into a fixed vector. We believe that such kind of representations are more expressive. [Dong *et al.*, 2015] represents questions using three CNNs with different parameters when dealing with different answer aspects including the answer path, the answer context and the answer type. We think simply selecting three independent CNNs is mechanical and inflexible. Thus, we go one step further, and propose an attention-based neural network to perform question answering over KB. Different to [Dong *et al.*, 2015], we represent the question differently according to different answer resources, not allowing them sharing the same network as [Dong *et al.*, 2015] does. For instance, /business/board_member and /location/country are both answer types, but the question representation will be different according to their different attention in our method.

On the other hand, we notice that the representations of the KB resources (entities and relations) are also limited in previous work. To be specific, they are often learned barely on the QA training data, which results in two limitations. 1) *The deficiency of the global information of the KB*. The previous methods merely utilize the answer-related part in the KB, i.e., answer path and answer context [Bordes *et al.*, 2014a; Dong *et al.*, 2015], to learn the representations of KB resources. The global information of the KB is completely ignored. For example, if question-answer pair (q, a) appears in the training data, and the global KB information implies us that a' is similar to a [2], denoted by $(a \sim a')$, then (q, a') is more probable to be right. However, current QA training mechanism cannot guarantee $(a \sim a')$ could be learned. 2) *The problem of out of vocabulary (OOV)*. Due to the limited coverage of the training data, the OOV problem is common while testing, and many answer entities in testing candidate set have never been seen before. In this scenario, the representation of such unseen KB resources could not be learned precisely. The attention of these resources become the same because they shared the same OOV embedding, and this will do harm to the proposed attention model. To tackle these two problems, we additionally incorporates KB itself as training data for training embeddings besides original question-answer pairs. In this way, the global structure of the whole knowledge could be captured, and the OOV problem could be alleviated naturally.

In summary, the contributions of this paper are as follows.

1) We present a novel attention-based NN model tailored to the KB-QA task, which considers the influence of the answer aspects for representing questions.

2) We leverage the global KB information, aiming at representing the answers more precisely. It also alleviates the OOV problem.

3) The experimental results on the open dataset WEBQUESTIONS demonstrate the effectiveness of the proposed approach.

2 Overview

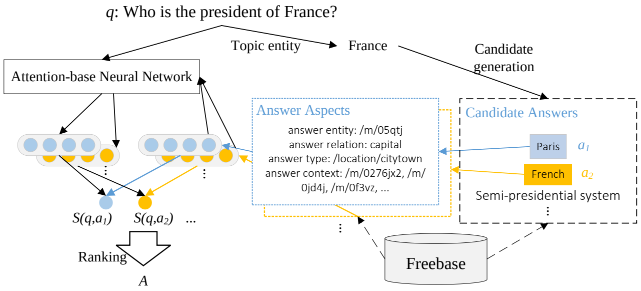

Figure 1: The overview of the proposed KB-QA system.

The goal of the KB-QA task could be formulated as follows. Given a natural language question q, return an entity set A as answers. The architecture of our proposed KB-QA system is shown in Figure 1, which illustrates the basic flow of our approach. First, we identify the topic entity of the question, and generate candidate answers from Freebase. Then, the candidate answers are represented with regard to their four aspects. Next, an attention-based neural network is employed to represent the question under the influence of the candidate answer aspects. Finally, the similarity score between the question and each corresponding candidate answer is calculated, and the candidates with the highest score will be selected as the final answers[3].

We utilize Freebase [Bollacker *et al.*, 2008] as our knowledge base. It now has more than 3 billion facts, and is used as the supporting KB for many QA tasks. In Freebase, the facts are represented by *subject-property-object* triples (s,p,o). For clarity, we call each basic element a resource, which could be either an entity or a relation. For example, (/m/0f8l9c, location.country.capital, /m/05qtj)[4] describe the fact that the capital of France is Paris, /m/0f8l9c and /m/05qtj are entities denoting France and Paris respectively, and location.country.capital is a relation.

[1] An answer aspect could be the answer entity itself, the answer type, the answer context, etc.

[2] The complete KB is able to offer this kind of information, e.g., a and a' share massive context.

[3] We also adopt a margin strategy to obtain multiple answers for a question and this will be explained in the next section.

[4] Note that the Freebase prefixes are omitted for neatness.

3 Our Approach

3.1 Candidate Generation

The candidate answers should be all the entities of Freebase ideally, but in practice, this is time consuming and not really necessary. For each question q, we can use Freebase API [Bollacker *et al.*, 2008] to identify a *topic entity*, which could be simply understood as the main entity of the question. For example, `France` is the topic entity of question "Who is the president of France?". Freebase API method is able to resolve as many as 86% questions if we use the top1 result [Yao and Van Durme, 2014]. After getting the topic entity, we collect all the entities directly connected to it and the ones connected with 2-hop[5]. These entities constitute a candidate set C_q .

3.2 The Proposed Neural Attention Model

We present an attention-based neural network, which represents the question dynamically according to different answer aspects. Concretely, each aspect of the answer pays different attention to the question and thus decides how the question is represented. The extent of the attention is used as the weight of each word in the question. Figure 2 is the architecture of our model. We will illustrate how the system works as follows.

LSTM

First of all, we have to obtain the representation of each word in the question. These representations retain all the information of the question, and could serve the following steps. Suppose question q is expressed as $q = (x_1, x_2, ..., x_n)$, where x_i denotes the ith word. As shown in Figure 2, we first look up a word embedding matrix $E_w \in \mathbb{R}^{d \times v_w}$ to get the word embeddings, which is randomly initialized, and updated during the training process. Here, d means the dimension of the embeddings and v_w denotes the vocabulary size of natural language words. Then, the embeddings are fed into a long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] networks. LSTM has been proven to be effective in many natural language processing (NLP) tasks such as machine translation [Sutskever *et al.*, 2014] and dependency parsing [Dyer *et al.*, 2015], and it is adept in harnessing long sentences. Note that if we use unidirectional LSTM, the outcome of a specific word contains only the information of the words before it, whereas the words after it is not taken into account. To avoid this, we employ bidirectional LSTM as [Bahdanau *et al.*, 2015] does, which consists of both forward and backward networks. The forward LSTM handles the question from left to right, and the backward LSTM processes in the reverse order. Thus, we could acquire two hidden state sequences, one from the forward one $(\vec{h}_1, \vec{h}_2, ..., \vec{h}_n)$ and the other from the backward one $(\overleftarrow{h}_1, \overleftarrow{h}_2, ..., \overleftarrow{h}_n)$. We concatenate the forward hidden state and the backward hidden state of each word, resulting in $h_j = [\vec{h}_j; \overleftarrow{h}_j]$. The hidden unit of forward and backward LSTM is $\frac{d}{2}$, so the concatenated vector is of dimension d. In this way, we obtain the representation of each word in the question.

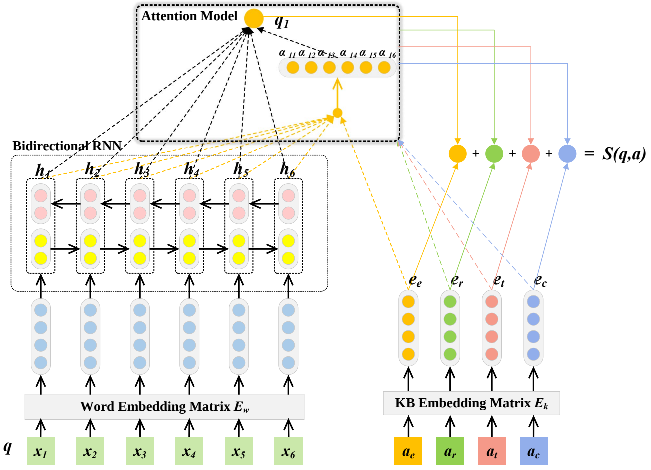
Attention Model
q_1
α_{11} α_{12} α_{13} α_{14} α_{15} α_{16}
Bidirectional RNN
h_1 h_2 h_3 h_4 h_5 h_6
Word Embedding Matrix E_w
q
x_1 x_2 x_3 x_4 x_5 x_6
e_e e_r e_t e_c
KB Embedding Matrix E_k
a_e a_r a_t a_c
+ + + = S(q,a)

Figure 2: The architecture of the proposed attention-based neural network. Note that only one aspect (in orange color) is depicted for clarity. The other three aspects follow the same way.

Answer aspect representation

In the answer end, we directly use the embedding for each answer aspect through the KB embedding matrix $E_k \in \mathbb{R}^{d \times v_k}$. Here, v_k means the vocabulary size of the KB resources. The embedding matrix is randomly initialized and learned during training, and could be further enhanced with the help of the global information as described in Section 3.3. Concretely, we employ four kinds of answer aspects, namely, *answer entity* a_e, *answer relation* a_r, *answer type* a_t and *answer context*[6] a_c. Their embeddings are denoted as e_e, e_r, e_t and e_c respectively. It is worth noting that the answer context consists of multiple KB resources, and we denote it as $(c_1, c_2, ..., c_n)$. We first acquire their KB embeddings $(ec_1, ec_2, ..., ec_n)$ through E_k, then calculate an average embedding by $e_c = \frac{1}{n} \sum_{i=1}^{n} ec_i$.

Attention model

The most crucial part of the proposed approach is the attention mechanism. Based on our assumption, each answer aspect should have different attention towards the same question. The extent of attention can be measured by the relatedness between each word representation h_j and an answer aspect embedding e_i. We propose the following formulas to calculate the weights.

$$\alpha_{ij} = \frac{\exp(w_{ij})}{\sum_{k=1}^{n} \exp(w_{ik})} \tag{1}$$

$$w_{ij} = W^T(tanh[h_j; e_i]) + b \tag{2}$$

Here, α_{ij} denotes the attention weight of the jth word in the question, in terms of answer aspect e_i, where $e_i \in \{e_e, e_r, e_t, e_c\}$. n is the length of the question. $W \in \mathbb{R}^{2d \times 1}$ is

[5] For example, (/m/0f8l9c, governing_officials, government_position_held.office_holder, /m/02qg4z) is a 2-hop connection.

[6] Here, the entities that directly connected to the answer entity is regarded as the answer context.

an intermediate matrix and b is an offset value. Both of them are randomly initialized and updated during training. Subsequently, the attention weights are employed to calculate a weighted sum of the words, resulting in a semantic vector that represent the question, according to the specific answer aspect e_i.

$$q_i = \sum_{j=1}^{n} \alpha_{ij} h_j \tag{3}$$

By now, the similarity score of question q and this particular candidate answer a could be defined as follows.

$$S(q,a) = \sum_{e_i \in \{e_e, e_r, e_t, e_c\}} q_i \cdot e_i \tag{4}$$

The proposed attention model could also be intuitively interpreted as a re-reading mechanism [Hermann *et al.*, 2015]. Our aim is to select correct answers from a candidate set. When we consider a candidate answer, suppose we first look at its type, and we will re-read the question to find out which part of the question should be more focused (handling attention). Then we go to next aspect and re-read the question again, until the all the aspects are utilized. We believe that this mechanism is beneficial for the system to better understand the question with the help of the answer aspects, and leads to a performance promotion.

Training

We first construct the training data. Since we have question-answer pairs (q, a) as supervision data, candidate set C_q of question q can be divided into two subsets, namely, correct answer set P_q and wrong answer set N_q. For each correct answer $a \in P_q$, we randomly select k wrong answers $a' \in N_q$ as negative examples. For some topic entities, there may be not enough wrong answers to acquire k wrong answers. Under this circumstance, we extend N_q from other randomly selected candidate set $C_{q'}$. With the generated training data, we are able to make use of pairwise training.

The training loss is given as follows.

$$L_{q,a,a'} = [\gamma + S(q,a') - S(q,a)]_+ \tag{5}$$

Where γ is a positive real number that ensure a margin between positive and negative examples. And $[z]_+$ means $\max(0, z)$. The intuition of this training strategy is to guarantee the score of positive question-answer pairs be higher than negative ones with a margin.

The objective function is as follows.

$$\min \sum_{q} \frac{1}{|P_q|} \sum_{a \in P_q} \sum_{a' \in N_q} L_{q,a,a'} \tag{6}$$

We adopt stochastic gradient descent (SGD) to implement the learning process, mini-batches are utilized.

Inference

In testing stage, we straightforwardly take advantage of the candidate answer set C_q of the question. We have to calculate $S(q, a)$ for each $a \in C_q$, and find out the maximum value S_{max}.

$$S_{max} = \arg\max_{a \in C_q} \{S(q,a)\} \tag{7}$$

It is worth noting that many questions have more than one answer, so it is improper to set S_{max} as the final answer. Instead, we make use of the margin γ in the loss function, if the score of an candidate answer is within the margin compared with s_{max}, we put it in the final answer set.

$$A = \{\hat{a} | S_{max} - S(q, \hat{a}) < \gamma\} \tag{8}$$

3.3 Combining Global Knowledge Information

In this section, we elaborate how the global information of the KB could be leveraged. As stated before, we try to take into account the complete structural information of the KB. To this end, we adopt TransE model [Bordes *et al.*, 2013] to represent the KB, and integrate the representations into the QA training process.

In TransE model, the entities and relations are represented by low dimensional embeddings. The basic idea is that the relations are regarded as translations in the embedding space. Here, for consistency, we denote each fact as (s, p, o), and use boldface $(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{o})$ to denote their embeddings. The embedding of the tail entity $\boldsymbol{o}$ should be close to the embedding of head entity $\boldsymbol{s}$ plus the embedding of relation $\boldsymbol{p}$, i.e., $(\boldsymbol{s} + \boldsymbol{p} \approx \boldsymbol{o})$. The energy of a triple (s, p, o) is equal to $d(s{+}p, o)$ for some dissimilarity d, defined as $\|\boldsymbol{s} + \boldsymbol{p} - \boldsymbol{o}\|_2^2$. To learn the embeddings, TransE minimizes the following loss function.

$$L_k = \sum_{(s,p,o) \in S} \sum_{(s',p,o') \in S'} [\gamma_k + d(s+p, o) - d(s'+p, o')]_+ \tag{9}$$

Where S is the set of KB facts and S' is the corrupted facts, which is composed of positive facts with either the head or tail replace by a random entity. The loss function favors lower values of the energy for positive facts than for negative facts.

In our implementation, we filter out the completely unrelated facts to save time. To be more specific, we first collect all the topic entities of all the questions as initial set. Then, we expand the set by adding direct connected and 2-hop entities. Finally, all the facts in which these entities appeared form the positive set. The negative facts are randomly corrupted ones. This a compromise solution due to the large scale of Freebase.

To combine the global information to our training process, we adopts a multi-task training strategy. Specifically, we perform our KB-QA training and TransE training in turn. After each epoch of KB-QA training, 100 epochs of TransE training is conducted, and the embeddings of the KB resources are shared and updated during both training processes. The proposed training process ensures that the global KB information act as additional supervision, and the interconnections among the resources are fully considered. In addition, as more KB resources are involved, the OOV problem will be relieved, which is able to bring additional benefits to the attention model.

4 Experiments

4.1 Datasets

To evaluate the proposed method, we select WEBQUESTIONS [Berant *et al.*, 2013] dataset that includes 3,778 question-answer pairs for training and 2,032 for testing. The questions are collected from Google Suggest API, and the answers are labeled manually by Amazon MTurk. All the answers are from Freebase. We use three-quarter (2,833) of the training data as training set, and the remaining quarter as validate set. F_1 score computed by the script provided by [Berant *et al.*, 2013] is select as the evaluation metric

4.2 Settings

For KB-QA training, we use mini-batch stochastic gradient descent to minimize the pairwise training loss. The mini-batch size is set to 50. The learning rate is set to 0.01. Both the word embedding matrix E_w and KB embedding matrix E_v are normalized after each epoch. The embedding size $d = 128$, and the hidden unit size is 64. Margin γ is set to 0.6. Negative example number $k = 500$. The TransE training process defines the embeddings dimension to 128, and the mini-batch size is also 50. γ_k is set to 1. All these hyperparameters of the proposed network is determined according to the performance on the validate set.

4.3 Results

The Effectiveness of the proposed approach

To demonstrate the effectiveness of the proposed approach, we compare our method with previous NN-based methods. Table 1 shows the results on WEBQUESTIONS test set. The methods listed in the table all employ neural network for KB-QA. [Bordes *et al.*, 2014b] applies BOW method to obtain a single vector for both questions and answers. [Bordes *et al.*, 2014a] further improves their work by proposing the concept of subgraph embeddings. Besides the answer path, the subgraph contains all the entities and relations connected to the answer entity. The final vector is also obtained by BOW strategy. [Yang *et al.*, 2014] follows the SP-based manner, but uses embeddings to map entities and relations into KB resources, then the question can be converted into logical forms. They jointly consider the two mapping process. [Dong *et al.*, 2015] uses three columns of CNNs to represent questions corresponding to three aspects of the answers, namely the answer context, the answer path and the answer type. [Bordes *et al.*, 2015] puts KB-QA into the memory networks [Sukhbaatar *et al.*, 2015] framework, and achieves the state-of-the-art performance. **ours** represents the proposed approach.

Method	F_1
Bordes et al., 2014b	29.7
Bordes et al., 2014a	39.2
Yang et al., 2014	41.3
Dong et al., 2015	40.8
Bordes et al., 2015	42.2
ours	**42.6**

Table 1: The evaluation results on WEBQUESTIONS.

From the results, we can observe that **ours** achieves the best performance on WEBQUESTIONS. Here [Bordes *et al.*, 2014b; Bordes *et al.*, 2014a; Bordes *et al.*, 2015] all utilize BOW model to represent the questions, while ours takes advantage of the attention of answer aspects to dynamically represent the questions. Also note that [Bordes *et al.*, 2015] uses additional training data such as Reverb [Fader *et al.*, 2011] and their original dataset SimpleQuestions. [Dong *et al.*, 2015] employs three fixed CNNs to represent questions, while ours is able to express the focus of each unique answer aspect in the question representation. Besides, the global KB information is leveraged. So, we believe that the results faithfully show that the proposed approach is more effective than the other competitive methods. It is worth noting that [Yih *et al.*, 2015] achieves an F_1 of 52.5, much higher than other methods. Their staged system is able to address more questions with constraints and aggregations. However, their approach applies numbers of manually designed rules and features, which come from the observations on the training set questions. These particular manual efforts reduce the adaptability of their approach.

Model Analysis

In this part, we further discuss the impacts of the components of our model. Table 2 indicates the effectiveness of different parts in the model.

Method	F_1
LSTM	38.2
Bi_LSTM	38.9
Bi_LSTM + ATT	41.6
Bi_LSTM + GKI	40.4
Bi_LSTM + ATT + GKI	42.6

Table 2: The results of different models.

LSTM employs unidirectional LSTM, and uses the last hidden state as the question representation. *Bi_LSTM* adopts a bidirectional LSTM. If we use $(\overrightarrow{h}_1, \overrightarrow{h}_2, ..., \overrightarrow{h}_n)$ to denote the forward LSTM, and use $(\overleftarrow{h}_1, \overleftarrow{h}_2, ..., \overleftarrow{h}_n)$ to indicate backward LSTM, then the final presentation of the question is $[\overrightarrow{h}_n; \overleftarrow{h}_1]$. *Bi_LSTM+ATT* is the bidirectional LSTM with neural attention (four answer aspects are used). *Bi_LSTM+GKI* denote the bidirectional LSTM model with global KB information (GKI). *Bi_LSTMS+ATT+GKI* is the same as **ours**, which is the bidirectional LSTM model with both attention model and global KB information.

From the results, we could observe the followings.

1) *Bi_LSTM+ATT* dramatically improves the F_1 score by 2.7% compared with *Bi_LSTM*. Similarly, *Bi_LSTM+ATT+GKI* significantly outperforms *Bi_LSTM+GKI* by 2.2%. They straightforwardly prove that the proposed attention model is effective.

2) *Bi_LSTM+GKI* performs better than *Bi_LSTM*, and achieves a 1.5% improvement. Similarly, *Bi_LSTM+ATT+GKI* improves *Bi_LSTM+ATT* by 1%. The results indicate that the proposed training strategy successfully leverages the global information of the underlying KB.

3) *Bi_LSTM+ATT+GKI* achieves the best performance as we expected, and improves the original *Bi_LSTM* dramatically by 3.7%. This directly shows the power of the attention model and the global KB information.

To clearly demonstrate the effectiveness of the attention mechanism in our approach, we present the attention weights of a question in the form of heat maps as shown in Figure 3.

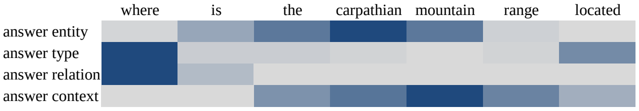

Figure 3: The visualized attention heat map. Answer entity: /m/06npd (Slovakia), answer relation: partially_containedby, answer type: /location/country, answer context: (/m/04dq9kf, /m/01mp, ...)

From this example we can observe that our methods is able to capture the attention properly. It is instructive to figure out the attention part of the question when dealing with different answer aspects. The heat map will help us understand which parts are most useful for selecting correct answers. For instance, from Figure 3, we can see that location.country is paying great attention to "Where", indicating that "Where" is much more important than the other parts in the question when dealing with this type. In other words, the other parts are not that crucial since 'Where' is strongly implying that the question is asking about a location.

4.4 Error Analysis

We randomly sample 100 imperfectly answered questions and categorize the errors into two main classes as follows.

Wrong attention

In some occasions (17 in 100 questions, 17%), we find the generated attention weights unreasonable. For instance, for question "What are the songs that Justin Bieber wrote?", answer type /music/composition pays the most attention on "What" rather than "songs". We think this is due to the bias of the training data, and we believe these errors could be solved by introducing more instructive training data in the future.

Complex questions and label errors

Another challenging problem is the complex questions (34%). For example, "When was the last time Knicks won the championship?" is actually to ask the last championship, but the predicted answers give all the championships. This is due to that the model cannot learn what does "last" mean in the training process. In addition, the label mistakes also influence the evaluation (3%) . For example, "What college did John Nash teach at?". The labeled answer is Princeton University, but Massachusetts Institute of Technology should also be an answer, and the proposed method is able to answer it correctly.

Other errors include topic entity generation error and the multiple answers error (giving more answers than expected). We guess these errors are caused by the simplest implementations of the related steps in our method, and we will not explain them in detail due to space limitation.

5 Related Work

5.1 Neural Network-based KB-QA

[Bordes *et al.*, 2014b] first applies NN-based method to solve KB-QA problem. The questions and KB triples are represented by vectors in a low dimensional space. Thus the cosine similarity could be used to find the most possible answer. BOW method is employed to obtain a single vector for both the questions and the answers. Pairwise training is utilized, and the negative examples are randomly selected from the KB facts. They also present a training data generation method, i.e., using KB facts to and some heuristics rules to generate natural language questions.

[Bordes *et al.*, 2014a] further improves their work by proposing the concept of subgraph embeddings. The key idea is to involve as much as information in the answer end. Besides the answer triple, the subgraph contains all the entities and relations connected to the answer entity. The final vector is also obtained by BOW strategy.

[Yih *et al.*, 2014] focuses on single-relation questions. The KB-QA task is divided into two parts, i.e., finding the entity mention-entity mapping and then mapping the remaining relation pattern to the KB relation. They train two CNN models to perform the mapping processes. [Yang *et al.*, 2014] handles entity and relation mapping as joint procedures. Strictly speaking, these two methods follow the SP-based manner, but they take advantage of neural networks to obtain intermediate mapping results.

The most similar work to ours is [Dong *et al.*, 2015]. They consider the different aspects of answers, using three columns of CNNs to represent questions respectively. The difference is that our approach uses attention mechanism for each unique answer aspect, so the question representation is not fixed to only three types. Moreover, we utilize the global KB information.

5.2 Attention-based Model

[Bahdanau *et al.*, 2015] first applies attention model in NLP. They improve the encoder-decoder Neural Machine Translation (NMT) framework by jointly learning alignment and translation. They argue that representing the source sentence by a fixed vector is unreasonable, and propose a soft-align method, which could be understood as the attention mechanism.

[Luong *et al.*, 2015] is also tackling machine translation task. They propose two attentions models, i.e., a global model and a local model. The latter further indicates a small scope to attend, and achieves better results.

[Rush *et al.*, 2015] implements sentence-level summarization task. They utilize local attention-based model that generate each word of the summary conditioned on the input sentence.

Our approach differs from previous work in that we are using attentions to help represent question dynamically, not generating current word from vocabulary as before.

6 Conclusion

In this paper, we focus on the KB-QA task. First, we consider the impacts of the different answers and their aspects when representing the question, and propose a novel attention-based model for KB-QA. Specifically, the attention of the answer aspect for each word in the question is used. This kind of dynamic representation is more precise and flexible. Second, we leverage the global KB information, which could take full advantage of the complete KB, and also could alleviate the OOV problem. The extensive experiments demonstrate that the proposed approach could achieve better performance compared with other state-of-the-art NN-based methods.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, 2015.

[Berant *et al.*, 2013] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*, pages 1533–1544, 2013.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, pages 1247–1250. ACM, 2008.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. 2013.

[Bordes *et al.*, 2014a] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In *Proceedings of EMNLP*, pages 615–620, 2014.

[Bordes *et al.*, 2014b] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer, 2014.

[Bordes *et al.*, 2015] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. In *Proceedings of ICLR*, 2015.

[Cai and Yates, 2013] Qingqing Cai and Alexander Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of ACL*, pages 423–433, 2013.

[Dong *et al.*, 2015] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of ACL and IJCNLP*, pages 260–269, 2015.

[Dyer *et al.*, 2015] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*, 2015.

[Fader *et al.*, 2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of EMNLP*, pages 1535–1545. Association for Computational Linguistics, 2011.

[Hermann *et al.*, 2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692, 2015.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Kwiatkowski *et al.*, 2013] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of EMNLP*, pages 1545–1556, 2013.

[Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*, pages 1412–1421, 2015.

[PrudHommeaux *et al.*, 2008] Eric PrudHommeaux, Andy Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008.

[Rush *et al.*, 2015] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. pages 379–389, 2015.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2431–2439, 2015.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[Unger *et al.*, 2014] Christina Unger, André Freitas, and Philipp Cimiano. An introduction to question answering over linked data. In *Reasoning Web. Reasoning on the Web in the Big Data Era*, pages 100–140. 2014.

[Yang *et al.*, 2014] Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. Joint relational embeddings for knowledge-based question answering. In *Proceedings of EMNLP*, pages 645–650, 2014.

[Yao and Van Durme, 2014] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*, pages 956–966, 2014.

[Yih *et al.*, 2014] Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *Proceedings of ACL*, pages 643–648, 2014.

[Yih *et al.*, 2015] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of ACL and IJCNLP*, pages 1321–1331, 2015.

[Zettlemoyer and Collins, 2005] Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of UAI*, pages 658–666, 2005.

[Zettlemoyer and Collins, 2009] Luke S Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. In *Proceedings of ACL-IJCNLP*, pages 976–984, 2009.