# Knowledge Base Question Answering Based on Deep Learning Models

Zhiwen Xie, Zhao Zeng, Guangyou Zhou(✉), and Tingting He

School of Computer, Central China Normal University, Wuhan 430079, China
{xiezhiwen,zhaozeng}@mails.ccnu.edu.cn, {gyzhou,tthe}@mail.ccnu.edu.cn

**Abstract.** This paper focuses on the task of knowledge-based question answering (KBQA). KBQA aims to match the questions with the structured semantics in knowledge base. In this paper, we propose a two-stage method. Firstly, we propose a topic entity extraction model (TEEM) to extract topic entities in questions, which does not rely on hand-crafted features or linguistic tools. We extract topic entities in questions with the TEEM and then search the knowledge triples which are related to the topic entities from the knowledge base as the candidate knowledge triples. Then, we apply Deep Structured Semantic Models based on convolutional neural network and bidirectional long short-term memory to match questions and predicates in the candidate knowledge triples. To obtain better training dataset, we use an iterative approach to retrieve the knowledge triples from the knowledge base. The evaluation result shows that our system achieves an Average$F_1$ measure of 79.57% on test dataset.

## 1 Introduction

Automatic question answering systems are aimed at returning the direct and exact answers to natural language questions. Recently, with the development of large-scale knowledge bases, such as Freebase [1] and DBPedia [2], knowledge bases become very important resources for open domain question answering. Recently, most research studies focus on the task of knowledge based QA (KBQA). In this paper, we focus on the task of NLPCC-ICCPOL 2016 KBQA for Chinese language.

The major challenge of KBQA is how to understand natural language questions and match the questions with structured semantics of knowledge bases. To address this challenge, previous work in the literature uses semantic parsing, which map the natural language question into a formal representation, such as logical form or SPARQL. However, most of the semantic parsers need to annotate logical forms of questions as supervision and rely on predefined rules and linguistic tools. The annotated logical forms always can't coverage all the predicates in the knowledge base. In addition, parsing the questions needs to recognize the topic entity which is the main entity referring to the subject of the corresponding knowledge triple in knowledge base. However, most word segmentation and named entity recognition tools are not very good and can't recognize some


C.-Y. Lin et al. (Eds.): NLPCC-ICCPOL 2016, LNAI 10102, pp. 300–311, 2016.
DOI: 10.1007/978-3-319-50496-4_25

complicated topic entities in questions. This problem can be solved by using an advanced entity linking system, but it still rely on quality of linguistic tools and always introduce many noise entities. In this paper, we use a topic entity extraction model based on deep learning to solve this problem.

Recently, deep learning models have achieved remarkable results in natural language processing, such as word vector representations [3–5], deep structured semantic models (DSSM) [6–8], machine translation [9] and text summarization [10].

In this paper, we use two deep learning strategies to address the KBQA task in two stages. Firstly, we propose a topic entity extraction model (TEEM) with deep learning, which is used to extract the topic entities in questions. Our entity extraction model does not rely on hand-draft features or linguistic tools and can extract topic entities in the questions accurately. This model is based on convolutional neural network. The results of TEEM is used to retrieve candidate knowledge triples which is related to the topic entities in the questions. Thus, we can prune the space for semantic matching and make the matching more efficient by focusing on the most related part of the knowledge base. Then, instead of pure lexical matching we apply deep structured semantic models (DSSM) to match questions with the predicates of the candidate knowledge triples in a semantic space. In this study, we develop the DSSM by using a bidirectional long short term memory neural network (BiLSTM-DSSM). To obtain more composite representation, we add an convolutional layer on the top of the BiLSTM Layer. We also leverage a recently developed structured semantic model based on convolutional neural network (C-DSSM) and we combine the scores of the DSSMs as the final semantic relevance score. The words in questions and knowledge base are represented by word vectors, which can represent words in a dense and semantic space. In this paper, we train all the models on top of word vectors obtained from an unsupervised neural language model.

To obtain better training dataset, we use an iterative approach to retrieve the knowledge triples from the knowledge base for training. Firstly, the question-answer pairs are used to search the corresponding knowledge triples from the knowledge base. The results are used as the initial training data to train the topic entity extraction model. Since the initial training data is always inaccurate, to better the training data, we use the entities extracted by the trained TEEM as additional information and retrieve the knowledge triples again. Thus, the training data will be better with less noise and the accuracy of models will improve.

The rest of this paper is structured as follows: Sect. 2 describes the related work. Section 3 introduces the details of the proposed methods; Experimental results are presented in Sect. 4. Finally, we conclude the paper in Sect. 5.

## 2 Related Work

Automatic question answering aims at returning a direct and exact answer for a natural language question. Recently, with the rise of structured and large-scale knowledge base, KBQA has attached much attention. Most of the KBQA systems are based on semantic parsing, where a question is converted into logical form and then transformed into structured query to be executed on knowledge base. Traditional semantic parsers require annotated logical forms as supervisions to train [11,12], which is very expensive. To reduce the costs, recently works focus on using question-answer pairs as weak training signals [13,14]. With the progress of deep learning in natural language processing, deep learning models are used by more and more KBQA systems and achieve a significant success. Yih et al. [15,16] developed semantic parsing frameworks based on semantic similarity by using Staged Query Graph Generation and convolutional neural network semantic models. Dong et al. [17] proposed multi-column convolutional neural networks to understand questions from three different aspects (namely, answer path, answer context, and answer type) and learn their distributed representations. Zhang [18] proposed a neural attention-based model to represent the questions dynamically according to the different aspects of various candidate answer aspects. Meanwhile, they also integrated the rich KB information into the representation of the answer.

## 3 Methods

The overview of our KBQA system framework is shown in Fig. 1. As is shown in Fig. 1, our system is mainly composed of four modules: TEEM module, DSSM module, IR module and Answer Extraction module. The TEEM module is used to extract topic entities from natural language questions. The DSSM module is used to transform the natural language questions and predicates into semantic vectors in the same semantic space, so that the semantic similarities between questions and predicates can be measured by cosine similarities of the semantic vectors. The KB index is an index of knowledge base. The results of TEEM module are fed into the IR module and we retrieve the candidate knowledge triples from the KB index. Then, in the Answer Extraction module, we calculate the semantic matching scores of the candidate knowledge triples and select triples with the highest score as the final answers.

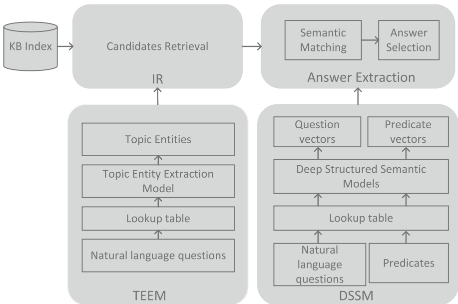


**Fig. 1.** The overview of the KBQA system framework.

### 3.1 Topic Entity Extraction Models

Topic entity for a question is the main entity which the question asks for. There may be many entities in a question. However we need extract the topic entity to retrieve the most relevant candidate triples. Topic entity extraction is an important step in the KBQA system, which can directly impact the results of candidate retrieval. Traditional QA systems extract topic entities by using linguistic tools, such as name entity recognition tools. These methods strongly rely on the quality of the linguistic tools, which are not always work well especially with Chinese. To extract high-quality topic entities in questions, we develop a Topic Entity Extraction Model (TEEM). The architecture of our TEEM model which is based on convolutional neutral network (CNN) is illustrated in Fig. 2. Each word in a question first is transformed into a word vector with k dimensions. Then a question of length n can be represented as a sequence of word vectors. In our experiment, the question length n is set to 20 (any tokens less than this range will be padded and out of this range will be discarded). We define the question as $q = (x_1, x_2, \cdots, x_n)$, where $x_i \in R^k$ is corresponding to the i-th word in the question. The convolution operation can be view as a sliding window which can extract local features in a question sentence. Whether a word is part of a topic entity is depended on the contextual information of the word. In the convolutional layer, we use multiple filters to obtain multiple local contextual features for each of the words in the questions. The word vectors of words within a sliding window of 3 words are mapped into a new local contextual feature, as shown in Fig. 2. A max-pooling layer is followed to select the most important feature in a max-pooling window and filter out undesirable features. The convolutions and max-pooling in deeper layers are defined in a similar way. In the final max-pooling layer, it can reach a fixed length vector. A full connection layer is followed to obtain the final output vector of the model.

For each word in a question, if the word is part of the topic entity, we set 1 as the tag of this word, otherwise we set 0 as the tag. Thus, we can obtain the supervise label vector of each question which is described as $y = (y_1, y_2, \cdots, y_n)$, where $y_i$ is the label of the i-th word in the question. For instance, for the question "命运石之门是哪一种类型的游戏", the label of each word in the question is "命运/1 石/1 之门/1 是/0 哪/0 一/0 种/0 类型/0 的/0 游戏/0" and the label vector is (1,1,1,0,0,0,0,0,0,0).

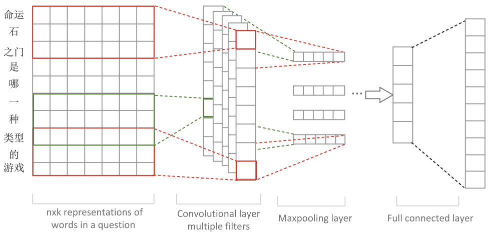


**Fig. 2.** TEEM architecture for an example question.

The output vector of our model is fixed length vector of n dimensions (n is the length of the input sequences). We denote the output vector as $z = (z_1, z_2, \cdots, z_n)$, where $z_i$ is corresponding to the tag of the i-th word in the input question. The goal of TEEM is to minimize the difference between the output vector $z$ and corresponding label vector. We use the mean squared error as the loss function of the TEEM, which is defined as

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

where $w$ is the weight parameters of the model, and $b$ is the bias parameters of the model. $\|w\|_2^2$ is the L2-regularization of the weight vectors which is added to combat overfitting. $\lambda$ is a hyper-parameter which controls the relative importance of the regularization parameter.

**Table 1.** Topic entities extracted by TEEM

<table>
<tr><th>Question</th><th>Topic Entity</th></tr>
<tr><td>3|-| 溴 | 化 | 苯甲 | 酮 | 的 | 熔点 | 是 | 多少 | 摄氏度 |?</td><td>3-溴化苯甲酮</td></tr>
<tr><td>金鱼 | 素 | 馅 | 包 | 的 | 辅料 | 都 | 有 | 什么 |?</td><td>金鱼素馅包</td></tr>
<tr><td>上海 | 逸 | 凡 | 居 | 旅店 | 有 | 哪些 | 休闲 | 设施 |?</td><td>上海逸凡居旅店</td></tr>
<tr><td>中 | 虹 | 花园 | 新都 | 苑 | 何时 | 竣工 | 的 |?</td><td>中虹花园新都苑</td></tr>
<tr><td>中国 | 电子 |m|5| 支持 | 电子书 | 功能 | 吗 |?</td><td>中国电子 m5</td></tr>
</table>

Through the TEEM module, we can extract topic entities in most questions. These topic entities can be used to retrieve the candidate knowledge triples and can significantly improve the quality of the retrieval results. Some topic entities extracted by the TEEM are shown in Table 1. The words in questions are separated by "|".

### 3.2 Deep Structured Semantic Models

The Deep Structured Semantic Model (DSSM) is used to measure the semantic similarity between the questions and the predicates in knowledge base. The predicate words mentioned in questions are always different from those defined in the knowledge base, which makes it very difficult to calculate the similarity between questions and predicates in knowledge base and may cause the ontology matching problem. This problem can be addressed using the recently proposed DSSM [6] and the improved methods Convolutional DSSM (C-DSSM) [8] and Long-Short-Term Memory DSSM (LSTM-DSSM) [19]. These semantic models map the sentences into k-dimensional vectors in a latent semantic space. So that their semantic similarity can be computed using some distance functions, such as cosine.

In this work, we extend the DSSM by using a Bidirectional Long Short-Term Memory (BiLSTM) neural network and the model is called BiLSTM-DSSM. Long short-Term Memory (LSTM) [20] is one of the popular variations of Recurrent Neural Networks (RNN) which is widely used to deal with variable-length

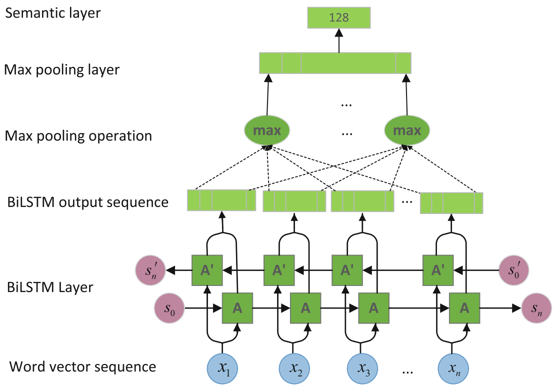


**Fig. 3.** Architecture of BiLSTM-DSSM.

sequence input. LSTMs are capable of learning long-term dependencies and remembering information for long periods of time. However, single directional LSTMs still suffer a weakness of not utilizing the contextual information from the future tokens. To handle this problem, we use the bidirectional LSTM, which can utilize both the previous and future context by processing the sequence on two directions. The architecture of the BiLSTM-DSSM is illustrated in Fig. 3. Unlike the previous semantic model, we use word vectors to represent the questions instead of the word hashing technique proposed in [6]. At the BiLSTM layer, $A$ and $A'$ are chunks of neural network which can look at an input vector and output a value. The input sequence is processed in the forward direction and the reverse direction. At each time step, we concatenate the two output vectors from both directions to obtain the final output vector of the BiLSTM Layer. Followed the BiLSTM Layer, we apply a max pooling layer to extract the most salient features and form a global feature vector with a fixed length. Then a feed-forward semantic layer is used to extract a high-level semantic feature vector for the input word sequence.

In order to obtain more composite representation of questions and predicates, we also develop a variant of the BiLSTM-DSSM model by integrate the CNN structures on the top of BiLSTM layer, which is called BiLSTM-CNN-DSSM, as shown in Fig. 4. The architecture the BiLSTM layer is similar to that in Fig. 3. We use the cosine similarity to measure the semantic relevance score between question and each predicate in knowledge base. Formally, the semantic relevance score between a question q and a predicate p is measured as:

$$R(q,p) = cosine(y_q, y_p) = \frac{y_q^T y_p}{\|y_q\| \|y_p\|} \tag{2}$$

where $y_q$ and $y_p$ are the semantic vectors of the question and the predicate, respectively.

Following [6], we first compute the posterior probability of a predicate given a question from the semantic relevance score between them through a softmax function:

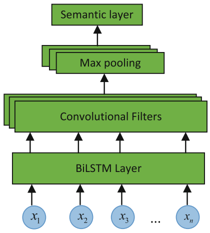


**Fig. 4.** BiLSTM-DSSM with CNN structures (BiLSTM-CNN-DSSM).

$$P(p|q) = \frac{\exp(\lambda R(q,p))}{\sum_{p' \in \mathbf{P}} \exp(\lambda R(q,p'))} \quad (3)$$

where $\lambda$ is a smoothing factor in the softmax function. $\mathbf{P}$ denotes the set of candidate predicates to be ranked, including a positive predicate and several negative predicates. The goal is to maximize the likelihood of the positive predicate of the given question. Therefore, we need to minimize the following loss function:

$$L(\Lambda) = -\log \prod_{r}^{R} P(p_r^+|q_r) \quad (4)$$

where $\Lambda$ is the parameters of the neural networks, $p_r^+$ is the positive predicate of the r-th question out of $R$ questions and $P(p_r^+|q_r)$ is the probability of the positive predicate given the r-th question.

### 3.3 Candidates Retrieval

Given a question, the main goal of the retrieval model is to find the most relevant knowledge triples in the knowledge base. First of all, we use the trained TEEM to extract the topic entity in the question. Then the topic entity is used to retrieve the relevant knowledge triples from the knowledge base as candidates. For example, for the question "上海逸凡居旅店有哪些休闲设施?" , the topic entity extracted by TEEM is "上海逸凡居旅店" and we can get the candidate knowledge triples by retrieve the knowledge base as shown in Table 2.

**Table 2.** The candidate knowledge triples for topic entity "上海逸凡居旅店"

| Subject | Predicate | Object |
|---|---|---|
| 上海逸凡居旅店 | 休闲设施 | 咖啡厅、理发美容室、棋牌麻将 |
| 上海逸凡居旅店 | 餐饮设施 | 中餐厅 |
| 上海逸凡居旅店 | 地址 | 上海市虹口区场中路 787 号 |
| 上海逸凡居旅店 | 别名 | 上海逸凡居旅店 |

### 3.4 Answer Selection

Once we obtain the candidate knowledge triples, the DSSM models can be used to map the question and candidate predicates into semantic vectors and then we can calculate the semantic relevant score between them. We use a question DSSM to encode questions and a predicate DSSM to encode predicates. To get better semantic matching score, we combine the semantic relevant scores of BiLSTM-DSSM, BiLSTM-CNN-DSSM and C-DSSM. And the final semantic relevant score is denoted as:

$$S(q,p) = \alpha R_b(q,p) + \beta R_{bc}(q,p) + \lambda R_c(q,p) \tag{5}$$

where $R_b(q,p)$, $R_{bc}(q,p)$ and $R_c(q,p)$ are the semantic relevance scores of the BiLSTM-DSSM, BiLSTM-CNN-DSSM and C-DSSM, respectively. And $\alpha$, $\beta$, $\lambda$ are the coefficient parameters of the three scores.

In order to select the best matching answers for a question, we also consider the lexical similarity between a question and a predicate at the character level. For each question-predicate pair, we segment the question and predicate into characters. The question vector $c_q$ and the predicate vector $c_p$ are constructed based on the characters. Each character is a dimension of the vector. If the character is appeared in the string the corresponding dimension is set to 1, otherwise set to 0. We use the cosine similarity to measure the lexical matching score:

$$LS(q,p) = \cos(c_q, c_p) = \frac{c_q^T c_p}{\|c_q\| \|c_p\|} \tag{6}$$

The final matching score can be defined as following:

$$score = \mu S(q,p) + \omega LS(q,p) \tag{7}$$

where $S(q,p)$ is the semantic relevant score, $LS(q,p)$ is the lexical matching score, and $\mu$, $\omega$ are the coefficient parameters of $S(q,p)$ and $LS(q,p)$ respectively.

In this paper, we rank the candidates according to the score defined in Eq. (7) and select the most relevant candidates as the final answer.

## 4 Experiment

### 4.1 Data Set

In this paper, we use the data set released by NLPCC-ICCPOL 2016 KBQA task. The data set includes 14,609 question-answer pairs and a knowledge base called nlpcc-iccpol-2016.kbqa.kb which contains 43M knowledge triples. The format of the triples in knowledge base nlpcc-iccpol-2016.kbqa.kb is: Subject ||| Predicate ||| Object.

In this paper, we need the question-entity pairs to train the TEEM and question-predicate pairs to train DSSMs. However, the given data set only contains question-answer pairs. To obtain better training dataset, we use an iterative approach to retrieve the knowledge triples from the knowledge base for training. Firstly, the question-answer pairs are used to search the corresponding knowledge triples from the knowledge base. The results are used as the initial training data to train the topic entity extraction model. Since there are plenty of noise entities in a question, the initial training data is always inaccurate. In order to get more high-quality training data, we use the entities extracted by the trained TEEM as additional information and retrieve the knowledge triples again. At last, we obtain 14165 question-entity pairs and 14165 question-predicate pairs. The negative examples for the DSSMs are randomly sampled from the training data.

### 4.2 Setup

The word vectors in this work is trained by word2vec [4], and the word vector size is 200. We use a zero vector to represent the word which is out of vocabulary. We use Stochastic Gradient Descent (SGD) to optimize the objective functions. The window size of the CNN layer in TEEM is 3. The dimension of the DSSMs output vector is set to 128. The smoothing factor $\lambda$ in Eq. (3) is set to 5. We train our models in mini-batches and the batch size is set to 10. We randomly sampled 5 negative samples for each question-predicate pair. The maximum length of word sequence for questions and predicates are set to 20 and 5 respectively. Any tokens less than this range will be padded and out of this range will be discarded, so that the length of the samples within a mini-batch can have the same length.

### 4.3 Evaluation Metric

The quality of the KBQA system is evaluated by Average$F_1$, which is defined as:

$$\text{Average}F_1 = \frac{1}{|Q|}\sum_{i=1}^{|Q|} F_i \tag{8}$$

$F_i$ denotes the $F_1$ score for question $Q_i$ computed based on $C_i$ and $A_i$. $F_i$ is set to 0 if the generated answer set $C_i$ for $Q_i$ is empty or doesn't overlap with the golden answers $A_i$ for $Q_i$. Otherwise, $F_i$ is computed as follows:

$$F_i = \frac{2 \cdot \frac{\#(C_i,A_i)}{|C_i|} \cdot \frac{\#(C_i,A_i)}{|A_i|}}{\frac{\#(C_i,A_i)}{|C_i|} + \frac{\#(C_i,A_i)}{|A_i|}} \tag{9}$$

where $\#(C_i, A_i)$ denotes the number of answers occur in both $C_i$ and $A_i$. $|C_i|$ and $|A_i|$ denote the number of answers in $C_i$ and $A_i$ respectively.

### 4.4 Experimental Results

In this section, we analyze the experimental results of our experiments. As introduced in Sect. 3, there are many parameters to be adjusted in the answer selection module. We can adjust the coefficient parameters in Eqs. (5) and (7) to get better results. Table 3 summarizes some results of different semantic matching methods in the case that the $\omega$ parameter in Eq. (7) is set to 0.5 and $\mu$ is set to 5. The Combined-DSSM uses Eq. (5) to combine the semantic relevant scores of C-DSSM, BiLSTM-DSSM and BiLSTM-CNN-DSSM. And the parameters in Eq. (5) are $\alpha = 0.4$, $\beta = 0.1$ and $\lambda = 0.5$. Some of the results of our experiment are shown in Table 3. The baseline system which is released by NLPCC-ICCPOL 2016 KBQA task is based on C-DSSM without using TEEM. As is shown in Table 3, the result of the Combined-DSSM, which is the submitted result for NLPCC-ICCPOL 2016 KBQA task, is much better than other methods. And all the models proposed in this paper substantially outperform the baseline system, which indicate that the proposed TEEM can significantly improve the results.

**Table 3.** The experimental results of different semantic matching methods

| Models | Average$F_1$ |
|---|---|
| Baseline system(C-DSSM) | 0.5247 |
| TEEM+C-DSSM | 0.7808 |
| TEEM+BiLSTM-DSSM | 0.7529 |
| TEEM+BiLSTM-CNN-DSSM | 0.7815 |
| **TEEM+Combined-DSSM** | **0.7957** |

Figure 5 shows the Average$F_1$-$\omega$ curves of the four sematic matching methods in the paper, which indicate the impact of the lexical matching score on the results. Here, $\omega$ is the parameter in Eq. (7). As is illustrated in Fig. 5, the Average$F_1$ can be improved by adjusting the parameter $\omega$ appropriately. The best result of our proposed methods can achieve an Average$F_1$ of 81.77% when the $\omega$ is set to 5.

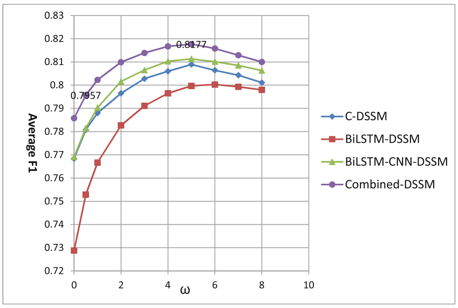


**Fig. 5.** The Average$F_1$-$\omega$ curves of the four sematic matching methods in the paper.

## 5 Conclusion

In this paper, we present a KBQA system framework to address the question answering problem. In our KBQA system, we propose a topic entity extraction model to extract the topic entities in questions at the candidate retrieval stage. And we apply deep structured semantic models to calculate the semantic similarity between questions and predicates at the answer selection stage. We extend the DSSM by using bidirectional long short-term memory and integrate a CNN structure on the top of the BiLSTM layer. The experimental results demonstrate that our system achieve good performances and substantially outperform the baseline system.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (No. 61303180 and No. 61573163), the Fundamental Research Funds for the Central Universities (No. CCNU15ZD003 and No. CCNU16A02024), and also supported by Wuhan Youth Science and technology plan. We thank the anonymous reviewers for their insightful comments.

## References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the SIGMOD 2008, pp. 1247–1250 (2008)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC - 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_52
3. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, vol. 14, pp. 1532–43 (2014)
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
6. Huang, P. S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 2333–2338. ACM, October 2013
7. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 373–374. ACM, April 2014
8. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pp. 101–110. ACM, November 2014
9. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

10. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
11. Zettlemoyer, L.S., Collins, M.: Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. arXiv preprint arXiv:1207.1420 (2012)
12. Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., Steedman, M.: Inducing probabilistic CCG grammars from logical form with higher-order unification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1223–1233, October 2010
13. Krishnamurthy, J., Mitchell, T.M.: Weakly supervised training of semantic parsers. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 754–765, July 2012
14. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: EMNLP, vol. 2, no. 5, p. 6, October 2013
15. Yih, W.T., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation question answering with knowledge base. In: Association for Computational Linguistics (ACL)
16. Yih, W.T., He, X., Meek, C.: Semantic parsing for single-relation question answering. In: ACL, vol. 2, pp. 643–648, June 2014
17. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of Association for Computational Linguistics, pp. 260–269 (2015)
18. Zhang, Y., Liu, K., He, S., Ji, G., Liu, Z., Wu, H., Zhao, J.: Question answering over knowledge base with neural attention combining global knowledge information. arXiv preprint arXiv:1606.00979 (2016)
19. Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Ward, R.: Semantic modelling with long-short-term memory for information retrieval. arXiv preprint arXiv:1412.6629 (2014)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)