BB-KBQA: BERT-Based Knowledge Base Question Answering

Aiting Liu(✉), Ziqi Huang, Hengtong Lu, Xiaojie Wang, and Caixia Yuan

Beijing University of Posts and Telecommunications, Beijing, China
{aitingliu,huangziqi,luhengtong,xjwang,yuancx}@bupt.edu.cn

Abstract. Knowledge base question answering aims to answer natural language questions by querying external knowledge base, which has been widely applied to many real-world systems. Most existing methods are template-based or training BiLSTMs or CNNs on the task-specific dataset. However, the hand-crafted templates are time-consuming to design as well as highly formalist without generalization ability. At the same time, BiLSTMs and CNNs require large-scale training data which is unpractical in most cases. To solve these problems, we utilize the prevailing pre-trained BERT model which leverages prior linguistic knowledge to obtain deep contextualized representations. Experimental results demonstrate that our model can achieve the state-of-the-art performance on the NLPCC- ICCPOL 2016 KBQA dataset, with an 84.12% averaged F1 score(1.65% absolute improvement).

Keywords: Chinese knowledge base question answering · Entity linking · Predicate mapping · BERT

1 Introduction

Recently, open domain knowledge base question answering (KBQA) has emerged as large-scale knowledge bases develop rapidly, such as DBpedia, Freebase, Yago2 and NLPCC Chinese Knowledge Base [1,2]. The goal of knowledge base question answering is to generate a related answer given a natural language question, which is challenging since it requires a high level of semantic understanding of questions. The mainstream methods can be divided into two paradigms. One line of research is built on semantic parsing-based methods [3–5] and the other utilizes information extraction-based methods [6–8]. In more detail, the former first converts the natural language question into a structured representation, such as logical forms or SPARQL [9,10], then query the knowledge base to obtain the answer. The latter, which is information extraction-based, first retrieves a set of candidate triples and then extracts features to rank these candidates. In this paper, we focus on the semantic-parsing method since it is more popular and general.

M. Sun et al. (Eds.): CCL 2019, LNAI 11856, pp. 81–92, 2019.
https://doi.org/10.1007/978-3-030-32381-3_7

In semantic parsing-based methods, the basic framework of KBQA [11–17] consists of three modules. The first one is ***entity linking***, which recognizes all entity mentions in a question (***mention detection***) and links each mention to an entity in KB (***entity disambiguation***). Normally, there are several candidate entities of a single mention, so entity disambiguation is needed. The second one is ***predicate mapping***, which finds candidate predicates in KB for the question. The last one is ***answer selection***, which ranks the candidate entity-predicate pairs, converts the top one into a query statement and queries the knowledge base to obtain the answer. For example, it first detects the mention "天堂鸟 ||*Bird of Paradise* " in the question "天堂鸟是什么界的动物呀? ||*Which kingdom does the animal Bird of Paradise belong to?*", then a candidate entity set {天堂鸟 (2001 年李幼斌主演电视剧) ||*Bird of Paradise(Teleplay starring Li Youbin in 2001)*, 天堂鸟 (迷你专辑) ||*Bird of Paradise(ep)*, 天堂鸟 (动物) ||*Bird of Paradise(animal)*, ...} and candidate predicate set {别名 ||*Alias*, 中文学名 ||*Chinese scientific name*, 界 ||*Kingdom*, 门 ||*Phylum*, 亚门 *Subphylum*, 纲 ||*Class*, 亚属 ||*Subgenus*, 种 ||*Species*} are obtained from the knowledge base. Finally, it ranks the candidate entity-predicate pairs and selects the top one "天堂鸟 (动物)-界 ||*Bird of Paradise(animal)-Kingdom*" to retrieve the factoid triple "< 天堂鸟 (动物)，界，动物界 > ||<*Bird of Paradise(animal), Kingdom, Kingdom Animalia*>" from the knowledge base, therefore the answer is "动物界 ||*Kingdom Animalia*".

In previous studies of entity linking module, Xie et al. [11] regards mention detection as a sequence labeling task with CNN model. Lai et al. [12], Yang et al. [13] and Zhou et al. [14] find all possible candidate entities of a question according to a pre-constructed alias dictionary. To disambiguate the candidate entities, Lai et al. [12] proposes a template-based algorithm which requires considerable hand-crafted templates, Yang et al. [13] utilizes the GBDT model and Zhou et al. [14] adopts a language model. In terms of predicate mapping module, Wang et al. [15] proposes the CGRU model and Yang et al. [13] combines the NBSVM model with CNN to rank candidate entities. Lai et al. [12] measures the token-level similarity between the question and each candidate predicate through a variety of hand-crafted extraction rules and gets the correct predicate. Xie et al. [11] introduces the CNN-DSSM [18] and BiLSTM-DSSM [19] which are variants of the deep semantic matching model (DSSM) [20] to calculate the semantic similarity between the question and each candidate predicate. However, above methods have two drawbacks: On the one hand, although prior linguistic knowledge can be combined directly into hand-crafted templates, the design of templates is time consuming. Meanwhile, hand-crafted templates are often with large granularity which prone to cause exceptions, which damage the generalization ability of models. On the other hand, the performances of BiLSTMs and CNNs are heavily dependent on large scale of training data which is often not available in practice. Recent years, pre-training [21–24] on large-scale unsupervised corpus, which is easy to collect, has shown its advantages on mining prior linguistic knowledge automatically, it indicates a possible way to deal with above two problems.

This paper focuses on exploiting pre-trained language models to ease the problems described above. BERT [23] is effectively combined into the semantic parsing-based framework for KBQA. Two different combining models for different subtasks in KBQA are designed. A BERT-CRF (Conditional Random Field [25]) model is proposed for mention detection, while a BERT-Softmax model is proposed for entity disambiguation and predicate mapping. In the end, we build a **B**ERT-**B**ased **KBQA** model, which achieves the state-of-the-art performance on the NLPCC-ICCPOL 2016 KBQA dataset with averaged F1 score of 84.12%.

Our contributions can be summarized as follows:

1. We propose the BERT-CRF model which integrates both advantages of BERT and CRF to train a efficient mention detection model. Furthermore, BB-KBQA model based on BERT-CRF and BERT-Softmax is proposed which leverages external knowledge and produces deep semantic representations of questions, entities and predicates.
2. Experimental results show that our method can achieve the state-of-the-art on NLPCC-ICCPOL 2016 KBQA dataset. Credit to the powerful feature extraction ability of BERT, our approach can produce more precise and related answer given the question.

2 BB-KBQA Model

As shown in Fig. 1, the KBQA framework consists of three modules: ***entity linking*** (including ***mention detection*** and ***entity disambiguation***), ***predicate mapping*** and ***answer selection***.

We propose a **B**ERT-**B**ased **KBQA** model based on this framework, where BERT-CRF is adopted for mention detection, and BERT-Softmax is adopted for entity disambiguation and predicate mapping. In the following, we first elaborate on the BERT-based models we design and then introduce these models in KBQA modules.

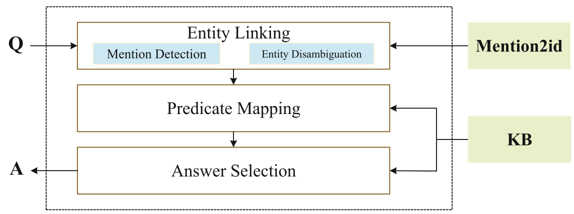

Fig. 1. KBQA framework.

2.1 Models

BERT. BERT is a multi-layer bidirectional Transformer [26] encoder. The input is a character-level token sequence, which is able to unambiguously represent either a single sentence or a pair of sentences separated with a special token [SEP]. For each token of the input sequence, the input representation is a sum of the corresponding token embedding, segment embedding and position embedding. The first token of every sequence is always the special classification symbol ([CLS]), and the final hidden state corresponding to this token can be used for classification tasks. BERT is pre-trained by two unsupervised prediction tasks: masked language model task and next sentence prediction task. After fine-tuning, the pre-trained BERT representations can be used in a wide range of natural language processing tasks. Readers can refer to [23] for more details.

BERT-Softmax. As shown in Fig. 2(a), following [23] fine-tuning procedure, the input sequence of BERT is $\mathbf{x} = \{x_1, \cdots, x_N\}$, and the final hidden state sequence is $\mathbf{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_N\}$,

$$\mathbf{H} = BERT(\mathbf{x}), \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{d\times N}$, $\mathbf{h}_i \in \mathbb{R}^d$ and $BERT(\cdot)$ denotes the network defined in [23]. Each hidden state $\mathbf{h}_i$ is followed by a softmax classification layer which outputs the label probability distribution $\mathbf{p}_i$,

$$\mathbf{p}_i = softmax(\mathbf{W}\mathbf{h}_i + \mathbf{b}), \tag{2}$$

here we view BERT-Softmax as a binary sequence classification task, where $\mathbf{W} \in \mathbb{R}^{2\times d}$, $\mathbf{b} \in \mathbb{R}^2$,$\mathbf{p}_i = \begin{bmatrix} \mathbf{p}_i{}^{(0)} \\ \mathbf{p}_i{}^{(1)} \end{bmatrix} \in \mathbb{R}^2$.

For sequence classification task, we only use the final hidden state of the first token (special symbol [CLS]) for softmax classification, which is employed in mention detection and predicate mapping modules; for sequence labeling task, there is a classifier on each hidden state $\mathbf{h}_i$, which is used in mention detection module.

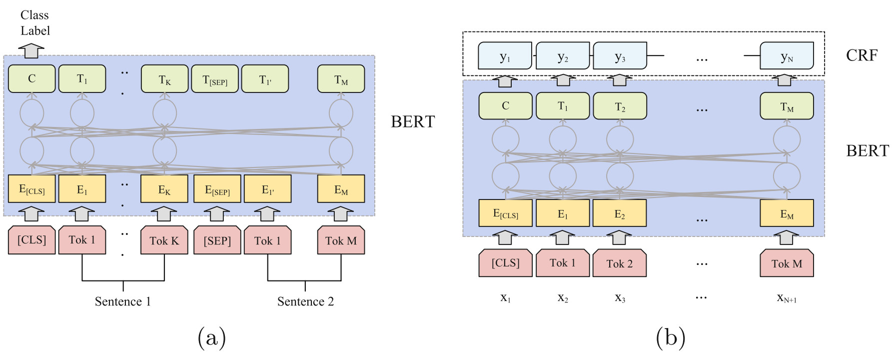

Fig. 2. BERT-Softmax model and BERT-CRF model.

BERT-CRF. The model structure is depicted in Fig. 2(b). In the sequence labeling task, the input sequence of the BERT is $\mathbf{x} = \{x_1, \cdots, x_N\}$, and the final hidden state sequence is defined the same as Eq. (1), which is further passed through a CRF [25] layer, then the final output is the predicted labels $\mathbf{Y}$ corresponding to each token,

$$\mathbf{Y} = CRF(\mathbf{WH} + \mathbf{b}), \tag{3}$$

where the label set is { "B" , "I" , "O" }, $W \in \mathbb{R}^{3\times d}$, $b \in \mathbb{R}^3$, $Y = \{y_1, \ldots, y_N\}$, $y_i \in \{0, 1, 2\}$, $i = 1, \ldots, N$. By employing a CRF layer, we can use past and future labels to predict the current label [27], leading to a state transition matrix of CRF layer, which focuses on sentence level instead of individual positions. Generally speaking, it can obtain higher labeling accuracy with the help of the CRF layer [27].

2.2 Modules

This paper builds each module in Fig. 1 based on the above models.

Entity Linking. Entity linking includes mention detection and entity disambiguation, where the former extracts the mention in a question, and the latter links the mention to its corresponding entity in the knowledge base.

Mention Detection. We treat mention detection as a sequence labeling task, where the BIO format is applied for representing mention labels. We construct a BERT-CRF model with question Q as input sequence $[Q]$[1] to detect the mention m in the question Q,

$$m = BERT\_CRF([Q]). \tag{4}$$

Entity Disambiguation. The candidate entity set $E = \{e_1, \cdots, e_T\}$ is obtained by a mention2id library[2] using the mention m, T is the number of candidate entities. The entity disambiguation can be regarded as a binary sequence classification task. We concatenate the question Q and each candidate entity e_i as input sequence $[Q; e_i]$[3], and feed it into the BERT-Softmax model to output the classification probability distribution $\mathbf{p}_i^e$,

$$\mathbf{p}_i^e = BERT\_Softmax([Q; e_i]), \tag{5}$$

where $\mathbf{p}_i^e = \begin{bmatrix} \mathbf{p}_i^{e(0)} \\ \mathbf{p}_i^{e(1)} \end{bmatrix} \in \mathbb{R}^2$, $i = 1, \ldots, T$. The predicted probability of label "1" is considered as the score $\mathcal{S}^e$ of the candidate entity, $\mathcal{S}^e \in \mathbb{R}^T$,

$$\mathcal{S}^e = \left[\mathbf{p}_1^{e(1)} \ldots \mathbf{p}_T^{e(1)}\right]. \tag{6}$$

1 Insert special symbol [CLS] as the first token of Q. We omit [CLS] from the notation for brevity.

2 mention2id library "nlpcc-iccpol-2016.kbqa.kb.mention2id" is introduced in [2], which maps the mention to all possible entities.

3 Insert special symbol [CLS] as the first token of Q. Delimiter [SEP] are added between Q and e_i. We omit [CLS] and [SEP] from the notation for brevity. $[Q; r_{ij}]$ ditto.

Predicate Mapping. Following the entity linking module, we get the candidate predicate set $R_i = \{r_{i1}, \cdots, r_{iL}\}$ from the KB according to the candidate entity e_i, L is the number of candidate predicates. Predicate mapping module scores all candidate predicates according to the semantic similarity between the question and each candidate predicate. The question Q is concatenated with the candidate predicate r_{ij} to form an input sequence $[Q; r_{ij}]$. Similar to entity disambiguation, BERT-Softmax model is employed to produce the score $\mathcal{S}^r$ for candidate predicates,

$$\mathbf{p}_{ij}^r = BERT\_Softmax([Q; r_{ij}]), \tag{7}$$

$$\mathcal{S}^r = \left[\mathbf{p}_{ij}^{r\ (1)}\right]_{T \times L}, \tag{8}$$

where $\mathbf{p}_{ij}^r$ is the label probability distribution, $\mathbf{p}_{ij}^r = \begin{bmatrix} \mathbf{p}_{ij}^{r\ (0)} \\ \mathbf{p}_{ij}^{r\ (1)} \end{bmatrix} \in \mathbb{R}^2$,$i = 1, \ldots, T$, $j = 1, \ldots, L$,$\mathcal{S}^p \in \mathbb{R}^{T \times L}$.

Answer Selection. In answer selection module, we calculate the weighted sum of candidate entity score $\mathcal{S}^e$ and candidate predicate score $\mathcal{S}^r$ as the final score of the candidate "entity-predicate" pair $\mathcal{S}$,

$$\mathcal{S} = \alpha \times \mathcal{S}^e + (1 - \alpha) \times \mathcal{S}^r, \tag{9}$$

where α is a hyper-parameter, $\mathcal{S} \in \mathbb{R}^{T \times L}$. We select the entity-predicate pair with the highest score and query the knowledge base through the query statement to get the answer.

3 Experiments

3.1 Datasets

The NLPCC-ICCPOL 2016 KBQA task [2] provides a training set with 14609 QA pairs, a test set with 9870 QA pairs, a Chinese knowledge base containing approximately 43M triples, and a mention2id library[4] that maps the mention to all possible entities. Since the mention detection, entity linking and predicate mapping modules require respective dataset, we create these three datasets in our own way. Specifically, we obtain the "entity-predicate" pair for the question via the golden answer. For mention detection task, we label the mention in the question manually. For entity disambiguation task, we collect all entities corresponding to the correct mention, and mark the correct entity as a positive example, other entities as negative examples. For predicate mapping dataset, we collect all predicates corresponding to the correct entity from the KB, and mark the correct predicate as a positive example, other predicates as negative examples.

[4] Chinese knowledge base "nlpcc-iccpol-2016.kbqa.kb" is introduced in [2].

The provided Chinese KB includes triples crawled from web. Each triple is in the form: <Subject, Predicate, Object>, where 'Subject' denotes a subject entity, 'Predicate' denotes a relation, and 'Object' denotes an object entity. There are about 43M triples in this knowledge base, in which about 6M subjects, 0.6M predicates and 16M objects are mentioned. On average, each subject entity corresponds to 7 triples, and each predicate corresponds to 73 triples. Some examples of triples are shown in Table 1.

Table 1. Triples in knowledge base.

Subject	Predicate	Object
北京 ||*Beijing*	别名 ||*Alias*	北京 ||*Beijing*
北京 ||*Beijing*	中文名 ||*Chinese name*	北京市 ||*Beijing City*
北京 ||*Beijing*	外文名 ||*Foreign name*	Municipality of Beijing
北京 ||*Beijing*	所属地区 ||*Region*	中国华北 ||*Northern China*
...	...	...

3.2 Training Details

We use Chinese BERT-Base model[5] pre-trained on Chinese Wikipedia corpus using character level tokenization, which has 12 layers, 768 hidden states, 12 heads and 110M parameters. For fine-tuning, all hyper-parameters are tuned on the development set. The maximum sequence length is set to 60 according to our dataset, the batch size is set to 32. We use Adam [28] for optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The dropout probability is 0.1. Typically, the initial learning rate is set to 1e-5 for BERT-CRF, 5e-5 for the BERT-Softmax, meanwhile a learning rate warmup strategy [23] is applied. The training epochs of BERT-CRF and BERT-Softmax are 30 and 3, respectively. Hyper-parameter α is set to 0.6 in answer selection module. For all baseline models, the word embedding is pre-trained by word2vec [29] using training set, and the embedding size is set to 300.

3.3 Compare with Baseline Models

We compare our model with the baseline model released in NLPCC-ICCPOL 2016 KBQA task [2], the state-of-the-art model [12] and some other baseline models [11,13–17]. Table 2 demonstrates the experimental results on the NLPCC-ICCPOL 2016 KBQA task. Our model outperforms all other methods. Compared with models using other sophisticated features and hand-craft rules (such as the use of part-of-speech features in the mention detection stage) [11,12], and models using simple LSTM, CNN [13–17], the BB-KBQA model we proposed achieves state-of-the-art result.

[5] https://github.com/google-research/bert.

Table 2. NLPCC-ICCPOL 2016 KBQA results (%)

Models	Averaged F1
Baseline model (C-DSSM)	52.47
Wang et al. [15]	79.14
Xie et al. [11]	79.57
Lei et al. [17]	80.97
Zhou et al. [14]	81.06
Yang et al. [13]	81.59
Xie et al. [16]	82.43
Lai et al. [12]	82.47
BB-KBQA	**84.12**

3.4 Module Analysis

Mention Detection. Table 3 summarizes the experimental results of our model and several baselines on mention detection task. BERT-Softmax is BERT model with a linear and softmax classification layer, BERT-BiLSTM-CRF combine BERT and BiLSTM-CRF model [27], BERT-CRF only adds a CRF layer based on BERT. Fine-tuned BERT-Softmax model has obvious improvement compared to traditional BiLSTM-CRF, where F1 score is relatively increased by 6.33%. BERT-BiLSTM-CRF is 0.29% higher than BERT-Softmax, and BERT-CRF which only employes a CRF layer get another 0.44% performance boost. The CRF layer can obtain the global optimal sequence labels instead of the local optimum, and the pre-trained BERT models the word order information and semantic information of the sequence. Adding a BiLSTM layer may disturb the valid information extracted by BERT.

Table 3. Mention detection results (%)

Models	F1
BiLSTM-CRF	90.28
BERT-Softmax	96.61
BERT-BiLSTM-CRF	96.90
BERT-CRF	**97.34**

Entity Disambiguation. It can be seen from Table 4 that the BERT-Softmax model outperforms all baseline models by approximately 2% on average, which shows that the fine-tuned BERT model can extract more comprehensive deep semantic information than other shallow neural network models, such as CNN and BiLSTM models.

Table 4. Entity disambiguation results (%)

Models	Accuracy@1	Accuracy@2	Accuracy@3
BiLSTM-DSSM [19]	85.89	88.50	90.81
Siamese BiLSTM [30]	87.85	92.58	94.59
Siamese CNN [31]	88.04	92.68	94.88
BERT-Softmax	**89.14**	**93.19**	**95.05**

Predicate Mapping. Table 5 demonstrates experimental results on predicate mapping task. The entity mention in the question may bring useful information as well as useless noise. Therefore, a set of comparative experiments are performed according to whether the entity mention in the question is replaced with a special token [ENT] (Siamese BiLSTM(2), Siamese CNN(2) and BERT-Softmax(2) represent models for such replacement operation). The experimental results show that this treatment is effective for the Siamese models, but does not work the same way in the BERT-Softmax model. While training the Siamese models, the entity mentions in the training set are sparse, resulting in insufficient training. However, the BERT model pre-trained with large-scale corpus covers a large amount of general knowledge, and the information of the mention in the question contributes to the predicate mapping task.

Table 5. Predicate mapping results (%)

Models	Accuracy@1	Accuracy@2	Accuracy@3
Siamese BiLSTM	92.54	96.74	98.12
Siamese BiLSTM(2)	93.74	97.46	98.38
Siamese CNN	86.47	93.80	96.16
Siamese CNN(2)	90.61	95.57	97.01
BERT-Softmax	**94.81**	**97.68**	**98.60**
BERT-Softmax(2)	94.66	97.63	98.41

3.5 Case Study

Table 6 gives some examples of our model and other baseline models. By modeling mention detection into a sequence labeling task instead of using hard matching method, our model can detect the mention even there are typos in the question. For example, the correct-written mention in the question "泡泡小兵中文版 的游戏目标是什么? ||*What is the goal of the Chinese version of Bubble Soldier?*" is "跑跑小兵中文版 ||*the Chinese version of Run Soldier*". Since there is no "泡泡小兵中文版 ||*the Chinese version of Bubble Soldier*" in the mention2id library, the hard matching method fails to detect it while our model works. By using BERT-CRF, we can detect the correct mentions that baseline models do wrongly. For the question "我要拼是什么国家的啊? ||*Which country is Wo Yao Pin?*", the detection of BERT-CRF is "我要拼 ||*Wo Yao Pin*" but the result of baseline models is "我 ||*Wo*". Similarly, BERT-Softmax is able to get the right result in some questions that are incorrectly resolved in the baseline models.

We also randomly sample some examples where our model does not generate correct answers. We find that some errors are caused by the dataset itself, which mainly includes: (1) there are unclarified entities of the question. For example, the mention "东山村 ||*Dongshan Village*" in the question "有人知道东山村的地理位置吗? ||*Does anyone know the location of Dongshan Village?*" has many corresponding entities in the knowledge base, such as "东山村 (云南省宜良县汤池镇东山村) ||*Dongshan Village (Dongshan Village, Tangchi Town, ziliang County, Yunnan Province)*", "东山村 (北京市门头沟军庄镇东山村) ||*Dongshan Village (Dongshan Village, Junzhuang Town, Mentougou, Beijing)*" and so on; (2) the question lacks an entity mention, like the question "我想知道官方语言是什么? ||*I want to know what the official language of the is?*" .

Table 6. Experiment result examples.

Question	[12]	[11]	BB-KBQA	False analysis
泡泡小兵中文版的游戏目标是什么? *What is the goal of the Chinese version of Bubble Soldier?*	×	√	√	Mentions are written-wrongly.
我要拼是什么国家的啊? *Which country is Wo Yao Pin?*	×	×	√	Mention detection fails.
告诉我《兄弟》这本书是几开的书? *Tell me the size of Brother.*	×	×	√	Entity Disambiguation fails.
沅水的流量有多少? *How much flow does the Yuanshui River have?*	×	×	√	Predicate mapping fails.
我想知道官方语言是什么 *I want to know what the official language of the is.*	×	×	×	Data error.
有人知道东山村的地理位置吗? *Does anyone know the location of Dongshan Village?*	×	×	×	Data error.

4 Conclusion

We propose a BERT-based knowledge base question answering model BB-KBQA. Compared to previous models, ours captures deep semantic information of questions, entities and predicates, which achieves a new state-of-the-art result of 84.12% on the NLPCC-ICCPOL 2016 KBQA dataset. In the future we plan to evaluate our model on other datasets and attempt to jointly model entity linking and predicate mapping to further improve the performance.

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 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
2. Duan, N.: Overview of the NLPCC-ICCPOL 2016 shared task: open domain Chinese question answering. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC-2016. LNCS (LNAI), vol. 10102, pp. 942–948. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_89
3. Wang, Y., Berant, J., Liang, P.: Building a semantic parser overnight. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1332–1342 (2015)
4. Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1470–1480 (2015)
5. Yang, M.-C., Duan, N., Zhou, M., Rim, H.-C.: Joint relational embeddings for knowledge-based question answering. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 645–650 (2014)
6. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks. arXiv preprint arXiv:1506.02075 (2015)
7. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 260–269 (2015)
8. Yih, W., Chang, M.-W., He, X., Gao, J.: Semantic parsing via staged query graph generation: question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1321–1331 (2015)
9. Berant, J., Chou, A., Frostig, R., Liang, P.: 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)
10. Berant, J., Liang, P.: Semantic parsing via paraphrasing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1415–1425 (2014)
11. Xie, Z., Zeng, Z., Zhou, G., He, T.: Knowledge base question answering based on deep learning models. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC-2016. LNCS (LNAI), vol. 10102, pp. 300–311. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_25
12. Lai, Y., Lin, Y., Chen, J., Feng, Y., Zhao, D.: Open domain question answering system based on knowledge base. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC-2016. LNCS (LNAI), vol. 10102, pp. 722–733. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_65
13. Yang, F., Gan, L., Li, A., Huang, D., Chou, X., Liu, H.: Combining deep learning with information retrieval for question answering. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC-2016. LNCS (LNAI), vol. 10102, pp. 917–925. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_86

14. Zhou, B., Sun, C., Lin, L., Liu, B.: LSTM based question answering for large scale knowledge base. Beijing Da Xue Xue Bao **54**(2), 286–292 (2018)
15. Wang, L., Zhang, Y., Liu, T.: A deep learning approach for question answering over knowledge base. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC-2016. LNCS (LNAI), vol. 10102, pp. 885–892. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_82
16. Xie, Z., Zeng, Z., Zhou, G., Wang, W.: Topic enhanced deep structured semantic models for knowledge base question answering. Sci. China Inf. Sci. **60**(11), 110103 (2017)
17. Lei, K., Deng, Y., Zhang, B., Shen, Y.: Open domain question answering with character-level deep learning models. In: 2017 10th International Symposium on Computational Intelligence and Design (ISCID), vol. 2, pp. 30–33. IEEE (2017)
18. 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 Conference on Information and Knowledge Management, pp. 101–110. ACM (2014)
19. Palangi, H., et al.: Semantic modelling with long-short-term memory for information retrieval. arXiv preprint arXiv:1412.6629 (2014)
20. 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 (2013)
21. Peters, M., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237 (2018)
22. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018). https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/languageunderstanding paper.pdf
23. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
24. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners
25. Lafferty, J., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data (2001)
26. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
27. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
28. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
30. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
31. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)