Open Domain Question Answering System Based on Knowledge Base

Yuxuan Lai[1], Yang Lin[1], Jiahao Chen[3], Yansong Feng[2(✉)], and Dongyan Zhao[2]

[1] School of Electronics Engineering and Computer Science, Peking University, Beijing, China
{erutan,linyang_}@pku.edu.cn
[2] Institute of Computer Science and Technology, Peking University, Beijing, China
{fengyansong,zhaody}@pku.edu.cn
[3] School of Mathematical Sciences, Peking University, Beijing, China
kaelchan@pku.edu.cn

Abstract. Aiming at the task of open domain question answering based on knowledge base in NLP&CC 2016, we propose a SPE (subject predicate extraction) algorithm which can automatically extract a subject-predicate pair from a simple question and translate it to a KB query. A novel method based on word vector similarity and predicate attention is used to score the candidate predicate after a simple topic entity linking method. Our approach achieved the F1-score of 82.47% on test data which obtained the first place in the contest of NLP&CC 2016 Shared Task 2 (KBQA sub-task). Furthermore, there are also a series of experiments and comprehensive error analysis which can show the properties and defects of the new data set.

Keywords: Chinese · Natural language question answering · Knowledge base · Information extraction

1 Introduction

Open-domain question answering (QA) is an important and yet challenging problem that remains largely unsolved. In recent years, a lot of works on QA in English have been published. But, to the best of our knowledge, the KBQA (question answering based on knowledge base) data set in NLP&CC 2016 evaluation task is the first large scale Chinese KBQA data set. In this paper we focus on answering single-relation factoid questions in Chinese, which is the main component of this data set. A SPE algorithm is proposed to translate a Chinese question to a KB querie. Logically, this algorithm can solve multiple-relation questions which can be expressed as a topic entity with a chain of predicates going from it. But limited by the data set, we did not carry out experiment about that.

Candidate predicate evaluation is the most important part of this algorithm, and a novel method based on word vector similarity and predicate attention

C.-Y. Lin et al. (Eds.): NLPCC-ICCPOL 2016, LNAI 10102, pp. 722–733, 2016.
DOI: 10.1007/978-3-319-50496-4_65

is applied. To a certain extent, this method looks like a neural network with attention mechanisms but it is so shallow that no parameters need to be trained except word vectors. As a result, it is more concise, efficient, interpretable and can be combined with prior knowledge more flexibly but lost some advantages of deep neural networks like the strong representational ability. There are also attempts to deal with knowledge base error caused by spider, question classification and training data to improve performance. Our approach achieved the F1-score of 82.47% on test data which obtained the first place in the evaluation task.

In the rest of the paper, we first review related work in Sect. 2, and in Sect. 3, we introduce the architecture of our method in detail. Experimental setup, results and implementation tricks are discussed in Sect. 4. We conclude the whole paper and look forward to the future research in Sect. 5.

2 Related Work

Open domain question answering is a perennial problem in the field of natural language processing, which is known as an AI-complete problem. Traditional method to solve this problem is basically based on information retrieval, such as the Mulder system [1] and the AskMSR system [2,3]. Meeting with the requirement of answering questions more directly and accurately, some knowledge bases were built to structure facts. Large-scale knowledge bases (KB) like DBPedia [4] and Freebase [5] have become important resources for supporting open-domain question answering. Most approaches to KBQA map a question to its semantic representation (e.g. first-order logical form) based on some kinds of parsing method such as: Semantic Parsing [6], dependency parsing [7], and CCG (Combinatory Categorial Gramma) [8]. But these works hardly use KBs to help with parsing and they are bounded by the accuracy of the parsing method which is particularly severe while dealing with Chinese. To avoid these disadvantages, some approaches extract KB queries from questions with the help of knowledge bases like the recent works on WebQuestions [9–11]. The Ye's system [12], which achieved the best performance in NLP&CC 2015 Chinese QA task, also had a SPE algorithm, but it was only used as a supplement to their web knowledge extraction algorithm.

3 Architecture

The architecture of our system is shown in Fig. 1. Several hand-written patterns are used to find out the core of each question (See Appendix A). The interrogative structures such as "我想知道 ‖ I want to know that" and "吗 ‖ (modal particle)" are abandoned. There are some rules to reduce errors which are obviously caused by spider such as unexpected special symbols and heritage of html tags in knowledge base (See Appendix B). For each question, we use a simple topic entity linking method to extract possible KB entities. All the predicates after these entities are evaluated based on word vector similarity and predicate

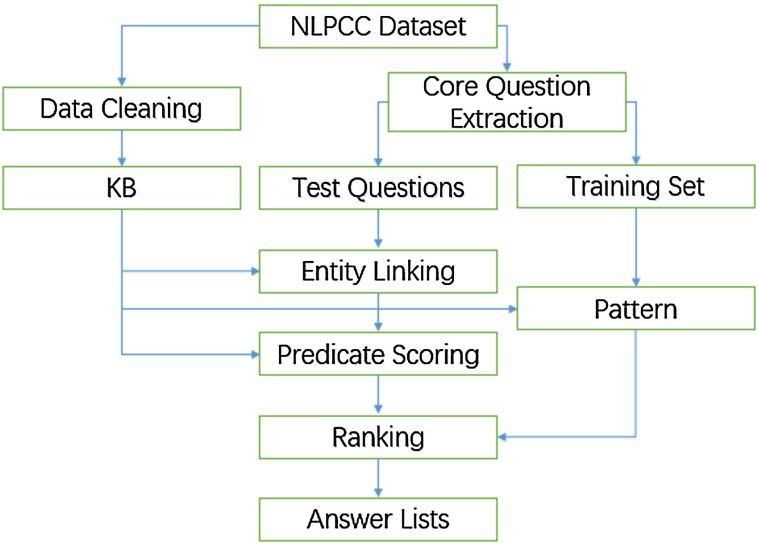

Fig. 1. Architecture of our KBQA system

attention. A linear combination of entity length and predicate score is used to rank the subject-predicate pairs as well as the answer patterns extracted from question-answer pairs in training set and a prior rule to deal with alternative questions.

3.1 Topic Entity Linking

Topic entities of questions are the core entities of the corresponding KB queries. In our system, all entities in KB which is a substring of a question and not overlapped by others are considered as potential topic entities. There is also a stop word list that consists of high frequency entities to reduce the noise entities (such as "是 ‖ is", "什么 ‖ what" and some auxiliary word) and improve efficiency. The length of entity is considered as a feature to use in the ranking stage.

For example, the potential topic entities of the question "做铁板鱼要用到哪些调料？ ‖ What sauce will be used when you make iron fish (Chinese cuisine)?" are "铁板鱼 ‖ iron fish", "到 ‖ to" and "调料 ‖ sauce". "做 ‖ make", "要 ‖ will", "用 ‖ use", 哪 and "些" ("哪些" means "what") are ignored for they are high frequency entities.

3.2 Predicate Scoring

After topic entity linking, there are some potential entities and every entity has a few predicates in KB. From our perspective, a good predicate can handle the semantic of the rest of the question perfectly. Then the score of a candidate predicate is:

$$S_p = \frac{\sum_i (lp_i * \max_j Sim(wp_i, wq_j))}{\sum_i lp_i} \quad (1)$$

Where wp_i is the i_{th} word in predicate, wq_j is the j_{th} word in the question. lp_i is the length of wp_i. Sim is the semantic similarity of the two words, here the cosine similarity of word vectors is adopted. This score will be used in ranking stage.

The semantic similarity is considered to represent how much the words in predicate care about words in question. If the similarity between p_i and q_{j1} is higher than that between p_i and q_{j2}, then q_{j1} involves more semantic of p_i. Since only one question word is considered in evaluation for each predicate word, the attention is similar to a weighted alignment procedure. The most concerned question words constitute the attention of the whole predicate. The weighted average of word similarities using the length of predicate words as the weight measures whether the predicate is suitable to this question.

To get words from predicates and questions, we build a Chinese word segmenter which print all possible words based on a large word list. For example, the sequence "使用人数 ‖ number of users" is separated to "使 ‖ make", "使用 ‖ use", "用 ‖ use", "用人 ‖ choose a person for a job", "人 ‖ person", "人数 ‖ number of person" and "数 ‖ number". Several segmentation tools such as hanLP and NLPIR are attempted but the statements are so casual that these tools bring more errors than benefits. The performance of omni-segmentation mode is better than normal mode but still worse than our segmenter obviously. Furthermore, auxiliary words and punctuation are ignored here since they are meaningless.

There are two reasons why this segmenter performs better. Firstly, in Chinese, if one word covers another word, they usually have similar semantic, especially in oral language. For example, "这个东西怎么使 ‖ How to use it", "这个东西怎么用 ‖ How to use it", and "这个东西怎么使用 ‖ How to use it". Secondly, there often exists a relation like "kind of" between covering words, e.g. "人数 ‖ number of person" and "数 ‖ number", which is much useful when sentence structure are changed. As a result, this segmenter can deal with high flexible lexical in oral language. There are also a few word ambiguities such as "用人 ‖ choose a person for a job" caused by this segmenter, some of them can be solved in the predicate attention part for the greatly difference on semantic. Anyway, some errors still remain but it is not so significant as the benefit.

Using word vector to calculate semantic similarity has trouble in dealing with interrogative words. The cosine similarity between "什么时候 ‖ when" and "日期 ‖ date", "在哪儿 ‖ where" and "地点 ‖ place" are much lower than expected. So some hand-written pattern are used to finger out question types and some symbol words are added to the segmenter result directly. Since the statements are very flexible, only 3 types of questions are processed (Table 1), which cover about 10% among all questions. It is just a small attempt to combine the predicate scoring with some prior rules, which has much room for improvement.

Table 1. Question type rules

Question Type	Question Symbol	Added Words
when	什么时候, 何时	时间, 日期
where	在哪	地点, 位置
how much money	多少钱	价格

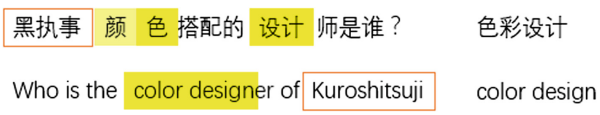

Fig. 2. Predicate attention example

An example of the predicate attention is shown in Fig. 2, and the cosine similarity and mapping relations are shown in Table 2.

Table 2. Cosine similarity and mapping relations in Fig. 1

Words in Predicate	Words in Question	Cosine Similarity
色	色	1.0
色彩	颜色	0.60
彩	色	0.42
设	设	1.0
设计	设计	1.0
计	计	1.0

3.3 Answer Pattern

Answer patterns are rules extracted from labeled question-answer pairs which can reveal the connection between problem statements and the KB structures corresponding to true answers. For each question-answer pair, candidate triples are SPO (subject-predicate-object) triples in KB whose subject is a substring of the question and object is the same as the answer. Several simple rules are used to filter high confidence triples (mainly based on the length of match). There are also a few cases that no candidate triple left and these question-answer pairs are ignored in order to ensure the high precision of best triples.

Generalizing the subject, answer patterns are extracted from question-answer pairs and best triples. How many times a answer pattern generated from a test question and a candidate subject-predicate pair occurs in training answer patterns is used as a feature in ranking stage. An example of answer pattern extraction is shown in Table 3.

Table 3. Answer pattern example

Question	做铁板鱼要用到哪些调料？ What sauce will be used when you make iron fish(Chinese cuisine)?
Answer	粗盐味精鸡精 coarse salt, Monosodium glutamate, chicken bouillon
Best Triple	铁板鱼,调味品,粗盐味精鸡精 iron fish, condiment, coarse salt Monosodium glutamate chicken bouillon
Answer Pattern	做(SUB)要用到哪些调料？-(SUB),调味品,(ANS) What sauce will be used when you make (SUB) - (SUB),condiment,(ANS)

3.4 Ranking

In ranking stage, a linear combination of features including entity length, predicate score and answer pattern occurrence is used to choose golden answers.

Some rules can be added here to deal with questions which are not suitable for this architecture. For example, alternative questions hardly involve the meaning of predicates and only have limited objects to choose. So rules make sure that the answers of these questions must occur between "是...还是... ‖ either ... or ...". But the statements of questions are so flexible that the improvement is limited.

4 Experiment

4.1 Dataset

The dataset is published by NLP&CC2016 evaluation task including a knowledge base and question-answer pairs for training and testing. There are about 43M SPO pairs in the knowledge, where about 6 M subjects, 0.6 M predicates and 16 M objects are mentioned. The training set contains 14,609 question-answer pairs and testing contains 9,870 question-answer pairs. The answers are labeled by human and most of them are objects in the KB (the rest are caused by human mistake or other unexpected reasons).

4.2 Experiment Settings

The entities which have occurred in both training and testing questions more than 150 times are considered as stop words during topic entity linking. There are 52,916 distinctive entities in 24,479 questions and only 496 high frequency entities, which is less than 1% of all.

We use word2vec software of Tomas Mikolov and his colleagues[1] to generate word vectors. The CBOW model [13–15] are used. The window size is 5, the desired vector dimensionality is 300 and threshold for downsampling the frequent words is 20. Sentences in Baidu baike are used as training data and 155,837 word vectors are generated, which is also the word list used in the segmenter.

[1] https://code.google.com/archive/p/word2vec.

The weights of the linear combination in ranking stage are assigned as follow: weight of entity length (an integer) is 1 (record as w_{el}), weight of predicate score (a float between 0 to 1) is 10 (record as w_{ps}), weight of answer pattern occurrence (an integer) is 100 (record as w_{po}). The confidence of answer patterns is naturally high to make sure the patterns will be followed as long as they matches. The combination is mainly a balance between the other two features. Let r be the rate of weights between predicate score and entity length. We tested the influence of r (fixing $w_{el} = 1$ and $w_{po} = 100$) alidation on training set. The results are shown in Fig. 3. The performance was robust when r is from 10 to 16, so we set $r = 10$, which means $w_{ps} = 10$.

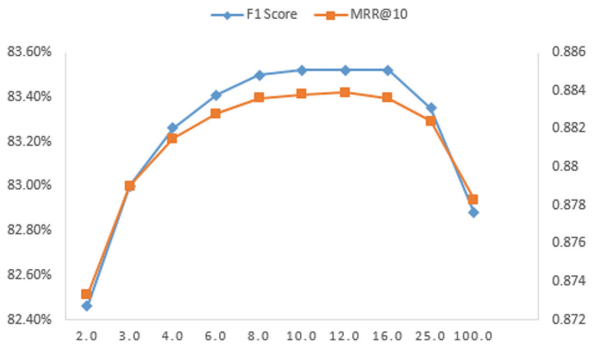

Fig. 3. Influence of tuning weight rate

4.3 Benchmark Systems

There are three benchmark systems. The first of them is provided by Duan Nan, which is basically based on the idea of two previous works [11,16] and is the official benchmark system of this KBQA task. So we just call it "NLPCC". The second one is called "PatternMatch", which only uses answer patterns extracted from training data to answer questions. Core question extraction is retained to improve hit rate. The precision of this benchmark system can be considered as the performance of the labeled answers in training data, which is also a sort of upper bound of this data set. The Third system is a naive SPE algorithm which just finds the longest subject-predicate pair in questions and the corresponding object is regarded as answers, the accuracy of which is the lower bound of all SPE algorithm. We call it "NaiveSearch".

4.4 Results

We compare our system with some benchmark systems, which is shown in Table 4. "Answered" means the rate of questions that the system can answer.

Table 4. Results compared with benchmarks

System	F1 Score	Pre@1	Pre@2	Pre@3	Pre@5	Pre@inf	Answered
NLPCC	52.48%	52.48%	60.46%	64.15%	67.33%	—	100%
PatternMatch	18.41%	88.59%	—	—	90.93%	90.93%	20.78%
NaiveSearch	47.39%	84.41%	—	—	88.19%	—	56.13%
PatternMatch+ NaiveSearch	52.99%	85.18%	—	—	—	—	62.21%
Ours	**82.47%**	82.47%	88.82%	91.17%	92.71%	95.08%	100%

So accuracy@N is equal to precision@N multiplied by the answered rate. "PatternMatch+NaiveSearch" is the combination of those two systems by adding the scores of candidate answers directly.

The F1 score of our system is much better than all baselines, which proves the effect of our approach. The accuracy@1 of our system is close to the precision@1 of PatternMatch and NaiveSearch system which are systems based on strong rules with high precision and poor recall, which means that our system is close to the theoretically best system on this data set.

We also test the performance of the core of our system and the influence of rule parts, which is shown in Table 5. Since all systems print ordered answers instead of parallel answers, the best strategy is to select the first answer, thus average F1 score is equal to accuracy@1. And because all of the "Answered" rates of these tests are 100%, we use accuracy@N to replace precision@N here.

Table 5. Contributions of each part

System	ACC@1	ACC@2	ACC@3	ACC@5	ACC@inf	MRR
Core	81.64%	88.37%	90.75%	92.33%	95.06%	0.8616
Core+Tr	81.77%	88.46%	90.82%	92.41%	95.06%	0.8627
Core+QCore	81.83%	88.47%	90.89%	92.46%	95.06%	0.8631
Core+Tr+QCore	82.16%	88.64%	91.07%	92.66%	95.08%	0.8657
Full-QClassify	82.18%	88.67%	91.09%	92.68%	95.08%	0.8660
Full-AQ	82.45%	88.79%	91.14%	92.68%	95.08%	0.8676
Full-Tr	82.33%	88.67%	90.99%	92.55%	95.06%	0.8663
Full	**82.47**%	88.82%	91.17%	92.71%	95.08%	0.8678

In Table 5, "Tr" means using answer patterns extracted from training data. "QCore" means using the core of questions instead of full questions. "QClassify" means adding question classification method to predicate scoring. "AQ" means posttreatment rules for alternative questions. "Full" equals to "Core + Tr + QCore + QClassify + AQ". The "Core" system is a unsupervised system using the basic SPE algorithm.

According to Table 5, the added methods improve the performance of the core system a little. There are two reasons why the influence is not significant. Firstly, all the added methods except "QCore" involve only few questions, and "QCore" is just a data cleaning method. Secondly, these rules improve some weakness which was found when making cross-validation in training data, but the distribution of questions in testing data is a little different, which makes the effect weaker. Details are shown in Table 6.

Table 6. Influence of each method

Method	Train-Influence		Test-Influence	
	Sphere	Improvement	Sphere	Improvement
AQ	0.21%	46.7%–63.3%	0.11%	72.7%–90.9%
Tr	22.35%	82.5%–84.4%	20.78%	88.4%–89.0%
QClassify	10.51%	71.0%–77.5%	8.74%	77.5%–80.9%

The results of NLP&CC2016 evaluation task are shown in Table 7 (the top 5 results of 21 submissions in total). Our system achieve the best performance in all teams. Compared with Table 5, even the core system is better than the second team in this evaluation task.

Table 7. Evaluation results in this evaluation task

Team	F1 score
Ours	**82.47**%
NUDT	81.59%
CCNU	79.57%
HIT-SCIR	79.14%
NEU (NLP Lab)	72.72%

4.5 Error Analysis

We randomly sampled 100 questions that our system did not generate the correct answer in order to analyze the room for improvement. Near half of errors are in fact due to label problems or question design which are not real mistakes. This includes unclarified entity in question (31%, e.g. "兴隆镇的邮编是多少‖what is the postcode number of Xinglong Town", there are a lot of Xinglong Towns in China), contradictory in KB (5%) and questions whose intent are unable to understand (2%) or with wrong labeled answer (5%). 30% of the errors are because of the unsuccessful entity linking or predicate scoring, which is the foremost part of the room for improvement. There are 27% of the errors are caused by questions whose answer is not an object of a subject in question, which can not be handled by our architecture. This will be discussed later in Sect. 4.6.

4.6 Dataset Analysis

Since this data set is the first large scale Chinese KBQA data set, we performed a series of experiments to show properties and defects, which can be useful to those who want to use this data set or to build another one.

This data set is a "simple-question" data set where every question can be answered by only one SPO pair in KB. For most questions, there is a SPO pair that the subject is a substring of the question and the object is equal to the answer. But there are 788 exceptions (3.22%), 343 in training set (2.35%) and 445 in testing set (4.51%). We randomly sampled 100 of them and find that the reasons why their answers can not be represented as objects of their subjects. Reasons include format problems (26%, e.g. "胡椒基氯的分子量是什么 ‖ What is the molecular weight of piperonyl chloride", the labeled answer is "170.6", while the answer in KB is "170.60"), wrong answers (11%, e.g. "钡d-3-三氟乙酰基樟脑酸的分子量是多少 ‖ what is the molecular weight of Barium D-3-trifluoroacetylcamphorate", the labeled answer is "9月22日 ‖ Sep 22nd"), typos in entities (29%), aliases of entities (14%), incomprehensible questions (2%). There are still 18% of them we cannot classify. The aliases of entities and some of the typos in entities can be solved in future works.

Some ambiguities are caused by entities with the same name and no clue in questions can help to distinguish them. There are 4773 such questions (19.50%), 3189 in training set (21.83%) and 1584 in testing set (16.05%). The original accuracy of our system on these questions is only 62.43%, 62.40% in training set and 62.50% in testing set. If the accuracy of these is judged by finding correct subject-predicate pair, the accuracy of our system on these questions in testing set is 82.07%. So with this change, our system performance will be up to 85.61%.

5 Conclusion

In this paper, we present a KBQA system which can answer simple-relation Chinese questions base on a SPE algorithm. We use this system to participate in the contest of NLP&CC 2016 Shared Task 2 (KBQA sub-task) and obtained the first place. Since this data set is the first large scale Chinese KBQA data set, we perform a series of experiments and comprehensive error analysis which can be useful to those who want to use this data set or to build another Chinese KBQA data set. In the future, we would like to extend our system to answer multi-relation questions and try some deeper models to improve the performance.

Acknowledgement. We would like to thank members in our NLP group and the anonymous reviewers for their helpful feedback. This work was supported by National High Technology R&D Program of China (Grant No. 2015AA015403, 2014AA015102), Natural Science Foundation of China (Grant No. 61202233, 61272344, 61370055) and the joint project with IBM Research. Any correspondence please refer to Yansong Feng.

Appendix A

The 8 regular expressions shown in Table 8 are used to capture the non-core parts. They are executed in order.

Table 8. Regular expressions for core question extraction

(啊|呀|(你知道)?吗|呢)?(？|\?)*$
来着$
^呃(……)?
^请问(一下|你知道)?
^(那么|什么是|我想知道|我很好奇|有谁了解|问一下|请问你知道|谁能告诉我一下)
^((谁|(请|麻烦)?你|请|)?(能|可以)?告诉我)
^((我想(问|请教)一下)，?)
^((有人|谁|你|你们|有谁|大家)(记得|知道))

Appendix B

The rules to clean KB are shown in Table 9.

Table 9. KB cleaning rules

Type	Times	e.g.	Disposal
Appendix labels in predicate	9110	性质[1]	Correct
Predicate prefix "-"	77332	- 社区数	Correct
Predicate prefix "•"	85953	• 密度	Correct
Space in predicate between Chinese characters	367218	国 籍	Correct
Predicate is the same as object	193716	陈祝龄旧居||| 天津市文物保护单位	Delete

References

1. Kwok, C.C.T., Etzioni, O., Weld, D.S.: Scaling question answering to the Web. In: Proceedings of the 10th International Conference on World Wide Web (2001)
2. Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A.: Data-intensive question answering. In: Proceedings of TREC (2001)
3. Tsai, C.-T., Yih, W.-T., Burges, C.J.C.: Web-based question answering: revisiting AskMSR. Technical report MSR-TR-2015-20, Microsoft Research (2015)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_52

5. 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, SIGMOD 2008, pp. 1247–1250 (2008)
6. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of EMNLP (2013)
7. Liang, P., Jordan, M., Klein, D.: Learning dependency-based compositional semantics. In: Proceedings of ACL (2011)
8. Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., Steedman, M.: Lexical generalization in CCG grammar induction for semantic parsing. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2011)
9. Yao, X., Van Durme, B.: Information extraction over structured data: question answering with freebase. In: Proceedings of ACL (2014)
10. Yao, X., Berant, J., Van Durme, B.: Freebase QA: information extraction or semantic parsing? In: Proceedings of ACL (2014)
11. Yih, W.-T., Chang, M.-W., He, X., Gao, J.: Semantic parsing via staged query graph generation: question answering with knowledge base. In: Proceedings of ACL Association for Computational Linguistics (2015)
12. Ye, Z., Jia, Z., Yang, Y., Huang, J., Yin, H.: Research on open domain question answering system. In: Li, J., Ji, H., Zhao, D., Feng, Y. (eds.) NLPCC 2015. LNCS (LNAI), vol. 9362, pp. 527–540. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25207-0_49
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR (2013)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS (2013)
15. Mikolov, T., Yih, W.-T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of NAACL HLT (2013)
16. Junwei, B., Nan, D., Ming, Z., Tiejun, Z.: Knowledge-based question answering as machine translation. In: Proceedings of ACL (2014)