A Chinese Question Answering System for Single-Relation Factoid Questions

Yuxuan Lai*, Yanyan Jia, Yang Lin, Yansong Feng, and Dongyan Zhao

Institute of Computer Science & Technology, Peking University, Beijing, China,
{erutan, jiayanyan, linyang, fengyansong, zhaody_}@pku.edu.cn

Abstract. Aiming at the task of open domain question answering based on knowledge base in NLPCC 2017, we build a question answering system which can automatically find the promised entities and predicates for single-relation questions. After a features based entity linking component and a word vector based candidate predicates generation component, deep convolutional neual networks are used to rerank the entity-predicate pairs, and all intermediary scores are used to choose the final predicted answers. Our approach achieved the F1-score of 47.23% on test data which obtained the first place in the contest of NLPCC 2017 Shared Task 5(KBQA sub-task). Furthermore, there are also a series of experiments which can help other developers understand the contribution of every part of our system.

Keywords: Natural Language Question Answering, Knowledge Base, Information Extraction, Deep Convolutional Neual Network

1 Introduction

Open-domain question answering is an important and yet challenging problem that remains largely unsolved. In recent years, with the development of large-scale knowledge bases, such as DBPedia[12] and Freebase[13], many studies focus on generating precise and reliable answers for open-domain questions from knowledge bases. In this paper, we introduce a system that can answer single-relation factoid questions in Chinese, which is the main component of the NLPCC KBQA evaluation task. We proposed a novel method based on deep CNNs to rerank the entity-predicate pairs which generated by approaches based on shallow features. Our system achieved the F1-score of 47.23% on test data which obtained the first place in the evaluation task.

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

open domain questions over knowledge bases can generate more precise and reliable answers. Many traditional KBQA technologies are based on information retrieval[7][8] and semantic parsing[9][10][11]. Recently, some works use representation learning to determine similarity between entity mentions and knowledge base entities[1], question patterns and knowledge base predicates[1] or knowledge base subgraphs[2]. They proved that neural network approaches can handle high-level semantic similarity better. When dealing with complicated natural language tasks such as question answering, it is rewarding to combine neural networks with traditional shallow features[2][3][4]. Following their ideas, we also combine traditional shallow features with CNNs features in our system.

Deep convolution neural networks have emerged great power in field of computer vision. Recently, a few works try to use deep architectures in NLP tasks such as text classification[5] and machine translation[6]. They followed the design of VGG[14] and ResNet[15], using narrow filters and residual connections to reduce parameters and make the deep architecture easier to train. We also attempt to achieve a deep CNNs in our system but followed the GoogLeNet[16] architecture, using multi-perspective filters with residual connections.

NLPCC have organized Chinese KBQA evaluation task for three years. The Ye's system[18], which achieved the best performance in NLPCC 2015 Chinese KBQA task, combined a subject predicate extraction algorithm with web knowledge retrieval. Lai[19] used word vector based features to search best subject-predicate pair and achieved the best performance in NLPCC 2016 KBQA task. Yang[20] combined features based entity linking, Naive Bayes based answers selection, and CNNs based reranking and achieved the second place in 2016. Our system is mainly inspired by their works[19][20], but we achieved a novel CNN architecture and combined advantages of their system appropriately. We also ameliorate the word vector based predicates selection algorithm in [19] and our entity linking approach is slightly different from [20]. Furthermore, an exquisite generative adversarial like negative sampling approach are adopted to deal with the data unbalance of CNN training.

3 Architecture

The architecture of our system is shown in Figure 1. Enlightened from previous works [19], several hand-written rules are adopted to correct spider error such as unexpected special symbols in knowledge base and extract core expressions from questions. Then, a feature based approach is used to select promised entity mentions followed by an unsupervised word vector based predicates scoring method. After candidate entity-predicate pairs are generated, deep CNNs models are used to rerank them. All intermediary scores are used to choose the final predicted answers.

The rules used to pretreat NLPCC dataset are almost the same as the pervious work (See Appendix in [19]). But when dealing with the knowledge base, delete rules are ignored. If the core expression of a question is an entity, we will add the word "introduce" so that our system will attempt to give an introduc-

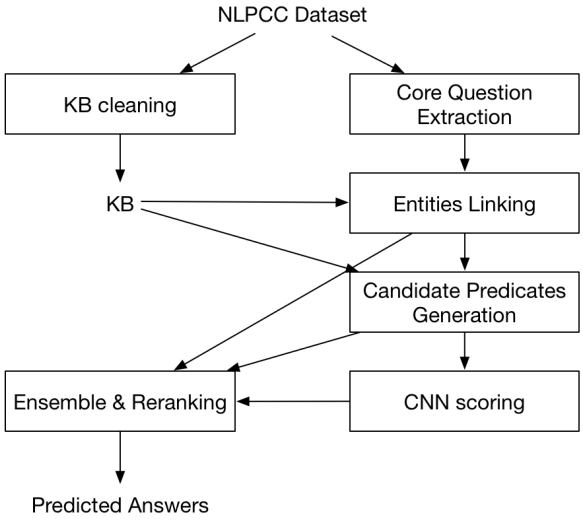

Fig. 1. Architecture of Our KBQA System

tion of this entity. Anyway, only 26 of the 7631 questions are influenced by this introduction trick.

3.1 Entity Linking

A KB entity "Li Na (Diving Athlete)" is consist of the entity name "Li Na" and the entity explanation "Diving Athlete" (sometimes absence). Topic entities of questions are the core entities of the corresponding KB queries and entity mentions are substrings of question which entails topic entities. An entity mention entails a topic entity if and only if the mention is the same as the topic entity, or just the name of, or the correspondence are mentioned in the provided file "nlpcc-iccpol-2016.kbqa.kb.mention2id". Enlightened from previous works [20], a features based GBDT(gradient boost decision tree) are trained to select promised entity mentions from all possible substrings of questions.

In order to train supervised entity linking models, golden mentions labeling is a prerequisite. A golden mention must entail a KB entity with an object same as the golden answer. To ensure the precision of the golden labeling, several rules considering coverage between mentions, mention lengths and positions are adopted and every question has at most one golden mention. The statistical results are demonstrated in Table 1. Inspected manually, most of the excluded mention candidates are defective.

All features adopted in entity linking model are demonstrated below, which is similar to the pervious work[20]. But no part-of-speech information is considered and most of the features have several perspectives. Since our mentions are substrings, not continuous words but Chinese characters, FMM (forward maximum

Table 1. Statistics of Golden Entity Labeling

Dataset	#All Questions	#Have Candidates	#Labeled Golden
16-train	14609	14323	14306
16-test	9870	9493	9482
17-test	7631	4833	4829

matching) is used to find the next word and RMM (reverse maximum matching) is used to find the last word. A GBDT model is trained on questions which have gloden mention based on these features. Settings and results are shown in Section 4.

- **Position and Length.** The absolute and relative position of the head, the middle, and the tail of the mention. The absolute and relative length of the mention. Whether the mention is a single Chinese character.
- **IDF Score.** IDF Score of the mentioned string in all questions. We use 4 methods to compute the IDF score according to wikipedia[1].
- **Post- and Pre-word Possibility.** The possibility of the preword and postword to appear before or after a golden mention. OOV will set to 0.05.
- **Other Features.** Whether there is any Chinese in the mention, whether the mention equals to the entity name, whether the mention is covered by other mentions.

3.2 Candidate Predicates Generation

We use the same method as [19] to evaluate whether semantic of the question pattern can cover the predicate (see eq 1), but most of tricks such as question classification and high frequency entities filtering are deleted. A variant (see eq 2) is used to evaluate whether semantic of the predicate can cover the question pattern, where ave_q is the average vector of words in all questions, which is designed to match the stop words. The word segmentation method in this section is the same as that in [19]. Therefore, all possible words in questions and predicates will take into account. The detailed explanation of this word vector based evaluation method and discussions of the chosen word segmentation method can be found in Section 3.2 of [19].

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

$$S_q = \frac{\sum_j (lq_j * \max_{wp_i \in p \bigcup \{ave_q\}} Cos(wp_i, wq_j))}{\sum_j lq_j} \tag{2}$$

[1] https://en.wikipedia.org/wiki/Tf–idf

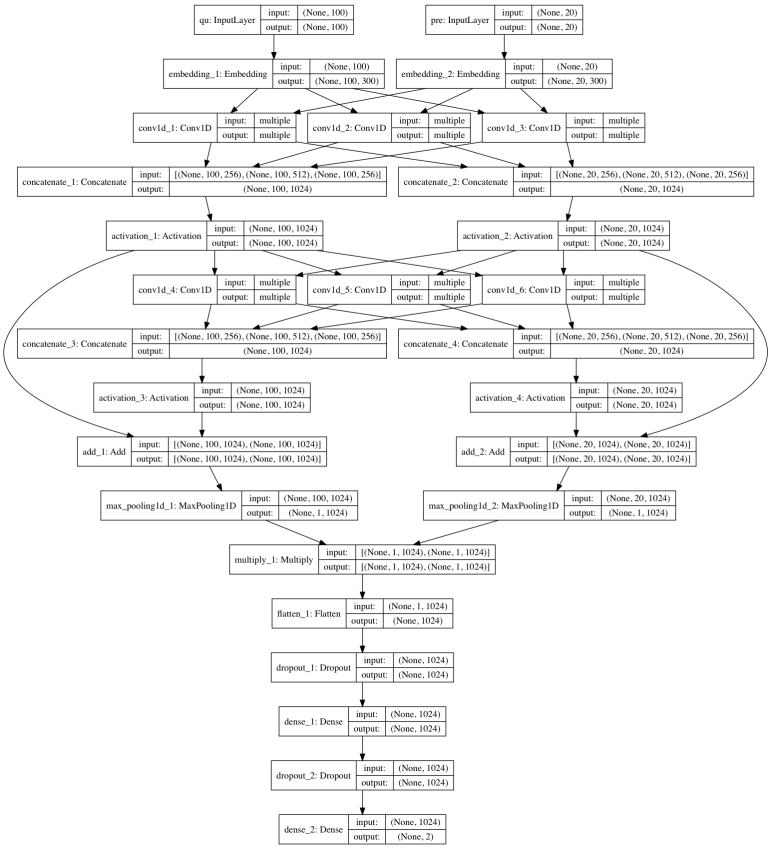

Fig. 2. Architecture of Our Deep CNN Model

In order to limit the amount of candidate entity-predicate pairs in reranking procedure, we used a linear combination of these feature(see eq 3) to filter out the unlikely candidates which is similar to the previous work[19]. Where l_{men} represents the length of the mention, and l_{pre} represents the length of the predicate. If an entity mention entails more than one KB entities which have the same predicate, only the predicate of the first entity (ordered by appearance in KB file) will be considered, so that no duplicate entity-predicate pair will be generated.

$$S_f = (S_p + S_q * 0.8)/1.8 * 1.4 + 0.1 * l_{men} + 0.00001 * l_{pre} \tag{3}$$

3.3 Deep CNNs Architecture

Deep convolutional neural networks are adopted to rerank the candidate entity-predicate pairs. The detailed architecture of our deep CNNs model used in submission version are illustrated in Figure 2. This model evaluates the similairty

between a prediate and a question pattern, that is the question without the entity mention. Pretrained word vectors are used to represent inputs, followed by several convolutional blocks (2 convolutional blocks in Figure 2) to generate high level features. Then, after max-pooling layers, element-wise multiplication are adopted to combine features from questions and predicates. Finally, a MLP(multilayer perceptron) with dropout is used to evaluate the final similarity. The parameters of convolutional layers are shared between the processing of predicates and questions.

Inspire by GoogLeNet[16], there are multiple filter widths in each convolutional block (in Figure 2, 256 filters with width 1, 512 filters with width 2, and 256 filters with width 3). Following ResNet[15], there are residual connections between neighbouring blocks. Limited by the pool improvement brought by deeper model and computing capability, the submission version has only 2 blocks.

3.4 Ranking

A linear combination of all intermediary scores is adopted to generate the final ranking of candidate answers. Since the high accuracy of entity linking (section 3.1) and the good performance of the single feature produced by deep CNNs (section 3.3) or word vector based approach (section 3.2), the combination equation is very rough without finely adjusting (see eq 4). Where S_{men}, S_f, and S_{cnn} are score of entity mentions, entity-predicate pairs evaluated by word vectors based approach, and predicates evaluated by CNNs respectively.

$$S_{final} = S_{men} + S_f + S_{cnn} * 2 \tag{4}$$

4 Experiment

4.1 Dataset

The dataset is published by NLPCC 2017 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 6M subjects, 0.6M predicates and 16M objects are involved. The 2017-training set contains 14,609 2016-training question-answer pairs and 9,870 2016-testing question-answer pairs. The 2017-testing set contains 7,631 question-answer pairs. The answers are labeled by human and most questions can be answer by a KB object.

4.2 Settings

All word vectors in our system are the same as the word vector list in the pervious work[19], which uses word2vector tools produced by Tomas Mikolov[1]

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

and CBOW[17] model trained on Baidubaike corpus. Word list used in word segmentation consists of all words in the word vectors list.

The parameters used in the GBDT entity linking model are: max depth=8, eta=0.1, objective=reg:logistic, nrounds=100. When training the CNN models, the batch size is 64, the loss function is binary crossentropy, and the optimizer is adadelta[23]. The submission version have trained for 21 epoches, but the best f1-score with the same settings appeared when 7 epoches finished and reached 47.35%. The CNN models are implemented by keras[2].

In entity linking procedure, only the mentions rank in top 3 with score higher than 0.01 times of the top mention's will left, which is our mentions filter rule. Only top 20 candidate entity-predicate pairs will be used in CNNs.

Because of the instability of the performance of CNNs over training epoches, an ensemble learning method is implemented. The S_{cnn} is the average of outputs of 8 CNNs. Four of them have the same architecture as Figure 2, and the others are similar but have 384 filters with width 1 and 640 filters with width 2 in every convolutional blocks. All of the CNN models have different seeds in initialization.

Although most of the negative entity-predicate pairs have been filtered out in candidate predicates generation before training CNN models, the amount of positive and negative samples is still unbalance. So a dynamic negative sampling approach is adopted. The possibility of a negative entity-predicate pair P_{ep_i} is shown in eq 5, where $rank_{ep_i}$ is the rank of this entity-predicate pair in its question scored by the end of the last iteration. It is just like a simple generative adversarial mechanism, where the generative model is the last iteration of the discriminative model.

$$P_{ep_i} = min(1.0, \frac{16.0}{rank_{ep_i}^2}) \tag{5}$$

4.3 Results

Table 2. Entity Linking Results

Dataset&Settings	Acc@1	Acc@3	Acc@10	#questions	Rec_filter
5f-cv 2016 train	98.75%	99.89%	——	14306	99.82%
2016 test	98.57%	99.81%	99.94%	9482	99.75%
5f-cv all trn	98.74%	99.89%	99.97%	23788	99.84%
2017 test	92.23%	98.41%	99.86%	4829	97.58%

Entity Linking The results of our entity linking model are shown in Table 2. We use 5-fold cross-validation to test our model on 2016 and 2017 training datasets

[2] https://keras.io

as well as each test datasets with the corresponding training data. Rec_filter is the recall of our mentions filter rule. Compared with the previous work[20], on 2016 training data, the accuracy of our model (98.75%) is a little lower than the f1-score of theirs (99.04%). But they just labeled 14033 questions while we labeled 14306 and every question in our data has only one golden mention. So it is not obvious that which model is better.

Candidate Predicates Generation Some detailed information is demonstrated in Table 3, including number of questions, number of candidate mentions per question, and number of candidate KB triples per question. Since the top-1 accuracy of entity linking on 2017 testing data gets lower, the entity filter holds more entity mentions per question automatically.

Table 3. Detailed Information in Candidate Predicates Generation

Dataset	#questions	#men_ave	#triple_ave
2016-train	9870	1.499	32.28
2016-test	14609	1.473	35.68
2017-test	7631	1.893	62.93

Table 4. Performance of Candidate Predicates Generation on 2016 Test-Set

System	Pre@1	Pre@2	Pre@5	Pre@20
baseline[19]	82.41%	87.06%	89.84%	91.02%
baseline-rules	81.76%	86.75%	89.70%	90.95%
Full-S_q	82.17%	87.18%	90.24%	92.01%
Full	82.97%	87.50%	90.44%	92.02%

Furthermore, results of the word vector based approach with different settings on 2016 testing set are shown in Table 4. Baseline is the best system in NLPCC 2016 KBQA task[19]. But for impartial comparison with our approaches, only one object will be answered for the same entity-predicate pair, so that the top-n precision (n>1) will be lower than reported. Baseline-rules is the baseline system without the tricks such as question classification and pattern based training, which is the actual baseline of our system. We think these rules should be summarized by CNNs automatically. Full system using entity linking filter and the reverse word vector based similarity S_q. From Table 4, it is obvious that both the entity linking and the reverse similarity can improve the performance, and the limitation of candidate entities can largely elevate pre@20, which is an important indicator for CNN reranking.

CNN Reranking The results on 2017 testing set of our CNN models with different depth are listed in Table 5. $+s_f\&s_{men}$ stands for the combination of ensemble CNNs and previous features. Each block has 1024 filters same as Figure 2. It seems that going deeper can bring an unsteady improvement. Limited by computing capability, we could not finely tune the parameters such as filter numbers and block structures. So there is still large potential for deep architectures. The best pre@1 of our submission architecture is 47.35%, which contains 8 2-block models. The word vector based feature s_f can achieve 43.10% on 2017 testing set. Combine pervious features with CNNs gains prominent improvement.

Table 5. Pre@1 of CNNs with Different Depth on 2017 Test-Set

#blocks	1	2	3	5	All
Single Model	43.57%	43.82%	43.85%	42.13%	—
*4 Ensemble	44.32%	44.45%	44.16%	43.28%	44.62%
$+s_f\&s_{men}$	47.32%	47.31%	47.23%	46.44%	47.45%

The detailed results of our submission architecture are demonstrated in Table 6. CNN models are trained for 17 epoches on 2016 dataset and 7 epoches on 2017 dataset. Accorading to [19], if the performance is judged by finding the correct entity-predicate pair, the accuracy of baseline system will be up to 85.61% on 2016 testing set while that of our system will be 89.65%.

Table 6. Full System Performance

	2016 testing set			2017 testing set		
	Pre@1	Pre@2	Pre@5	Pre@1	Pre@2	Pre@5
baseline[19]	82.41%	87.06%	89.84%			
s_f only	82.97%	87.50%	90.36%	42.94%	48.67%	54.75%
CNN Single	84.55%	88.63%	91.03%	43.63%	49.98%	55.59%
CNN Ensemble	85.40%	89.01%	91.17%	44.31%	50.18%	56.05%
name_system(Full)	86.60%	89.67%	91.38%	47.35%	52.47%	56.74%

The submission results of NLPCC2017 evaluation task are shown in Table 7(the top 5 results of 14 submissions in total). Our system achieves the best performance among all teams.

4.4 Upper Bound Analysis

However, our system can only label golden KB triples on 63.28% of 2017 testing questions, witch is the upper bound of our system. About 22% of answers in 2017

Table 7. Evaluation Results in this Evaluation Task

Team	F1 Score
PKU.name_system(Ours)	**47.23**%
NEU	41.96%
PKU.ICL	40.68%
ZJU.TeamTCM	40.08%
CCNU.NLP-Blaze	38.63%

testing dataset are not KB objects and about 14% of them whose topic entity mentions are aliases of KB entities and are not mentioned in file "nlpcc-iccpol-2016.kbqa.kb.mention2id" so that our entity liking method becomes invalid with them. So, aliases linking is also very important and more imformation besides the given KB is also in demand.

4.5 Further Experiments

We also do experiments on a subset of qald dataset[2]. Since training data in qald are not large enough for deep CNN models, the s_f only setting are used, which is an unsupervised method. 78 single relation factoid questions in English are selected from the training set of qald-6 task 1 (Multilingual question answering over RDF data, which contains 350 questions) and 82% (64/78) of them can be answered correctly by our system, which demonstrates that language is not a restriction of our system.

5 Conclusion

In this paper, we present a complicated KBQA system consists of features based entity liking, word vector based candidate predicate generation, and deep CNNs based reranking approach, which can answer simple-relation Chinese questions. For the unbalance of CNNs inputs, we present a generative adversarial like negative sampling approach. Our system obtained the first place in the contest of NLPCC 2017 Shared Task 5 (KBQA sub-task). Detailed experimental results are demonstrated, which can be helpful for other developers to understand the contributions of our components. In the future, we would like to extend our system to answer multi-relation questions and try to combine information from object of KB triples.

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

[2] https://qald.sebastianwalter.org

R&D Program of China (Grant No. 2015AA015403), Natural Science Foundation of China (Grant No. 61672057, 61672058).

References

1. Wen-tau Yih, Xiaodong He, and Christopher Meek.: Semantic Parsing for Single-Relation Question Answering. Meeting of the association for computational linguistics.(2014)
2. 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. (2015)
3. Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman: Deep Learning for Answer Sentence Selection. Computer Science. (2014)
4. Yi Yang, Wen-tau Yih, and Christopher Meek WikiQA: A Challenge Dataset for Open-Domain Question Answering. In Proceedings of EMNLP. (2015)
5. Alexis Conneau, Holger Schwenk, Yann Le Cun, and Loïc Barrault: Very Deep Convolutional Networks for Text Classification. In Proceedings of EACL. (2017)
6. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin: Convolutional Sequence to Sequence Learning. arXiv preprint arXiv: 1705.03122v2. (2017)
7. Xuchen Yao, and Benjamin Van Durme.: Information extraction over structured data: Question answering with freebase. In Proceedings of ACL.(2014)
8. Anthony Fader, Luke Zettlemoyer, and Oren Etzioni.: Open Question Answering Over Curated and Extracted Knowledge Bases. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM. (2014)
9. Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang.: Semantic Parsing on Freebase from Question-Answer Pairs. In Proceedings of EMNLP. (2013)
10. Jonathan Berant, and Percy Liang Semantic Parsing via Paraphrasing. In Proceedings of ACL.(2014)
11. Percy Liang Michael Jordan, and Dan Klein.: Learning dependency-based compositional semantics. In Proceedings of ACL. (2011)
12. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives.: DBpedia: A nucleus for a web of open data. In The semantic web, pages 722–735. (2007)
13. Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor.: Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, pages 1247–1250 (2008)
14. Karen Simonyan, and Andrew Zisserman.: Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science. (2014)
15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.: Deep Residual Learning for Image Recognition. In Proceedings of Computer Vision and Pattern Recognition. (2015)
16. Szegedy, Christian, Ioffe, Sergey, Vanhoucke, Vincent, abd Alemi, Alexander.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI Conference on Artificial Intelligence. (2017)
17. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean.: Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS.(2013)

18. Zhonglin Ye, Zheng Jia, Yan Yang, Junfu Huang, and Hongfeng Yin.: Research on Open Domain Question Answering System. In Proceedings of NLPCC.(2015)
19. Yuxuan Lai, Yang Lin, Jiahao Chen, Yansong Feng, and Dongyan Zhao.: Open Domain Question Answering System Based on Knowledge Base. In Proceedings of NLPCC.(2016)
20. Fengyu Yang, Liang Gan, Aiping Li, Dongchuan Huang, Xiaohui Chou, and Hongmei Liu.: Combining Deep Learning with Information Retrieval for Question Answering. In Proceedings of NLPCC.(2016)
21. Zhiwen Xie, Zhao Zeng, Guangyou Zhou, and Tingting He.: Knowledge Base Question Answering Based on Deep Learning Models. In Proceedings of NLPCC.(2016)
22. Linjie Wang, Yu Zhang, Ting Liu.: A Deep Learning Approach for Question Answering over Knowledge Base. In Proceedings of NLPCC.(2016)
23. Matthew D. Zeiler.: ADADELTA: An Adaptive Learning Rate Method. Computer Science. (2012)