

A Chinese Question Answering Approach Integrating Count-based and Embedding-based Features

Benyou Wang¹, Jiabin Niu¹, Liqun Ma¹, Yuhua Zhang¹, Lipeng Zhang¹,
Jingfei Li¹, Peng Zhang^{*1}, and Dawei Song^{1,2}

¹ Tianjin Key Laboratory of Cognitive Computing and Application, School of
Computer Science and Technology, Tianjin University, Tianjin, P.R. China
{waby,niujiabin,liqun,yuhuazhang,jingfeili,pzhang}@tju.edu.cn,zh583007354@qq.com

² Department of Computing and Communications, The Open University, UK
dawei.song2010@gmail.com

Abstract. Document-based Question Answering system, which needs to match semantically the short text pairs, has gradually become an important topic in the fields of natural language processing and information retrieval. Question Answering system based on English corpus has developed rapidly with the utilization of the deep learning technology, whereas an effective Chinese-customized system needs to be paid more attention. Thus, we explore a Question Answering system which is characterized in Chinese for the QA task of NLPCC. In our approach, the ordered sequential information of text and deep matching of semantics of Chinese textual pairs have been captured by our count-based traditional methods and embedding-based neural network. The ensemble strategy has achieved a good performance which is much stronger than the provided baselines.

Keywords: Question Answer, DBQA, semantic matching, Chinese text

1 Introduction

Question Answering (QA) has attracted great attention with the development of Natural Language Processing (NLP) and Information Retrieval (IR) techniques. One of the typical tasks named document-based question answering (DBQA) focuses on finding answers from the question's given document candidates. Compared with the traditional document retrieval task, DBQA system usually uses fluent natural language to express the query intent and desires an accurate result which has discarded most unmatching candidates.

Due to the short length of the text in DBQA task, data sparsity have become more serious problems than those of the traditional retrieval task. The relevance-based IR methods like TFIDF or BM-25 cannot solve these semantic matching problems effectively. Thus, word embedding technology [1] has been applied in

* Corresponding author: pzhang@tju.edu.cn

some English QA system as well as the Chinese QA system. Moreover, the question text is natural language with complete syntax structures instead of some keywords in document-retrieve task. A sentence should be considered as a sequence or a tree instead of an unordered word bag, and each components has different semantic contributions to the whole sentence. In summary, an effective QA system should consider the following problems simultaneously.

- 1) Matching the semantics-similar texts which is synonymous paraphrased.
- 2) Taking the sequantial information of the question text into consideration, instead of an unordered set of words.

For the first problem, enumerating all the paraphrase rules of text seems to be impossible. We usually adopt the embedding-based method in which two words have a closed embedding representations when they usually appear in the similar context. These representation can capture the semantic link between independent terms to some extent in a distributed way. For the second problem, people are more likely to firstly elaborate the premise and then ask the related issues under such premise according to the Chinese expression habit. In the bag-of-words model, an unordered set of words in questions will lose the information to distinguish the premise and issues. We utilize the position-aware information in our count-based model and keep the order of the word or character sequence in the neural network during the row-pooling and col-pooling operations.

This paper elaborates an approach for the Open Domain Questin Answering shared sub-task of Document-based QA task in NLPCC-ICCPOL 2016. We combine the count-based and embedding-based method with an ensemble strategy. In order to adapt to the Chinese expression habit, we integrate the features of Chinese into both the count-based method and embedding-based method, which achieves significant improvement upon baselines in the final evaluation.

2 Related Work

QA task focuses on automatically understanding natural language questions and selecting or generating one or more answers which can match semantically the question. Due to the shorter text than the traditional task of document retrieval, structured syntactic information and the lexical gap are two key points for QA system. For the first point, tree [2] and sequential [3] structure have been proposed to utilize the syntactic information instead of an unorderd bag-of-word model. Some efforts like lexical semantics [4], probabilistic paraphrase or translation [5] have been made to alleviate the problem of lexical gap. Moreover, feature-based ensemble method [6] tries to combine both the semantic and syntactic information to rank the answers by the data-driven learning mechanism.

Recently, the end-to-end strategy motivates researchers to build a deep symanctic matching model which can also model the sequential text. With the development of the embedding-based neural network, deep learning has achieved a good performance in the QA task [7]. Severyn et al. propose a shallow convolutional neural network (CNN) which combines the ordered overlapped information into the hidden layer [8]. Recurrent neural network (RNN) and the following long

short-term memory neural network (LSTM) [9][10] which can model the sequential text are also applicable for the textual representation and matching of question and answers. Santos et al. propose an attentive pooling networks with two-way attention mechanism for modelling the interactions between two sequential text, which can easily integrating a CNN or RNN network [11].

3 Methods

3.1 Data Exploration

The provided dataset of the DBQA task contains a training dataset and a testing dataset. There are 181882 question-answer pairs with 8772 questions in the training set, and 122532 pairs with 5997 questions in the testing set.

Word-level and character-level overlap Intuitively, The question-answer pair with more overlapped words seems to be more topic-relevant, which means a higher matching probability. In the whole training test, we get the trend as showed in the Fig. 1. It is easily found in the range from 0 to 13 of the x-axis that the more overlapped words between the question-answer pairs, the more likely the QA pairs match. Data are dispersed in the range between 15 and 28 because the samples are not enough. Moreover, the information of character-level overlap showed in Fig. 2 will cover many paraphrased patterns of Chinese.

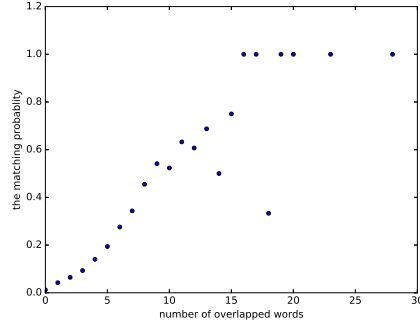


Fig. 1. The x-axis means the number of overlapped words in both question and answer sentences. y-axis refers to the probabilities of becoming the target answers.

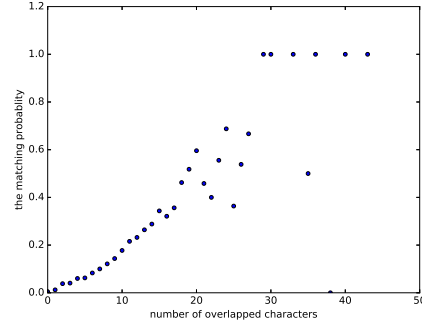


Fig. 2. The x-axis means the number of overlapped characters in both question and answer sentences. While y-axis refers to the probability of becoming the target answer.

Sequential structure information Traditional IR model like TF-IDF or BM25 model treats a query or a document as a bag of words, in which the sequential information of structure is ignored. In the scenario of QA system with

a shorter length of questions and answers, sequential information may help a lot for the matching of the question-answer pairs and a more elaborate model which takes the sequential information into consideration is needed. Roughly speaking, the words in different positions of a sentence may reflect different syntactic and semantic structures. For the example of the question “中央大学的首任校长是谁？”, the word “中央大学” in the forward position is the limited premise of the issues of the latter words “首任” and “校长” while the rearward word may be more relevant to the issues. In the training and testing set, we easily find the positive statistical correlation between the overlapped position and its corresponding probability of question-answer matching in Fig. 3 for word-level overlap and Fig. 4 for character-level overlap.

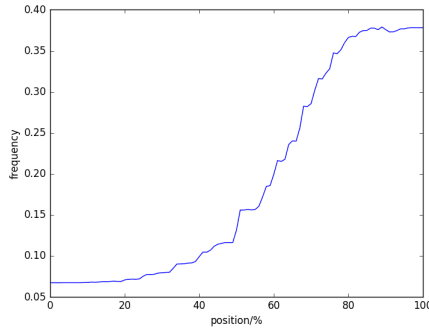


Fig. 3. The x-axis refers to the relative position of the overlapped word in the question sentence. the y-axis means the question-answer matching probability

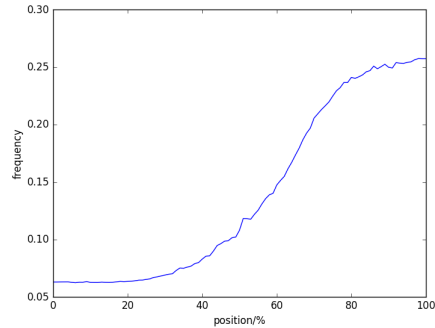


Fig. 4. Similar to Fig. 3, it shows the relationship between the relative position of the overlapped characters in question sentences and the matching probabilities.

3.2 Data Preprocessing

Due to the lack of the obvious boundaries of Chinese, we use the pynlpir³ [12] to segment the Chinese text. Stopwords are removed for dropping the useless high-frequency words which are not discriminative and have little semantic meaning. The 300-dimension word embedding is provided by the NLPCC competition. Moreover, we have trained an embedding model of some crawled pages from the site of Baidu Baike (<http://baike.baidu.com/>).

3.3 Feature Extraction

Questions’ categories and answers’ classification The questions is divided into 5 categories in our paper. 3 of them are concerned with name entity, which

³ <https://github.com/tsroten/pynlpir>

are *person*, *place* and *organization*. The remaining two are *time* and *number*. Due to the lack of large-scale labeled data, we can not adopt a learning-based question classifier [13]. Alternatively, a template-based question classifier can cover most cases for the simplified taxonomy. The first three NER-based categories can be recognized by the LTP online API⁴. Meanwhile, we use templates of regular expression to distinguish the types of *number* and *time*.

Overlap score In Sec 3.1, there is a statistical correlation between the matching probability and the overlapped information. For Chinese text, it's hard for us to find the dictionary which can contain all the near-synonym pairs. An alternative approach is to use the character-based metric due to the fact that many synonymous paraphrased pairs share the same characters in Chinese. We calculate both the word-level and character-level scores of overlap as follows:

$$Score_{overlap}(Q, A) = \sum_{q_i \in Q}^n freq_A(q_i) \cdot weight(q_i) \quad (1)$$

where a question Q has n words (characters) and the answer A has m words (characters). The weighted model is based on the position of q_i in the sentence. $freq_A(q_i)$ is denoted as the smoothed frequency of the q_i in the answer A .

BM25 score The BM25 model is implemented as Eq. 2.

$$Score_{bm25}(Q, A) = \sum_{q_i \in Q}^n IDF(q_i) \cdot \frac{freq_A(q_i) \cdot (k + 1)}{freq_A(q_i) + k \cdot (1 - b + b \cdot \frac{Length_A}{Length_{avg}})} \quad (2)$$

where $freq_A(q_i)$ is the frequency of the q_i in the answer A . k and b are adjustable parameters for the specific task. $Length_A$ and $Length_{avg}$ are the length of the answers A and the average length of the whole answers, respectively.

Weighted Embedding Embedding technology embeds words into a uniform semantic space, which makes it possible to find the relationship between words. Sentence is simply considered to have been lapped by words linearly. Different words can contribute different weights for the whole meaning of a sentence, which depends on their position, semantic structure and IDF. We get the representation of a word or a Chinese character as Eq. 3

$$Representation(S) = \frac{\sum_{i=0}^n weight(s_i) \cdot \overrightarrow{embedding(s_i)}}{\sum_{i=0}^n weight(s_i)} \quad (3)$$

s_i is the character or word in a sentence (question or answer), and $embedding(s_i)$ is the corresponding embedding vector. Then we calculate the inner product between the representation of questions and answers as the final score.

Neural network Besides the weighted combination of the inside word embedding, we have built a neural network which is showed as Fig. 5. In our approach,

⁴ <http://www.ltp-cloud.com/>

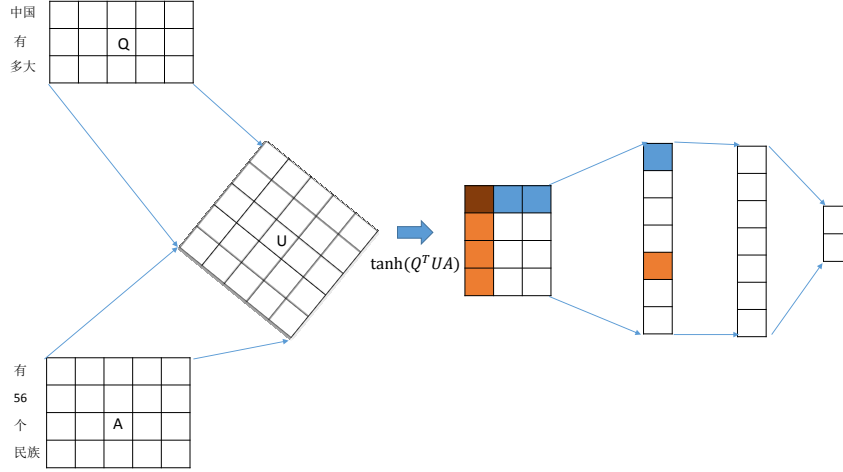


Fig. 5. The structure of our neural network

both word-level embedding and character-level embedding have been adopted to form the sentence matrices of question and answers. A trainable matrix U is used for bridging the question embedding matrix and the answer embedding matrix. The following \tanh function can avoid the explosion of the previous activated value. The information of the ordered position can still be remained in the full-connection layer by the operation of row-pooling and col-pooling instead of max-pooling [11]. After the softmax layer, the last output layer contains two floating numbers which represent the probabilities of question-answer matching and unmatching respectively. Cross-entropy loss function is used for the optimization process.

Other features Edit distance is usually used to measure the similarity of textual strings. While Jaccard index gives the similarity of morphemic sets between the textual pairs. The length of answer is often considered as a significant feature.

3.4 Model Ensemble

We have presented various fundamental features in last chapter, which will directly affect the degree of how question and answer matches. We adopt a linear regression model, learn-to-rank model⁵ and tree-based boost model to integrate those features after the normalization of Z-score.

⁵ <https://sourceforge.net/p/lemur/wiki/RankLib/>

4 Experiment

The provided baselines and our result are showed as follows:

Table 1. The result of our approach.

method	MAP	MRR
Average Word Embedding	0.4610	0.4610
Machine Translation	0.2410	0.2412
Paraphrase	0.4886	0.4906
Word Overlap	0.5114	0.5134
Count-based features	0.7750	0.7756
Embedding-based features	0.7467	0.7470
All features	0.8005	0.8008

Due to the fact that the above baselines are based on the bag-of-word model and do not have a learn-based mechanism, the performance is rather poor. In the final evaluations, our approach gets the MRR of 0.8008 and ranks 5th among the 18 submissions (4th among the 15 teams).

In our approach, the final scores of some models are treated as features of the ensemble method. As mentioned in the Sec. 3.1, features which can effectively model both syntax and semantic information may be more likely to be correlated to matching labels. In the syntax of Chinese expression, for example, the key words which are related to the issues of question usually appear in the latter positions of the question sentence, while the words in the front positions are more related to the indiscriminative premise which are satisfied by most candidate answers. Moreover, an effective semantic match strategy is also needed. We adopt both the character-level and word-level models in our approach, and a deep neural network may help a lot while our network is a little shallow and compact.

In our experiment, the traditional models like BM25 do not have the potential to do the semantic matching, while the character-based model outperforms the word-character in Chinese. A position-aware deep neural network with the end-to-end strategy may be the trend for the QA tasks. Due to the low-dimension features space, the linear regression has achieved a pretty good performance comparing to the learn-to-rank method or the tree-based boosting methods.

5 Conclusion and Future Work

In this paper, we report technique details of our approach for the sub-task of NLPCC 2016 shared task Open Domain Question answering. Some traditional methods and neural-network based methods have been proposed. In our approach, we combine the characteristics of Chinese text with our models and achieve a good performance by an ensemble learning strategy. Our final performance is not so great due to the shallow structure of the neural network.

In our opinions, an effective representation which contains the sequential (or tree-based) information of short text and the corresponding effective semantic matching are the two key factors of the QA system. Both a RNN network which can directly models sequential texts and a CNN network which is more flexible have the potential to get better performances after some adaptations in the textual data. Moreover, although there are many shared characteristics between English and Chinese text, an end-to-end system which is specifically applicable for Chinese can also be the trend for Chinese Question-Answering system.

Acknowledgements. The work presented in this paper is sponsored in part by the Chinese National Program on Key Basic Research Project (973 Program, grant No. 2013CB329304, 2014CB744604), the Chinese 863 Program (grant No. 2015AA015403), the Natural Science Foundation of China (grant No. 61402324, 61272265), and the Research Fund for the Doctoral Program of Higher Education of China (grant No. 20130032120044).

References

1. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.
2. X. Yao, B. V. Durme, C. Callison-Burch, and P. Clark, "Answer extraction as sequence tagging with tree edit distance," in *Conference of the North American Chapter of the Association for Computational Linguistics*, 2013.
3. Z. Wang and A. Ittycheriah, "Faq-based question answering via word alignment," 2015.
4. W. T. Yih, M. W. Chang, C. Meek, and A. Pastusiak, "Question answering using enhanced lexical semantic models," in *Meeting of the Association for Computational Linguistics*, 2013, pp. 1744–1753.
5. G. Zhou, L. Cai, J. Zhao, and K. Liu, "Phrase-based translation model for question retrieval in community question answer archives," in *The Meeting of the Association for Computational Linguistics*, 2011, pp. 653–662.
6. A. Severyn, "Automatic feature engineering for answer selection and extraction," in *EMNLP*, 2013.
7. L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *Computer Science*, 2014.
8. A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *SIGIR*. ACM, 2015, pp. 373–382.
9. D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in *Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2015.
10. M. Tan, B. Xiang, and B. Zhou, "Lstm-based deep learning models for non-factoid answer selection," *Computer Science*, 2015.
11. C. D. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," 2016.
12. T. Liu, W. Che, and L. I. Zhenghua, "Language technology platform," in *COLING 2010*, 2010, pp. 13–16.
13. X. Li and D. Roth, "Learning question classifiers," *Coling*, vol. 12, no. 24, pp. 556–562, 2003.