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

No Author Given

No Institute Given

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

Quesion 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 quetion 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 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 English or Chinese 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 [2]. For a query *'a cute pet'*, *'a dog'* or *'a cat'* cannot be found in the return list due to the gap between independent terms, but the distributed representation can capture the semantic link between 'pet' and 'dog' ('cat'). In Chinese, many similar words may share the same character based on the specific word formation of Chinese. For exmaple, the two words "表演" or "演出" have the same meaning as "perform or show". Some character-based technology may help a lot [3]. 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 method and embedding-based method with an ensemble learning 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.

The aim of the task is to build a system to select the answers of thousands of questions. The target answers are only supposed to be selected from the question's given document which contains a set of answer sentences. The results will finally be evaluated by the evaluation metrics to figure out the performance of our system. The model should generate a set of relevance scores between the question and each answer sentence including the correct answer in the testing set. The evaluation toolkit which has testing set labeled golden answer annotations to questions will rank all answer sentences according to MRR scores and MAP scores.

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 [4] and sequential [5] structure have been proposed to utilize the syntactic information instead of an unorderd bag-of-word model. Some efforts like lexical semantics [6], probabilistic paraphrase or translation [7] have been made to alleviate the problem of lexical gap. Moreover, feature-based ensemble method [8] 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 symantic 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 [9][10]. Severyn et al. propose a shallow convolutional neural network (CNN) which combines the ordered overlapped information into the hidden layer [11]. Recurrent neural network (RNN) and the following long short-term memory neural network (LSTM) [12][13] which can model the sequential text are also applicable for the textual represention and matching of quetion 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 [14].

3 Methods

3.1 Data Exploration

There are 181882 quetion-answer pairs with 8772 questions in the training set, and 122532 question-answer pairs with 5997 questions in the testing set. More detailed information is showed in Fig. 1.

Table 1. The basic information of the training and testing set.

	training set	**testing set**
number of qa pairs	181882	122532
number of questions	8772	5779
average number of candidate answer	20.7	21.2
average length of the questions (charater)	46.3	46.2
average length of the answers (charater)	106.0	106.5
average number of correct answers	1.05	1.06

Question classification Question sentences' type is usually a kind of vital signals [15], which is a prerequisite but not sufficient condition when distinguishing whether the given sentence is a correct answer or not. For the question "中央大学的首任校长是谁?", a candidate answer can be correct only if it appears a name entity of person. We divide the question into the following categories as showed in Fig 2.

Word-level and character-level overlap For a question "佛罗伦萨什么时候降水比较多？", the unique answer "降水主要集中在冬季" will share the same

Table 2. The number of different types of question.

	training set	**testing set**
time	950	861
number	2135	1433
person name	1049	526
place name	583	394
organazation name	185	137
others	3870	2646
total	8772	5997

component “降水”. In most Chinese word segmentation tools, “降水” can be segmented as a single word, which is usually considered as the minimal granularity of semantic units. 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. For the question “ “年” 字有多少笔？ ” with the segmented list [“ “”,“年”,“” ”,“字”,“有”,“多少”,“笔”,“？ ”], the correct answer “笔划：6” has three words [“笔划”， “:”,“6”] and does not have any overlapped words with the question. But the character-based method can capture the overlapped Chinese charcter “笔”.

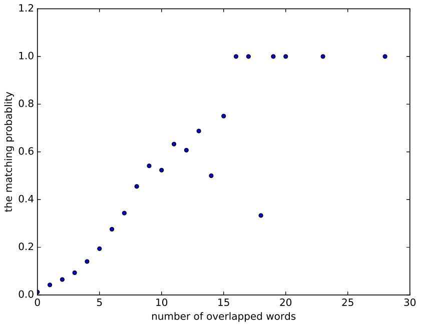

Fig. 1. The x-axis means the number of overlapped words in both question and answer sentences. y-axis refers to the probablites of becoming the target answers.

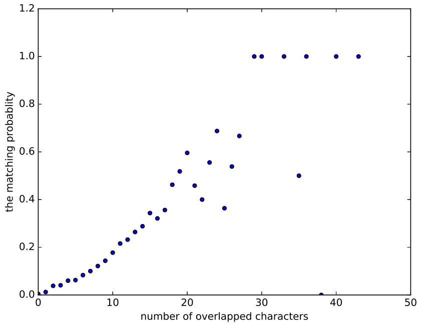

Fig. 2. The x-axis means the number of overlapped characters in both question and answer sentences. While y-axis refers to the probability of becoming the target answer.

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 “首任校长”. The rearward word “首任校长” may be more related to the issues, which will contribute more to question-answer matching. 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.

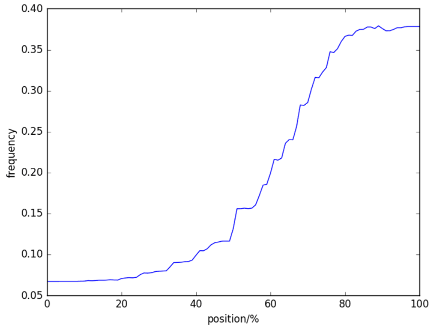

Fig. 3. The x-axis refers to the relative position of the overlapped word in a question sentence. x=0 means that the overlapped word is on the front of a sentence while x=100% is on the back. y-axis means the probality of becoming the correct answer.

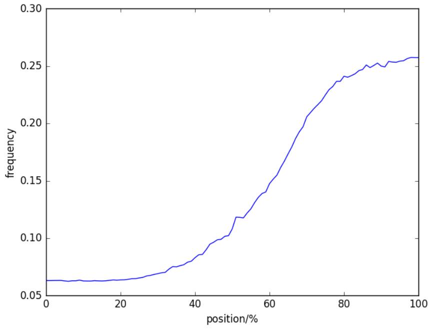

Fig. 4. Similar to Fig. 3, it shows the relationship between the relative position of the overlapped characters in question sentences and the matching probalities.

3.2 Data Preprocessing

Some preprocessing in Fig. 5 have been done before the feature extraction. Due to the lack of the obvious boundaries of Chinese text, we use the pynlpir package [1] [15] to segment both the question sentences and the answer texts. Stopwords [2] are removed for dropping the useless high-frequency words which are not dis-

[1] https://github.com/tsroten/pynlpir
[2] stopwords in http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html

criminative and have little semantic meaning. In order to get the classfication information of answer, we adopt the LTP online API [3] for Named Entity Recognition (NER). The 300-dimention word embedding is provided by the NLPCC competition. Moreover, we have trained an embedding model of some crawled pages in Baidu Baike [4].

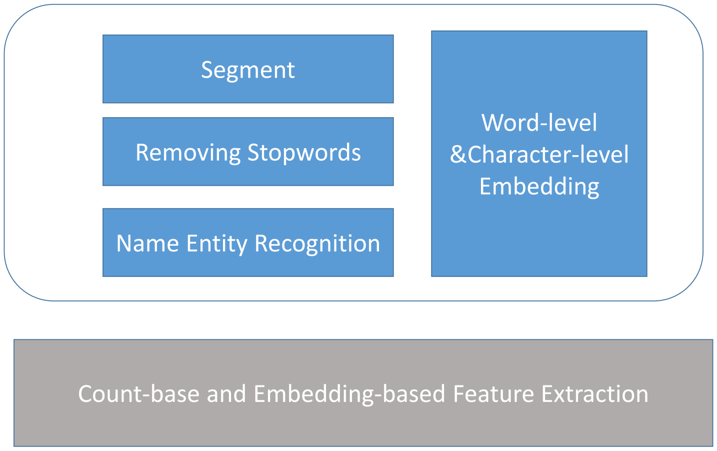
Segment
Removing Stopwords
Name Entity Recognition
Word-level &Character-level Embedding
Count-base and Embedding-based Feature Extraction

Fig. 5. The basic structure of the data prepare

3.3 Feature Extraction

Questions' categories and answers' classfication As mentioned in the Sec. 3.1, the questions can be divided into 5 categories. 3 of them are concerned with name entity, which 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 [16]. Alternatively, a template-based question classifier can cover most cases for the simplified taxonomy. Answers' classfication may be a multilabeled task, which means an answer can belong to many categories. The first three NER-based categories can be recognized by the LTP online API. Meanwhile, we use templetes of regular expression to distinguish the types of *number* and *time*.

In practice, two methods are adopted to form the category features. The first one is dummy Number, 5 categories are viewed as 5-dimention binary features whose default value is unmatching. If the question is identified as one kind of

[3] http://www.ltp-cloud.com/
[4] http://baike.baidu.com/

category, the corresponding feature will be filled with matching flags. The second method is adopting one-dimention feature, which reflects whether the category of question-answer pair matched or not.

Overlap In Sec 3.1, there is a statistical correlation between the matching probability and the overlapped information. In these features, the deletion of stop words seems to be vital due to the ignorance of the meaningless words. For Chinese text, it's hard for us to find the dictionay which can contain all the near-synonym pairs. An alternative approch is to use the charcter-based metric due to the fact that many synonymous paraphrased pairs share the same charcters in Chinese. We calculate both the word-level and character-level scores of overlap as follows:

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

where a question sentence Q has n words (characters) and the answer sentence 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 which is based on bag-of-words model is also implemented as Eq. 2.

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

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 $Lenght_{avg}$ 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 easier 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

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

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 represention of questions and answers as the final score.

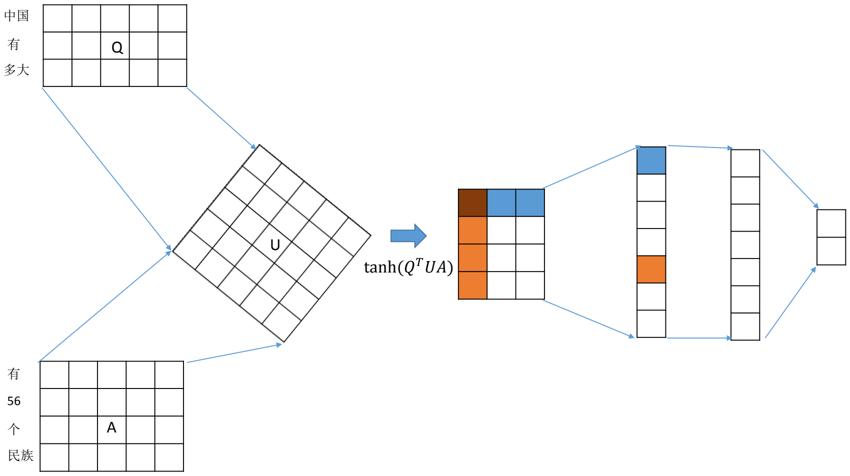

Fig. 6. The structure of our neural network

Neural network Besides the weighted combination of the inside word embedding, we have built a neural network which is showed as Fig. 6.

In our approach, 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 martrix and the answer embedding matrix. The following *tanh* function can avoid the explosion of the previous activated value. The information of the ordered postion can still be remained in the full-connection layer by the operation of row-pooling and col-pooling instead of max-pooling [14]. After the softmax layer, the last output layer contains two floating numbers which represent the probalities 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 length of similar strings in English. While Jaccard index gives the similarity of morphemic sets between the questions and answers. The length of answer sentence 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 and learn-to-rank model [17] to integrate those features [5] after

[5] https://sourceforge.net/p/lemur/wiki/RankLib/

the normalization of Z-score. Meanwhile some ensemble learning methods like boosting and bagging models [6] are also used [18].

4 Results

4.1 Dataset

The provided dataset [7] of the DBQA task contains a training dataset which has the ground truth and a testing dataset which does not have the ground truth. In the training dataset, each question has many candidate answers. The structure of the dataset can be illustrated by Tab.3, where “1” in the last column means the correct answer. The provided testing set only contains questions and their candidate answers. Each submission should only include a column of scores which will be evaluated by the evalution toolkit.

Table 3. Training data structure

question	candidate answers	label
俄罗斯贝加尔湖的面积有多大?	贝加尔湖是世界上最深和蓄水量最大的淡水湖。	0
俄罗斯贝加尔湖的面积有多大?	它位于布里亚特共和国(Buryatiya)和伊尔库茨克州(Irkutsk)境内。	0
俄罗斯贝加尔湖的面积有多大?	湖型狭长弯曲，宛如一弯新月，所以又有“月亮湖”之称。	0
俄罗斯贝加尔湖的面积有多大?	贝加尔湖长636公里，平均宽48公里，最宽79.4公里，面积3.15万平方公里	1
俄罗斯贝加尔湖的面积有多大?	贝加尔湖湖水澄澈清冽，且稳定透明（透明度达40.8米），为世界第二。	0

4.2 Evaluation Metrics

Our Question Answering system will be evaluated by MRP and MAP. Mean Reciprocal Rank (MRR) mainly indicates the quality of the return results depending on the rank of the correct answer, namely the higher the correct answer ranked, the better the results are.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{4}$$

where $|Q|$ stands for the total number of questions in the evaluation set. For the i^{th} question Q_i, $rank_i$ represents the position of the first correct answer in the generated answer set C_i. $\frac{1}{rank_i}$ equals 0, if C_i doesn't have the correspond answer with the golden answers A_i for Q_i.

Another evaluation metric is Mean Average Precision (MAP), which can be defined as follows.

$$MAP = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{\sum_{k=1}^{n}(P(k) \cdot rel(k))}{min(m, n)} \tag{5}$$

[6] http://xgboost.readthedocs.io/en/latest/

[7] http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html

MAP mainly refers to the average precision of the results. If the correct anwser ranks high in the retrieved answer set, the value of MAP will also be high accordingly. m denotes the number of correct answer sentences and n is the number of retrieved answer sentences. $P(k)$ is the precision at cut-off k, which is the rank in the sequence of retrieved answer sentences. $rel(k)$ equals 1 if the item at rank k is an answer sentence, and 0 if not.

4.3 Results

The competition has provided four baselines as follows:

Table 4. 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
Our approach	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 result ranks 5th among the 18 submissions (4th among the 15 teams).

4.4 Discussion

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 of QA data. In the syntax of Chiness 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 satified 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 the process of feature engineering, the dummy strategy (mentioned in Sec. 3.3) is less effective than the single value due to the data sparseness. The trandiational models like BM25 do not have the potential to do the semantic matching, while the characeter-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 [8], 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 repersentation which contains the sequencial (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 adaptions 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.

References

1. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.
2. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.
3. Y. Sun, L. Lin, N. Yang, Z. Ji, and X. Wang, *Radical-Enhanced Chinese Character Embedding.* Springer International Publishing, 2014.
4. 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.
5. Z. Wang and A. Ittycheriah, "Faq-based question answering via word alignment," 2015.
6. 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.
7. 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: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, Usa*, 2011, pp. 653–662.
8. A. Severyn, "Automatic feature engineering for answer selection and extraction," in *EMNLP*, 2013.

[8] Our codes can be found in the site https://github.com/anonymous.site which will be open after review due to the double-blind policy.

9. L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *Computer Science*, 2014.
10. M. Feng, B. Xiang, M. R. Glass, and L. Wang, "Applying deep learning to answer selection: A study and an open task," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2015.
11. A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 2015, pp. 373–382.
12. 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.
13. M. Tan, B. Xiang, and B. Zhou, "Lstm-based deep learning models for non-factoid answer selection," *Computer Science*, 2015.
14. C. D. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," 2016.
15. T. Liu, W. Che, and L. I. Zhenghua, "Language technology platform," in *COLING 2010, International Conference on Computational Linguistics, Demonstrations Volume, 23-27 August 2010, Beijing, China*, 2010, pp. 13–16.
16. X. Li and D. Roth, "Learning question classifiers," *Coling*, vol. 12, no. 24, pp. 556–562, 2003.
17. T. Y. Liu, *Learning to Rank for Information Retrieval.* Now Publishers, 2009.
18. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," 2016.