QAAN: Question Answering Attention Network for Community Question Classification

Yuntao Wang[1, 2], Weiqing Huang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

2021 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-7281-9505-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/WCNC49053.2021.9417482

***Abstract*—Community Question Answering (CQA) provides platforms for users with various backgrounds to obtain information and share knowledge. In recent years, with the rapid development of such online platforms, an enormous amount of archive data has accumulated, it becomes more and more difficult for expert users to identify desirable questions. In order to reduce the proportion of unanswered questions in CQA, facilitate expert users to find the questions they are interested in, question classification becomes an important task of CQA, which aims to assign a newly posted question to a specific preset category. In this paper, we propose a novel question answering attention network (QAAN) for investigating the role of the paired answer of questions for classification. Specifically, QAAN studies the correlation between question and paired answer, taking the questions as the primary part of the question representation, and the answer information is aggregated based on similarity and disparity with the answer. Our experiment is implemented on Yahoo! Answers dataset. The results show that QAAN outperforms all the baseline models.**

Keywords—Question Classification, Question Answering Attention Network, Community Question Answering.

1 Introduction

With the rapid development of wireless network, Community Question Answering (CQA) becomes a novel and popular approach to obtain information, which mainly overcomes some shortcomings of conventional search engines, provides an interactive searching experience and efficient retrieve in specific domains. Question Answering Communities are open-domain online platforms, such as Yahoo!Answers, Baidu Knows, etc., users can register easily, post questions of their own concerns, or provide answers. Since the popularity of CQA, a massive number of questions has accumulated in the community during the past few years. Hence, question classification becomes a very important task of CQA, which aims to assign a specific preset category to each question. Specifically, the question classification task uses reposited question-answer pairs to solve new-coming posts from users, to facilitate users to answer the questions that belong to their familiar areas more efficiently. However, it is difficult to assign questions to pre-defined categories in the community since a large number of synonyms, semantic features, and syntactic features in natural language. For example, "How can I lose weight in a few months?" and "Are there any ways of losing pound in a short period?" both are questions about seeking ways to lose weight, but the two questions contain almost no identical vocabulary and syntactic features. The conventional methods based on term frequency are not able to solve such issue. Also, due to the subjectivity of users, the redundancy is prevalent in CQA [1], which makes it even more difficult to identify the semantic meaning of different questions.

Aiming to retrieve archived information and reduce redundancy, we propose a Question Answering Attention Network (QAAN) for question classification of CQA, which uses an attention mechanism to assign different attention weights to questions and their paired answer. Therefore, the performance of question classification in CQA can be improved. Why the performance of question classification can be improved when the paired answer is taken into consideration? The reasons are from two perspectives. First, users of CQA can be simply categorized into two types, expert users and common users. Expert users often specialize in a certain domain, who are able to provide very detailed and professional answers. These answers are more likely to demonstrate the essential intention of the question, and more informative than the questions themselves. Therefore, studying the paired answers in CQA can improve the performance of question classification effectively. Secondly, we investigated a wide range of well-known CQA before proposing the QAAN, and found that the questions often comprise fewer words than answers. Also we observed that words and phrases of questions are very colloquial, but paired answers especially the best answers often are rich in vocabulary and highly specialized. The Attention mechanism captures the attention weights both in question and paired answer, extracting more semantic features. The main contributions of this work include:

(1) The paper adds answers information rather than question classification, enriched the input parameters of the model, which make classification more accurate.

(2) In the process of introducing answer information, an orthogonal decomposition strategy is used, which effectively reduces the redundant information brought by the introduction of answer information, thereby improving the performance of the model.

The rest of this paper is organized as follows: Section 2 introduces related work about question classification in CQA; Section 3 describes our proposed model QAAN; Section 4 details our experiment implementation, and analysis the experimental results; Section 5 summarizes the conclusion and suggests the further research potentials for the future.

978-1-7281-9505-6/21/$31.00 ©2021 IEEE

2 Related work

CQA is one of the tasks in natural language processing (NLP). Most of the conventional natural language processing tasks are based on statistical machine learning approaches. With the rapid development of deep learning [2], more and more researchers and scholars have applied deep learning to natural language processing tasks. Adhikari et al. [3] questioned the complexity of the existing neural network architecture for document classification, and proposed embedding dropout, weight dropping, and temporal averaging in the training process of simple Bi-LSTM. This model has achieved good performance on different datasets. The recent research by Melis et al. [4] showed that the fine-tuned model based on the standard LSTM outperforms other models. Vaswani et al. [5] showed that the model uses attention mechanisms respectively can achieve comparable performance with the model uses an encoder with attention mechanisms to convert sequences. This model has demonstrated that most of the complex neural network mechanisms are not imperative. Mohammed et al. [6] illustrated that Vanilla RNN and basic CNN models can achieve better results in knowledge-based questions and answers than complex architecture neural network models. Sculley et al. [7] claimed that the lack of rigor in domain knowledge can be easily solved by removing the noise, which has been by many examples in the paper. Lipton et al. [8] also agreed with these observations, clarifying that many authors often use fancy data formulas to confuse or impress reviewers rather than clarify factual issues. Yang et al. [9] proposed a sequence generation model (SGMs) based on encoder-decoders to generate a pair of labels for each document. This model has achieved good results on the relevant data sets. These are some recent models that have performed well in natural language processing tasks.

In recent years, with the rapid development of deep learning, which has been widely used in the domain of natural language processing. As the focus of many NLP researchers, the Community Question Answering has been flourished with the state-of-the-art models. Kim et al. [10] studied the deep learning model − convolutional neural network (CNN) with trained word embedding for sentence classification. Kalchbrenner et al. [11] developed a dynamic convolutional neural network (DCNN) that learns sentence semantics by simulating semantic information through the DCNN network for question classification. DCNN used a global k-max pooling operation to solve the issue that the sentences with different lengths, also learned the dependence of lengths for different sentences. Le et al. [12] developed a forest convolutional neural network (FCN) using forset as the input of convolutional neural networks. A random increase or decrease of branches was realized in FCN. Experiment results demonstrated that FCN has achieved state-of-the-art results in both sentiment analysis and question classification tasks. Mou et al. [13] proposed a tree-based convolutional neural network (TBCNN) with the sentence-dependent syntactic tree and component tree. TBCNN used the extracted structural features of sentences as components, applying the maximum pooling to merge multiple features. Komninos et al. [14] studied the effects of word embeddings on deep neural networks. The results showed that context-based word embedding achieved better performance in sentence classification tasks.

The methods we mentioned above have shown good results in question classification tasks, however, the answer information corresponding to the question neglected, especially for topics in a specific filed. But the information provided by answers usually is more informative than the questions themselves. Therefore, it is critical to use the corresponding answer to improve the performance of the question classification task. Hence, we propose QAAN to study the attention weighted features for both questions and answers.

3 Methodology

In this paper, the question classification task of community question answering can be described as a tuple of three elements $(\mathrm{Q}, \mathrm{A}, \mathrm{y})$, where $\mathrm{Q} = [q_1, q_2, \ldots., q_n]$ represents a question whose length is *N*. Each q_i is encoded by a one-hot vector, whose dimension is the same with the dimensions of vocabulary L. $\mathrm{A} = [a_1, a_2, \ldots., a_m]$ denotes the answer corresponding to the question whose length is *M*. Each a_i is encoded by a one-hot vector, whose dimension is the same with the dimensions of vocabulary L. $\mathrm{y} \in \mathrm{Y}$ indicates the category corresponding to the question. Therefore, the task is defined as: Given question-answer pair {Q, A}, the distribution of probability $\Pr(\mathrm{y}|\mathrm{Q}, \mathrm{A})$ is modeled by QAAN. The category with the maximum probability distribution is assigned as the label.

3.1 Overview of QAAN

The corpus used in this research is extracted from the Yahoo!Answers website. For each question we ensure that there is at least one paired answer. Each question is labeled as one of the categories. Two sets of word embeddings are obtained on the corpus by implementing the character embeddings (Kim et al. [16]) and GloVe (Pennington et al. [15]), respectively. The two embeddings are concatenated to preserve the validity of the position information effectively. The concatenation is used as the input embeddings. The overview of our QAAN model is illustrated as Fig.1.

Q_{order} represents the position information for each word, Q_{emb} is the concatenation of the two question embeddings. Similarly, A_{emb} is the concatenated word vectors in the corresponding answer, A_{order} indicates the position information of each word in the corresponding answer.

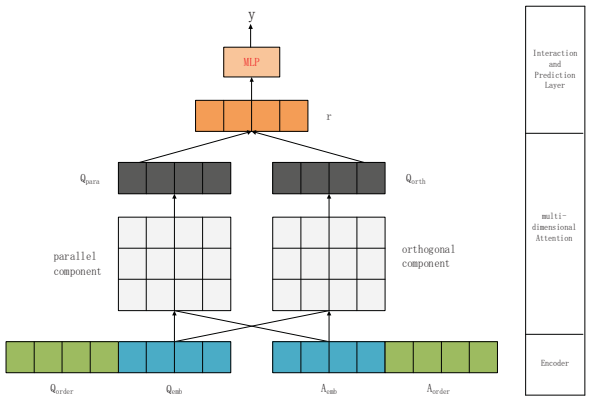

Fig.1 Overview of our proposed QAAN model

The word-level embeddings are composed of two modules: the Glove model proposed by Pennington et al. [15] which is trained on the Yahoo!Answers corpus, and character embedding proposed by Kim et al[16]. The concatenation of the two embeddings provides various advantages. The relationship of words is captured more accurately and precisely when the embedding is trained on the extracted corpus since the texts in CQA are different in grammar and spelling from News and reports. Also it has been proved that character embedding is effective for OOV (out-of-vocabulary), especially for CQA tasks.

The two embedding vectors are concatenated to form a word-level embedding. For each question and the corresponding answer, the word embeddings are represented respectively as, $Q_{emb} \in R^{d\times n}$ and $A_{emb} \in R^{d\times m}$, d denotes the dimension of the concatenated word embedding.

3.2 Question Answer Attention Network

The question-answers are analyzed from the perspective of similarity and disparity to aggregate the information to better represent the question-answer relationship. We apply the orthogonal decomposition strategy proposed by Wang et al.[17] to achieve the representation. We embed the words of corresponding answers into two directions, horizontal and vertical. The formula is shown as follows (Equation (1)(2)):

$$a_{para}^{ij} = \frac{a_{emb}^{j}\cdot q_{emb}^{i}}{q_{emb}^{i}\cdot q_{emb}^{i}} q_{emb}^{i} \quad (1)$$

$$a_{orth}^{ij} = a_{emb}^{j} - a_{emb}^{i,j} \quad (2)$$

The length of vectors in the formula is *d*. The details of obtaining the horizontal representation of paired answers are described, similarly, the process can be applied to the vertical direction. Q_{para} and Q_{para} are obtained during the process. Q_{para} and Q_{orth} are passed through a fully connected neural network to obtain multi-dimensional attention weights. Tanh is used as an activation function, which is similar to the method proposed by Shen et al.[18]. To maintain a sufficient amount of output while preventing huge fluctuations in the final score, the following formula is used to normalize the output. The output for each dimension is shown as Equation (3):

$$b_{para}^{i,j} = c\cdot \tanh\left(\frac{\left[W_{p1}a_{para}^{i,j}+b_{p1}\right]}{c}\right) \quad (3)$$

where $W_{p1} \in R^{d\times d}$ and $b_{p1} \in R^{d}$ are parameters learned by QAAN, C is a manually tuned hyper-parameter.

The word-level vector B is aligned by Equation (3). Then we normalize and expand the third party of Vector B to get the attention weight of each word in questions. The output is the weighted sum of the embedding of each word in the question divided by the embedding of the questions. The formulas are shown as: (Equation (4)(5)):

$$w_{para}^{i,j} = \frac{\exp(b_{para}^{i,j})}{\sum_{j=1}^{m}\exp(b_{para}^{i,j})} \quad (4)$$

$$q_{ap}^{i} = \sum_{j}^{m} w_{para}^{i,j} \odot a_{emb}^{j} \quad (5)$$

Where $\odot$ represents the point-wise product. The advantage of the multi-dimensional attention mechanism is that an optional feature for each word is extracted given the context. We apply the fusion gate to unify the relationship between the words of questions and words of corresponding answers. The formulas are shown as: (Equation(6)(7)):

$$\mathrm{FGate}_{para} = \sigma(W_{p2}Q_{emb} + W_{p3}Q_{ap} + b_{p2}) \quad (6)$$

$$Q_{para} = \mathrm{FGate}_{para}\odot Q_{emb} + (1-\mathrm{FGate}_{para})\odot Q_{ap} \quad (7)$$

where $W_{p1,}W_{p2} \in R^{d\times d}$ and $b_{p2} \in R^{d}$ are learned by the fusion gate. $FGate_{para}, Q_{emb,}Q_{ap}, Q_{para} \in R^{d\times n}$ .In the same way, we can obtain $Q_{orth} \in R^{d\times \mathrm{m}}$.

Then, we train our model with BiGRU to acquire the key information vector between Q_{para} and the Q_{orth}. The detailed acquisition methods are shown as follows:

$$Q_{i}^{para} = BiGRU(Q_{orth_i}) \quad (8)$$

$$Q_{j}^{orth} = BiGRU(Q_{orth_j}) \quad (9)$$

After that, we use two different ways, the max pooling and mean pooling to acquire the vector with a fixed length. Specifically, we compute the max pooling of Q^{para} and A^{orth} respectively, as well as the mean pooling, where $Q^{para} = (Q_1^{para}, Q_2^{para}, \dots, Q_n^{para})$ and $A^{orth} = (Q_1^{orth}, Q_2^{orth}, \dots, Q_m^{orth})$. Then, we concatenate the vectors to get the vector r with fixed length, the details are indicated as follows:

$$r_Q^{mean} = \sum_{i=1}^{n}\frac{Q_i^{para}}{m} \quad (10)$$

$$r_Q^{max} = max_{i=1}^{n} Q_i^{para} \quad (11)$$

$$r_A^{mean} = \sum_{j=1}^{m}\frac{A_j^{orth}}{m} \quad (12)$$

$$r_A^{max} = max_{j=1}^{m} A_j^{orth} \quad (13)$$

$$\mathrm{r} = [r_Q^{mean}; r_Q^{max}; r_A^{mean}; r_A^{max};] \quad (14)$$

Finally, we get the global representation r .The question classification task requires the model to predict whether the given question-answer pair (Q, A, y) is semantically identical or not, hence it is a multi-classification task. We use a multi-layer perceptron (MLP) classifier to predict the label:

$$\mathrm{v} = \tanh(W_r \cdot \mathrm{r} + b_r) \quad (15)$$

$$\hat{y} = \mathrm{softmax}(W_v \cdot \mathrm{v} + b_v) \quad (16)$$

where W_r, b_r, W_v, and b_v are trainable parameters. The entire model is trained end-to-end.

4 Experiments

4.1 DataSet

The dataset in this paper is extracted from Yahoo!Answers. Each category is sorted based on the number of questions. 10 categories are selected with the largest number of questions, and the number of questions in these categories is more than 2000. All the samples are question-answer pairs to ensure that each question has a best answer. Therefore, 2000 question-answer pairs are randomly selected from 10 categories are selected. 70% of sample is the training set, 20% is the testing set, and 10% for validation. The statistics of the corpus is shown in **Table (1-3)**:

Tab 1. Statistics of the original Corpus

Number of questions	Number of answers	Number of best answers	Number of classes	Number of answers per question
200,998	1,848,441	201,075	60	9.405

Tab 2. Statistics of the extracted corpus

Number of questions	**Number of answers**	**Number of classes**
20,000	20,000	10

Tab 3. The experiment corpus

	Training set	**Testing set**	**Validation set**
Number of questions	14,000	4,000	2,000
Number of answers	14,000	4,000	2,000
Average length of question	10.68	10.32	10.45
Average length of answer	44.56	43.18	43.58

We demonstrate the distribution of the percentage of answers with regard to the length of answers (number of words) in Fig 2. From Figure 2 we can easily see that the proportion of answers with less than 50 words is very small, 5%. A vast amount of answers in the dataset contain 100-200 words, and a high proportion contains more than 200 words.

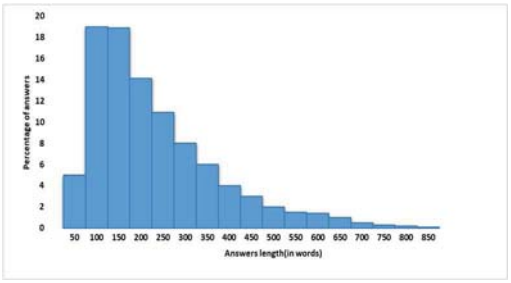

Fig.2 Percentage of answers vs number of words

Fig 3 shows the percentage of different types of questions in our dataset. In essence, the questions of type "How" or "Why" are mostly non-factoid. These questions are open-ended, and require detailed answers. The question denoted as type “What” are generally considered to be factoid which accounts for a large proportion in our dataset, but after analysis, we find that a large part of such questions is also non-factoid. Examples of such questions include “What is the outlook of natural language processing?".

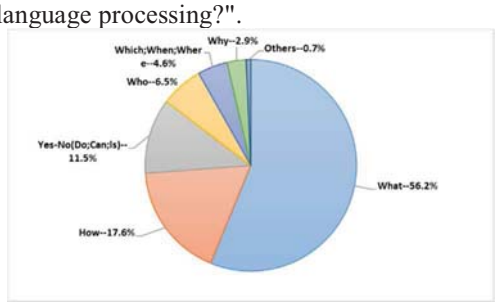

Fig.3 Percentage of questions by type

4.2 Training and hyper parameters

The NLTK toolkit is used in the text preprocessing procedure for each question and corresponding answer, including capitalization conversion, stemming, removal of stop words, et al. The preprocessed dataset is trained to obtain 300-dimensional initialized word vectors (GloVe proposed by Pennington[15]). The vectors for out-of-vocabulary are set to zero. The specific hyper parameters are as follows Tab 4:

Tab 4. hyper parameters

Hyper-parameters	Value
Bi-GRU hidden size	**300**
Batch size	**128**
Learning rate	**0.001**
L2 regularization parameters	**0.001**
Gradient Clipping	**5**
Early stop patience	**10**
Bi-GRU dropout rate	**0.5**
GloVe Embedding-size	**300**
Word2Vec Embedding-size	**300**

4.3 Results and Analysis

We adopt the following three evaluation metrics, F1, accuracy (Acc), and MAP (Mean Average of Precision) to compare the performance of QAAN and the baseline models.

Tab 5. Experimental Methods and Description

No.	Description
（1）	Proposed [19] It used an SVM classifier to incorporate various kinds of features, including topic model based features and word vector representations.
（2）	Proposed [20] It proposed ensemble learning and hierarchical classification method to classify answers
（3）	Proposed [21] It used the memory mechanism to iteratively aggregate more relevant information which is useful to identify the relationship between questions and answers.
（4）	Proposed [22] It combined a supervised model using traditional features and a convolutional neural network to represent the question-answer pair.

Four question classification models are used as comparison models. As shown in **Tab 5**, (1), (2), (3), (4) are the Baseline of QAAN. To be specific, (1) and (2) are top systems from SemEval 2015. Compared to our model which learns various important features automatically, Baseline (1), (2) highly reply on feature engineering. (3) uses thread level information for global inference, and (4) is the top technique for Q&A task from SemEval 2017. Our model QAAN performs the best and significantly outperforms all baseline models ($p < 0.05$ based on student t-test) on Yahoo! Answers datasets, outperforming the best baseline model (7).

Tab 6. MAP, F1 and Acc Performance of the Six Models on Relevant Corpus

Model	MAP	F1	Acc
（1）JAIST	0.7473	0.6587	0.6635
（2）HITSZ-ICRC	0.7304	0.6368	0.6256
（3）BGMN	0.7378	0.6468	0.6537
（4）ECUN	0.7658	0.6875	0.6938
（5）QAAN(our)	**0.7784**	**0.7038**	**0.7126**

The experiment results in **Tab 6** demonstrate that:

(1) JAIST and HITTZ-ICRC are two models with good performance on the data set SemEval2015. We choose these two models as our baseline. The experiment results show that JAIST outperform BGMN model by 0.0095 in terms of MAP, 0.0119 in terms of F1, and 0.0098 in terms of Acc. In general, JAIST shows better performance than BGMN on three evaluation metrics, which proves that not all deep learning models can outperform conventional machine learning models. (Row1 VS Row3)

(2) The ECUN model combining convolutional neural network with supervised learning outperforms BGMN, HITZZ-ICRC, and JAIST in terms of MAP, F1, and Acc. The results demonstrate that the combination of conventional machine learning model and deep learning model can achieve better performance inn question classification task. (Row4 VS Row3、Row2、Row1)

(3) Our QAAN model employs attention mechanism combining the answer information with corresponding answer information. The experiment results show that QAAN outperform other five models in terms of three different evaluation metrics, which prove that corresponding answers can provide important information for question classification. QAAN can successfully learn these crucial resources. (Row6 VS Row5、Row4 、Row3、Row2、Row1)

4.4 Ablation Study

To prove the validity of QAAN, in addition to the six baseline models above, six comparative experiments were conducted to demonstrate the improvement of QAAN. As shown in **Tab 7**.

Tab.7 Experimental Methods and Description

Methodology	Description
(1) without task-specific word embeddings	where word embeddings are initialized with the 300-dimensional GloVe word vectors trained on Wikipedia 2014 and Gigaword 5.
(2) without character embeddings	where wordlevel embeddings are only composed of 600-dimensional GloVe word vectors trained on the domain-specific unannotated corpus
(3) question only	where only question is used as question representation.
(4) answer only	where only answer body is used as question representation.
(5) similarity only	where the parallel component alone is used in question - answer interaction.
(6) disparity only	where the orthogonal component alone is used in question - answer interaction.

Tab 8. MAP, F1 and Acc Performance of the Six Models on Relevant Corpus

Model	MAP	F1	Acc
（1）without task-specific word embeddings	0.7236	0.6357	0.6838
（2）without character embeddings	0.7286	0.6374	0.6826
（3）question only	0.6756	0.6175	0.6365
（4）answer only	0.6852	0.6248	0.6475
（5）similarity only	0.7648	0.6985	0.7068
（6）disparity only	0.7538	0.6849	0.6988
（7）QAAN(our)	**0.7784**	**0.7038**	**0.7126**

The experiment results show in Table 8 show that:

(1) Comparing Row1 VS Row2, we can see that the task-specific word embedding trained on the corpus show better

performance, indicating that the questions in the community are distinct from text in News and web pages. This is because there are typos, syntactic errors, abbreviations due to the lack of professional background, which is a challenge for question classification.

(2) Comparing Row7 VS Row1 Row2, QAAN shows better performances both with special-trained word embedding and character embedding, which prove that QAAN can also solve the out-of-vocabulary issue.

(3) Comparing Row3 VS Row 4, paired answers provide more useful information than questions themselves, and the meanings of questions are represented better by the paired answers. This is mainly because, in question answering community, most of the questions is generally ambiguity. But the respondents, also known as, the providers of answers, have strong domain expertise. The answers are often more informative and represent the question better.

(4) From the comparison of Row5 VS Row6, we can see that the parallel components show better performance than vertical components in question classification task. This is because the parallel component can more comprehensively use the sequence information of texts to improve the classification performance.

5 Conclusion

It is of significance to study the question classification in community question answer since question classification is the basis of many tasks, such as question retrieval, question recommendation, expert discovery, et al. A novel multi-dimensional Question Answer Attention Network, QAAN is proposed in this paper. QAAN uses the linguistic knowledge characteristics comprehensively to explore the question-corresponding answer relationship. The comparison experiment results show that the combination of question and paired answer information can effectively solve the problem of question classification of Community Question Answering. The only pitfall is that QAAN model is not completely satisfactory in terms of computing speed. In future work, we will try to optimize the computing speed. Future more, we will try different architectures for question classification, such as capsule networks, to pursue exceptional performance.

Acknowledgment: This work is supported by National Natural Science Foundation of China (No. 61572469).

REFERENCES

[1] Zhang X, Li S, Sha L, et al. Attentive Interactive Neural Networks for Answer Selection in Community Question Answering.[C]. national conference on artificial intelligence, 2017: 3525-3531.

[2] Lecun Y, Bengio Y, Hinton G E, et al. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[3] Adhikari A, Ram A, Tang R, et al. Rethinking Complex Neural Network Architectures for Document Classification[C]. north american chapter of the association for computational linguistics, 2019.

[4] Merity S, Keskar N S, Socher R, et al. Regularizing and Optimizing LSTM Language Models[J]. international conference on learning representations, 2018.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[J]. neural information processing systems, 2017: 5998-6008.

[6] Mohammed S, Shi P, Lin J J, et al. STRONG BASELINES FOR SIMPLE QUESTION ANSWERING OVER KNOWLEDGE GRAPHS WITH AND WITHOUT NEURAL NETWORKS[J]. north american chapter of the association for computational linguistics, 2018: 291-296.

[7] Sculley D, Snoek J, Wiltschko A B, et al. Winner's Curse? On Pace, Progress, and Empirical Rigor[C]. international conference on learning representations, 2018.

[8] Lipton Z C, Steinhardt J. Troubling Trends in Machine Learning Scholarship[J]. ACM Queue, 2019, 17(1).

[9] Yang P, Sun X, Li W, et al. SGM: Sequence Generation Model for Multi-label Classification[J]. international conference on computational linguistics, 2018: 3915-3926.

[10] Kim Y. Convolutional Neural Networks for Sentence Classification[J]. empirical methods in natural language processing, 2014: 1746-1751.

[11] Kalchbrenner N, Grefenstette E, Blunsom P, et al. A Convolutional Neural Network for Modelling Sentences[J]. meeting of the association for computational linguistics, 2014: 655-665.

[12] Le P, Zuidema W H. The Forest Convolutional Network: Compositional Distributional Semantics with a Neural Chart and without Binarization[C]. empirical methods in natural language processing, 2015: 1155-1164.

[13] Mou L, Peng H, Li G, et al. Discriminative Neural Sentence Modeling by Tree-Based Convolution[J]. empirical methods in natural language processing, 2015: 2315-2325.

[14] Komninos A, Manandhar S. Dependency Based Embeddings for Sentence Classification Tasks[C]. north american chapter of the association for computational linguistics, 2016: 1490-1500.

[15] Pennington J, Socher R, Manning C D, et al. Glove: Global Vectors for Word Representation[C]. empirical methods in natural language processing, 2014: 1532-1543.

[16] Kim Y, Jernite Y, Sontag D, et al. Character-aware neural language models[C]. national conference on artificial intelligence, 2016: 2741-2749.

[17] Wang Z, Mi H, Ittycheriah A, et al. Sentence Similarity Learning by Lexical Decomposition and Composition[C]. international conference on computational linguistics, 2016: 1340-1349.

[18] Shen T, Zhou T, Long G, et al. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding[C]. national conference on artificial intelligence, 2018: 5446-5455.

[19] Tran Q H, Tran V D, Vu T T, et al. JAIST: Combining multiple features for Answer Selection in Community Question Answering[C]. north american chapter of the association for computational linguistics, 2015: 215-219.

[20] Yongshuai Hou, Cong Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Qingcai Chen. 2015. HITSZICRC: exploiting classification approach for answer selection in community question answering. In Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015. pages 196– 202.

[21] Wei Wu, Houfeng Wang, and Sujian Li. 2017. Bidirectional gated memory networks for answer selection. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. LNAI 10565, Springer. pages 251–262.

[22] GuoShun Wu, Yixuan Sheng, Man Lan, and Yuanbin Wu. 2017a. ECNU at semeval-2017 task 3: Using traditional and deep learning methods to address community question answering task. In Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017. pages 365–369.

[23] Pappas N, Popescubelis A. Multilingual Hierarchical Attention Networks for Document Classification[C]. international joint conference on natural language processing, 2017: 1015-1025.