A Dual Attentive Neural Network Framework with Community Metadata for Answer Selection

Zhiqiang Liu[1,2], Mengzhang Li[2], Tianyu Bai[2], Rui Yan[1] and Yan Zhang[1]

[1]School of Electronics Engineering and Computer Science, Peking University, China
[2]Academy for Advanced Interdisciplinary Studies, Peking University, China
{lucien,mcmong,tybai,ruiyan,zhyzhy001}@pku.edu.cn

Abstract. Nowadays the community-based question answering (cQA) sites become popular Web service, which have accumulated millions of questions and their associated answers over time. Thus, the answer selection component plays an important role in a cQA system, which ranks the relevant answers to the given question. With the development of this area, problems of noise prevalence and data sparsity become more tough. In our paper, we consider the task of answer selection from two aspects including deep semantic matching and user community metadata representation. We propose a novel dual attentive neural network framework (DANN) to embed question topics and user network structures for answer selection. The representation of questions and answers are first learned by convolutional neural networks (CNNs). Then the DANN learns interactions of questions and answers, which is guided via user network structures and semantic matching of question topics with double attention. We evaluate the performance of our method on the well-known question answering site Stack exchange. The experiments show that our framework outperforms other state-of-the-art solutions to the problem.

1 Introduction

Community-based question answering(cQA) is an Internet-based web service which enables users to post their questions on a cQA website, which might be answered by other users later. Some cQA sites are popular such as Yahoo! Answers[1] and Quora[2], which have accumulated millions of questions and answers pairs over time [2]. However, there are some challenges needing to be overcome in cQA field. One is the problems of redundancy and noise prevalent which were usually ignored by previous research. The other is the bottleneck of data sparsity: previous research relies on the content similarity between questions and answers, therefore it suffers from the sparsity of cQA data.

Most of the existing works consider cQA problem as a text matching task. In recent years, many researchers have proposed various deep learning methods that automatically select answers. These methods usually learn representations of two pieces of texts using neural networks, e.g. convolutional neural networks (CNNs) [12] or recurrent neural networks (RNNs) [17]. Based on the representations, a function is given to calculate the matching score. Instead of learning a global representation of the texts, some

[1] https://answers.yahoo.com
[2] https://www.quora.com

researchers have proposed models that learn the interaction information of the representations and have achieved better results [5]. Although previous question answering methods have achieved promising performance, they mainly focus on the deep semantic matching models of the problem, which is not suitable for redundant and noisy text and ignores the importance of user community metadata in cQA sites.

On the other hand, with the prevalence of online social networks in cQA sites [2], some researchers adopt a random walk method to exploit the rich social information from heterogeneous social networks, aiming to solve the problem of sparsity in cQA tasks. They combine it with a deep recurrent neural network which excellently models the textual contents of questions and answers. However, this method is still complicated during integrating community network information and text information. Its effect is limited as a result that additional social network data is necessary in an effective way [20].

In this paper, we formulate the problem of community-based question answering from two aspects including deep semantic relevance of question-answer pairs with question topic attention and the user expertise authority with community metadata attention in order to solve the problems of the data sparsity and noise prevalent. The major contribution of this paper is listed below:

- Different from previous studies, we formulate the problem of answer selection from two aspects to solve the problems of the data sparsity and noise prevalent. That is, we learn the ranking function based on both the deep semantic matching and user community metadata in cQA sites.
- We propose a novel dual attentive neural network framework named as DANN, which yields better performance than other state-of-the-art methods. This framework can also be used in other tasks.

2 Related Work

2.1 Deep Semantic Matching

Recently, some works are proposed on applications of deep neural networks of cQA tasks, aiming to solve a general sentence matching problem. In detail, these methods usually learn representations of two pieces of texts using neural networks, e.g. convolutional neural networks (CNNs) or recurrent neural networks (RNNs). In [14], the authors calculate a similarity matrix for each pair of questions and answers to contain the lexical and sequential information and then use a deep convolutional neural network (CNN) to estimate the suitable answer probability. Different from the classical convolutional neural network used in [14], some researchers [12] introduce a dynamic convolutional neural network [8] to encode the sentences of questions and answers in semantic space and model their interactions with tensors on the top layer. Besides the CNNs, another kind of neural networks has been successfully applied in textual content analysis. In [9], recurrent neural network is employed to represent each sentence or document by a dense vector which is trained to predict words in the document and in [15], a multi-layer RNN is used to map the input sentence into a fixed dimensional vector.

2.2 Network Representation Learning

On the other hand, aiming to improve the quality of QA tasks, additional information is introduced by some methods. In [6], a recursive neural network with sentence dependency-tree is utilized for simulating question answering tasks. In [21], the authors propose a concept base of world knowledge of Wikipeida and then change these semantic relations to optimize the question retrieval task. Recently, network representation learning has been proposed as a critical technique for community network analysis tasks. For example, DeepWalk [11] performs random walks over networks to learn network embeddings. LINE [16] optimizes the joint and conditional probabilities of edges in large-scale networks to learn vertex representations. Node2vec [3] modifies the random walk strategy in DeepWalk into biased random walks to explore the network structure more efficiently.

With the prevalence of online social networks in cQA sites, some researchers adopt the Network Representation Learning(NRL) methods to exploit the rich social information from heterogeneous social networks to solve the sparsity problem in cQA tasks and combine it with a deep neural network. Some works are proposed on exploiting the social information for QA tasks. In [2], the authors develop a graph-regularized matrix completion algorithm for inferring the user model and thus improve the performance of expert finding in cQA systems. The cross-domain social information integration is considered in [7]. They represent a social network as a star-structured hybrid graph centered on a social domain and propose a hybrid random walk method which incorporates cross-domain social information to predict user-item links in a target domain. We think these methods does make sense.

2.3 Attention Mechanism

Attention-based deep learning systems are studied in NLP after its success in computer vision and speech recognition, and mainly rely on recurrent neural network for end-to-end encoder-decoder system for tasks such as machine translation [1] and text reconstruction [13]. In [18], they take the lead in exploring attention mechanism in CNN for NLP tasks and propose multi-level attention convolutional neural network for modeling sentence pairs. After that, some researchers adopt attention mechanism to measure the importance of each segment and combine the interactions to obtain fixed-length representations for questions and answers[19]. Similar to the spirit of these studies, we propose this dual attentive neural network framework with community metadata to solve answer selection in cQA sites.

3 Model

In this section, we introduce a dual attentive neural network model. Firstly, we consider the problem of community-based question answering as a high-quality answer selecting task, which is based on the question-answer pairs according to deep semantic relevance with the question topics and user community metadata. Then we implement this model and experiments on our dataset from a real world cQA site Stack exchange show that the model indeed achieves better performance than other cQA methods.

3.1 The problem

Before presenting the problem, we first introduce some basic notions and terminologies. Since the questions and answers in cQA sites are sentences, which are the sequential data with variant length, we then encode their contents into fixed length feature vectors for abstractive representation. Different from [5], which use convolution to learn interaction directly, we split the process of representation learning and interaction learning. In our work, we use GoogleNews[3] corpus to pre-train representations.

3.2 User Community Metadata

An overview of the cQA heterogeneous network is illustrated in Figure 1. The heterogeneous network is composed of three parts, i.e. **User** $U = \{u_1, u_2, ..., u_k\}$ consisting answerer and asker information such as user badges, user reputations, **Question** $Q = \{q_1, q_2, ..., q_n\}$ and **Answer** $A = \{a_1, a_2, ..., a_m\}$. Among them, the relationship between users can be either following relation or user badges relation.

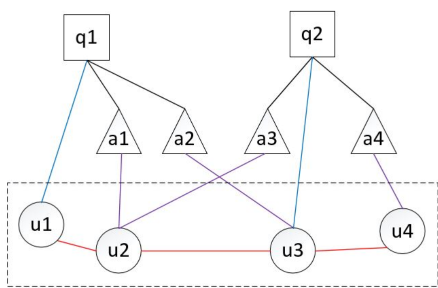

Fig. 1. The heterogeneous cQA network. The network contains three types of nodes: User, Question and Answer. And the edges include Asker-Question, Answerer-Answer, Question-Answer and User-User.

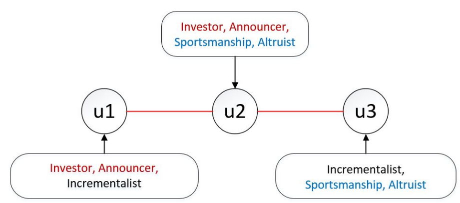

Fig. 2. The user network. The network nodes are users and the $\lambda = 2$ in this instance. (Red and blue fonts represent the common badges of the users.)

In our work, we denote the user network by $G = (V, E)$, where the set of nodes is Users **U**, the set of edges is user relations including *Asker-Question, Answerer-Answer, Question-Answer and User-User* in cQA sites. We define that there is an edge between two users in G if there are more than λ common *badges* between them. Then our method learn network embeddings of users via *DeepWalk* [11]. In this way, user badges are contained at the users set as **community metadata** in the cQA system, which will guide deep semantic learning of question-answer pairs in DANN. It is also applicable to create user networks in other ways. In this paper, we aim to propose a framework, which make the most of user information to guide deep semantic learning with attention. The reason why we created the network in this way is the experiment's conclusion. In our opinion, users with similar badges have similar characteristics and can be referred

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

in the selection of answers. An overview of the user network is illustrated in Figure 2, where the value of λ is 2 in order to show it intuitively.

3.3 Model illustration

We now introduce our model that is based on CNN model. It consists of two weight-sharing CNNs, one to process question sentences and the other to process answer sentences. In the semantic matching process, we introduce the double attention mechanism including word level attention and sentence level attention (Figure 3). We refer to this architecture as DANN. There are four layers in DANN: input layer with attention, convolution layer with attention, pooling layer and output layer.

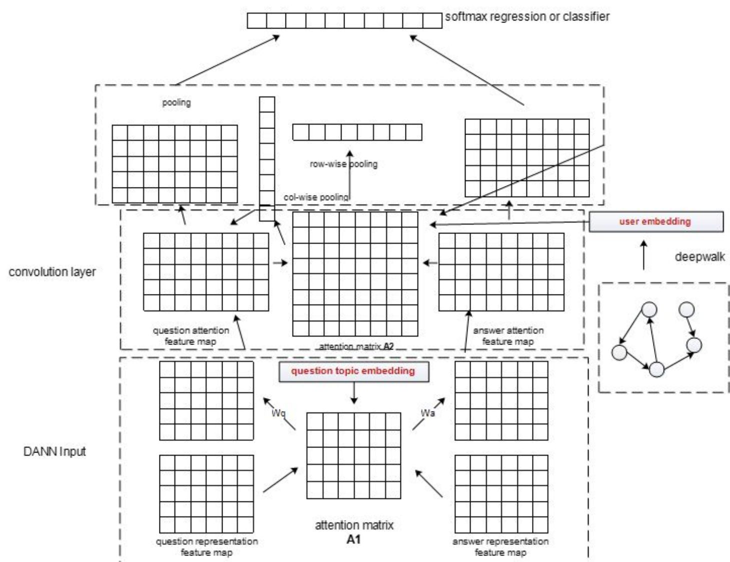

Fig. 3. The structure of DANN framework.

Input layer with attention: The two input sentences have no more than n words, respectively in it. Each word is represented as a d-dimensional precomputed *word2vec* [9, 10] embedding, d = 300. As a result, each sentence is represented as a feature map of dimension $d \times s$.

In detail, the DANN method employs an attention feature matrix $A^{(1)}$ to influence convolution as the first attention. Attention features are intended to weight those units of s^q more highly in convolution that are relevant to a unit of s^a, and weight those units of s^a more highly in convolution that are relevant to a unit of s^q. In addition, they are relevant to the topics of question. Each column is the representation of a word. We first describe the attention feature matrix $A^{(1)}$ informally. $A^{(1)}$ is generated by matching words of the question representation feature map with words of the answer representation feature map such that the attention values of row i in $A^{(1)}$ denote the attention distribution of the i-th word of s^q with respect to s^a, and the attention values of column j in $A^{(1)}$ denote the attention distribution of the j-th word of s^a with respect to s^q. $A^{(1)}$ can be viewed as a new feature map of s^q in row direction because each row is a new feature vector of a word in s^q. Thus, it is reasonable to combine this new feature map with the representation of feature maps and use both as input to the convolution operation. We achieve this by transforming $A^{(1)}$ into the two matrices that have the same format as the representation of feature maps in Figure 3. As a result, the new input of convolution has two feature maps for each sentence. Our motivation is that the attention feature map will guide the convolution to learn sentence representations with question topics.

More formally, let $q = \{e_i^q | i \in [1, s_q]\}$ and $a = \{e_i^a | i \in [1, s_a]\}$ be the representation feature map of question sentences and answer sentences. And let $t = \{e_i^t | i \in [1, s_t]\}$ be the embeddings of question topics. Then we define the attention matrix $A^{(1)} \in R^{s \times s}$ as follows:

$$A_{i,j}^{(1)} = kernel(e_i^q, e_i^a, e_i^t) \tag{1}$$

The function kernel can be defined:

$$max\{\frac{1}{1 + |e_i^q - e_j^a|}, \frac{1}{1 + |e_i^q - e_i^t|}, \frac{1}{1 + |e_j^a - e_i^t|}\} \tag{2}$$

where $|\cdot|$ is distance function. This kernel function can measure the matching scores in the words level of questions and answers based on the topics of the questions. As a result, given attention matrix $\mathbf{A}^{(1)}$, we generate the attention feature map E for s_i as follows:

$$E^q = W^q \cdot A^T \tag{3}$$

$$E^a = W^a \cdot A \tag{4}$$

The weight matrices $W^q \in R^{d \times s}$, $W^a \in R^{d \times s}$ are parameters of the model to be learned in training.

Convolution layer with attention: Let $\{e_1, e_2, ..., e_s\}$ be the words of a sentence and $c_i \in R^{w \cdot d_0}$, $0 < i < s + w$, the concatenated embeddings of $e_{i-w+1}, e_{i-w}, ..., e_i$ where embeddings for e_i (where $i < 1$ and $i > s$), are set to zero. We then generate the representation $h_i \in R^{d_1}$ for the phrase $e_{i-w+1}, e_{i-w}, ..., e_i$ using the convolution weights $\mathbf{W} \in R^{d_1 \times wd_0}$ as follows:

$$h_i = tanh(\mathbf{W} \cdot c_i + b) \tag{5}$$

where $b \in R^{d_1}$ is the bias. We use wide convolution by applying the convolution weights W to words e_i (where $i < 1$ and $i > s$). This ensures that each word v_i can be detected by all weights in W.

The second level attention of DANN computes attention weights on the output of convolution with the aim of re-weighting this convolution output. In addition, we define asker embedding as u^q and answerer embedding as u^a, which are learnt via the user network in Figure 2.

Let A be the attention matrix and α is defined as follows, where $f(a,b) = a^T b$:

$$\alpha_i^q = \frac{exp(f(A[i,:], u^q))}{\sum_k exp(f(A[k,:], u^q))} \tag{6}$$

$$\alpha_j^a = \frac{exp(f(A[:,j], u^a))}{\sum_k exp(f(A[:,k], u^a))} \tag{7}$$

Attention Pooling layer: The pooling layer, including min pooling, max pooling and average pooling, is commonly used to extract robust features from convolution. In this paper, we use attention pooling. According to α^q and α^a, we can obtain attention pooling result R, which will be the representation of question-answer pairs. In detail, R is defined as follows:

$$r_i^q = \sum_{k=i:i+w} \alpha_k^q \cdot h_k^q (i = 1, ..., s_q) \tag{8}$$

$$r_j^a = \sum_{k=j:j+w} \alpha_k^a \cdot h_k^a (j = 1, ..., s_a) \tag{9}$$

$$R = [r_{avg}^q, r_{avg}^a] \tag{10}$$

Output layer: The last layer is an output layer, chosen according to the task; e.g., for binary classification tasks, this layer is logistic regression (see Figure 3). The simplest way to train the model is to use a Softmax classifier with cross entropy as the loss function.

$$score(q,a) = softmax(W^T \cdot R + b) \tag{11}$$

3.4 Attention Calculation

In our model, two distinctive attentive methods are introduced. Question topic embedding attention matrix $A^{(1)}$ and user embedding attention matrix $A^{(2)}$. In actual Q &A forum such as Stack Overflow and Quora, many segments are redundant. Then refine the essence and discard the waste. By introducing prevalent attention mechanism at neural network, we aim to calculate the importance of each text segment. The numerical value of attention model is trained together with the whole neural network. Five kinds of information are computed the attention, including segment representation, word representation of pairwise question and answer as interaction with other segment, question topic and user community metadata, and what's more, we introduced the network representation by *DeepWalk*. Basically, the second-level attention α_j^q and α_j^a of each segment in a question or answer is calculated as Formula 6 and Formula 7. We aim to propose this dual attentive framework and you can define other styles using different additional information based on our framework.

4 Experiment

4.1 Dataset

We evaluate DANN on Stack exchange dataset. This dataset consists of real data from the community-created Stack Exchange forums. The whole dataset consists and over 133 question answering forums and the Stack Overflow is the most popular forum among them. In our experiment, we choose two forums history data to validate our framework against some baselines. The themes of these two forums are “English”, “Academia” and the “English” forum is a smaller dataset and the “Academia” forum is a larger dataset. We present the detail of these two forums data in Table 1. There are three parts: user, question and answer. Given a question from a certain topic, the participant systems rank the comments according to their relevance associated with the question. Each answer has its own score and is labeled with one of two labels “Good” or “Not good”. Table 1 demonstrates statistics of the datasets.

Table 1. Statistics of the stack exchange dataset

Forums	English	Academia
Question	5000	12052
Answer	13461	31046
User	6472	5875
Avg len of ques.	65.87	84.33
Avg len of ans.	255.54	326.75

As we can see, questions in two forums received distinct proportion of answers, and the average length of questions and answers vary from each other, which can demonstrate the versatility of our model. In addition, we choose CQADupstack[4] [4] scripts to process these datasets. We then split the datasets into training set, validation set and testing set without overlapping in our experiments. We fix the validation set as 10% of the total data to tune the hyper parameters and the size of testing set is 30%.

4.2 Baselines

We compare our model with other methods of answer selection in cQA sites. For example *AvgWord* method utilizes average word embedding to achieve text vectors for every question and answer. Then they are predicted by some method such as random forest and svm. It is reasonable that DANN is more useful than those traditional methods for its network representation and the attention mechanism. We divide our model into two related methods DANN with and without the attention mechanism. Then they are compared with other baseline methods.

– **AvgWord + Random Forest**: this method first achieves the sum of every text word embeddings and divide number of words. The results can be considered as representation of sentences. Then some regression methods are utilized such as random

[4] http://nlp.cis.unimelb.edu.au/resources/CQAdupstack/

forests and svm to compare related sentence representations of questions and answers.

- **BOW**: Bag-of-words (BOW) is a classical representation for natural language processing tasks. In our experiments, we represent the questions and answers by BOW feature vectors and then calculate the relevant score to rank the candidate answers for each question.
- **BM25**: Okapi BM25 (BM stands for Best Matching) is a ranking function used by search engines to rank matching documents according to their relevance to a given search query.
- **CNTN**[12]: this method introduces a convolutional neural tensor network to integrate sentence modeling and semantic matching information, which can hardly be captured by convolutional and pooling layers.
- **LSTM + Multi-Layer-Perceptron**: it learns the sentence pairs embedding via LSTM and use MLP predict the matching score of question-answer pairs.
- **AI-CNN**[19]: this model distinguishes different text segments differently and designs an attentive interactive neural network (AI-NN) to focus on those text segments useful to answer selection. The representations of questions and answers are first learned by convolutional neural networks (CNNs). No user network structure is considered.

4.3 Evaluation Criteria

Two ranking metrics are used for evaluation, mean reciprocal rank(MRR) and mean average precision(MAP). The definitions are given below and we utilize them as evaluation metrics.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{12}$$

$|Q|$denotes the total number of questions in the evaluation set. $rank_i$ denotes the position of the first correct answer in the generated answer set C_i for the i^{th} question Q_i. If C_i doesn't overlap with the golden answers A_i for Q_i, $\frac{1}{rank_i}$ is set to 0.

$$MAP = \frac{1}{|Q|} \sum_{i=1}^{|Q|} AveP(C_i, A_i) \tag{13}$$

$AveP(C_i, A_i) = \frac{\sum_{k=1}^{n} (P(k) \cdot rel(k))}{min(m,n)}$ denotes the average precision. k is the rank in the sequence of retrieved answer sentences. m is the number of correct answer sentences. n is the number of retrieved answer sentences. If $min(m, n)$ is 0, $AveP(C, A)$ is set to 0. $P(k)$ is the precision at cut-off k in the list. $rel(k)$ is an indictor function equaling 1 if the item at rank k is an answer sentence, and 0 otherwise.

4.4 Experimental settings

The questions and answers in English are tokenized and lemmatized using NLTK[5]. In addition, we do some pre-processing. We remove stop words, delete identical questions and put the question titles and bodies into one single text string. The GoogleNews corpus is utilized to pre-train word embeddings in out experiment. Adagrad is used to update parameters. During the model training, we try to compare our model by varying value of window size and parameter λ in Figure 4. Finally, we observe that our method achieves the best performance when the value of window size is set to 4 and the value of parameter λ is set to 10. The dimension of question topics, words and user embeddings is 300 and initial learning rate is 0.01. The details are known in Table 2.

Table 2. Experiment parameters setting

Papameters	Value
λ	10
Learning rate	0.01
Window size	4
Batch size	64
Vector dimension	300
User embedding method	Deepwalk
Word2vec	GoogleNews

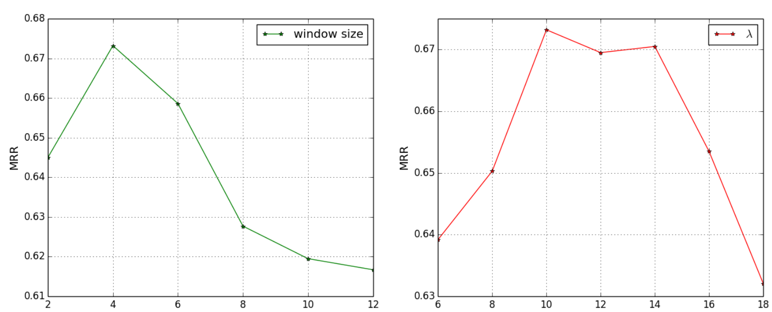

Fig. 4. Effect of parameter window size, λ on MRR using the English cQA forum.

4.5 Experimental results

Table 3 and Table 4 present the results of our model and the baseline methods. The experiments show that our framework can outperform other state-of-the-art solutions to

[5] http://www.nltk.org

the problem. We can see that DANN outperforms DANN(w/o attention), demonstrating that the multi-level attention is helpful for answer selection in cQA cites. And the CNTN and AI-CNN model behave better than other baselines. Compared with these method, our model can obtain best result. Particularly, the model in the larger Academia forum dataset can shows a greater advantage than other baselines because fewer users and more edges can learning effective embeddings in user network.

Table 3. Experiment results on English forum

Method	MAP	MRR
AvgWord + RF	0.4812	0.5240
BOW	0.5157	0.5361
BM25	0.5676	0.5742
LSTM + MLP	0.5907	0.6163
CNTN[12]	0.6369	0.6403
AI-CNN[19]	0.6425	0.6591
DANN(w/o attention)	0.6322	0.6584
DANN	**0.6557**	**0.6732**

Table 4. Experiment results on Academia forum

Method	MAP	MRR
AvgWord + RF	0.5106	0.5311
BOW	0.5425	0.5507
BM25	0.5899	0.6102
LSTM + MLP	0.6692	0.6920
CNTN[12]	0.7411	0.7667
AI-CNN[19]	0.7795	0.7931
DANN(w/o attention)	0.7524	0.7652
DANN	**0.7856**	**0.8095**

5 Conclusion and Future Work

In this paper, we construct a novel dual attentive neural network framework(DANN) to achieve answer selection in the cQA field. The DANN with community metadata learns network structures of users and uses it to guide interaction learning of texts by the attention mechanism which can avoid redundant and noisy text. This double attention mechanism measure segments to select answers. The extensive experiments demonstrate that our method can achieve better performance than several state-of-the-art solutions to the problem. In the future, we will explore the following directions:

– We will extend this model to other fields such as question retrieval, expert finding in cQA sites.
– It is of interest to explore other QA or user additional information and such data can be used to enhance the performance.

6 Acknowledgment

This work is supported by NSFC under Grant No.61532001 and No.61370054. We thank the three anonymous reviewers for their valuable comments.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

2. Fang, H., Wu, F., Zhao, Z., Duan, X., Zhuang, Y., Ester, M.: Community-based question answering via heterogeneous social network learning. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
3. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
4. Hoogeveen, D., Verspoor, K.M., Baldwin, T.: Cqadupstack: A benchmark data set for community question-answering research. In: Proceedings of the 20th Australasian Document Computing Symposium. p. 3. ACM (2015)
5. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in neural information processing systems. pp. 2042–2050 (2014)
6. Iyyer, M., Boyd-Graber, J.L., Claudino, L.M.B., Socher, R., Daumé III, H.: A neural network for factoid question answering over paragraphs. In: EMNLP. pp. 633–644 (2014)
7. Jiang, M., Cui, P., Chen, X., Wang, F., Zhu, W., Yang, S.: Social recommendation with cross-domain transferable knowledge. IEEE Transactions on Knowledge and Data Engineering 27(11), 3084–3097 (2015)
8. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
9. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). pp. 1188–1196 (2014)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
11. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
12. Qiu, X., Huang, X.: Convolutional neural tensor network architecture for community-based question answering. In: IJCAI. pp. 1305–1311 (2015)
13. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
14. Shen, Y., Rong, W., Sun, Z., Ouyang, Y., Xiong, Z.: Question/answer matching for cqa system via combining lexical and sequential information. In: AAAI. pp. 275–281 (2015)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
16. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
17. Wang, D., Nyberg, E.: A long short-term memory model for answer sentence selection in question answering. In: ACL (2). pp. 707–712 (2015)
18. Yin, W., Schütze, H., Xiang, B., Zhou, B.: Abcnn: Attention-based convolutional neural network for modeling sentence pairs. arXiv preprint arXiv:1512.05193 (2015)
19. Zhang, X., Li, S., Sha, L., Wang, H.: Attentive interactive neural networks for answer selection in community question answering. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
20. Zhao, Z., Lu, H., Zheng, V.W., Cai, D., He, X., Zhuang, Y.: Community-based question answering via asymmetric multi-faceted ranking network learning. In: AAAI. pp. 3532–3539 (2017)
21. Zhou, G., Liu, Y., Liu, F., Zeng, D., Zhao, J.: Improving question retrieval in community question answering using world knowledge. In: IJCAI. vol. 13, pp. 2239–2245 (2013)