SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Natural Language Processing and Social Computing

November 2017, Vol. 60 110102:1–110102:9
doi: 10.1007/s11432-016-9197-0

Convolutional neural networks for expert recommendation in community question answering

Jian WANG*, Jiqing SUN, Hongfei LIN*, Hualei DONG & Shaowu ZHANG

School of Computer Science and Technology, Dalian University of Technology, Dalian 116024*, China*

Received June 5, 2017; accepted July 17, 2017; published online October 13, 2017

Abstract Community Question Answering (CQA) is becoming an increasingly important web service for people to search for expertise and to share their own. With lots of questions being solved, CQA have built a massive, freely accessible knowledge repository, which can provide valuable information for the broader society rather than just satisfy the question askers. It is critically important for CQA services to get high quality answers in order to maximize the benefit of this process. However, people are considered as experts only in their own specialized areas. This paper is concerned with the problem of expert recommendation for a newly posed question, which will reduce the questioner's waiting time and improve the quality of the answer, so as to improve the satisfaction of the whole community. We propose an approach based on convolutional neural networks (CNN) to resolve this issue. Experimental analysis over a large real-world dataset from Stack Overflow demonstrates that our approach achieves a significant improvement over several baseline methods.

Keywords community question answering, expert recommendation, convolutional neural networks, classification-based method, expert modeling

Citation Wang J, Sun J Q, Lin H F, et al. Convolutional neural networks for expert recommendation in community question answering. Sci China Inf Sci, 2017, 60(11): 110102, doi: 10.1007/s11432-016-9197-0

1 Introduction

Community Question Answering (CQA) services are intended to provide an open platform for knowledge and experience sharing. A large number of users utilize these services, making CQA sites increasingly popular. For example, Stack Overflow, Yahoo! Answer and Baidu Knows have attracted lots of people and the newly posted question in those sites are increasing day by day.

On CQA sites, one of the main problems is low participation. Only a small number of users are responsible for answering the majority of questions. There are two reasons for low participation. One is that a majority of users are not willing to answer questions and the other is that users willing to answer questions cannot find the new questions of interest to them. Therefore, it is necessary to route questions to experts who have a high probability of giving the best answer. Such research can reduce user waiting time and make valuable contributions to receiving high quality answers.

In traditional content-oriented models, co-occurrence information about the user mentioned with the question words in the same context is assumed to be evidence of expertise. The stronger the association

* Corresponding author (email: wangjian@dlut.edu.cn, hflin@dlut.edu.cn)

between a user and a question, the more likely it is that the user is an expert on that question. However, the questions in CQA are usually too short to get enough information for dealing with the word-matching between posted questions and user profiles. That is, there are semantic gaps between questions and user profiles. To address this problem, we used word embedding to represent the questions and the user profile, which is a well-known method for capturing meaningful syntactic and semantic information.

We recast expert recommendation as a classification problem and treat the best answerer of a question as positive data and others as negative data. Traditional sentence classification algorithms (e.g., a support vector machine: SVM), extract a rich set of hand-designed features form sentence. Deep learning is a representation learning method that can automatically learn internal structural features of complex problems and improve performance. For this reason, we used convolutional neural networks (CNN) to predict which users are more likely to give the best answer for a newly posted question. We conducted expert recommendation on a large real-world dataset from Stack Overflow. Experimental results show that the proposed model shows superior performance over all baseline methods. The main contributions of this work are as follows:

- We selected candidate experts to be classified and build a profile for each candidate expert.
- We used word embedding to represent the question and user profiles and then used convolutional neural networks to predict which users were more likely to give the best answer for the newly posted question.
- We reported the empirical results on a large real-world dataset from Stack Overflow, and showed that the approach outperforms several other methods for profile-based expert finding.

2 Related work

The recommendation of experts is commonly modeled in terms of associations between query topics and people: the stronger the association between a person and a topic, the likelier it is that the person is an expert on that topic. A number of models have been developed to capture these associations between query terms and expert candidates [1]. The content-oriented expert recommendation methods are based on the content of the questions. In an early study on expert-finding, information retrieval techniques were adopted to identify the group of experts most likely to provide answers to a given question [2]. Li et al. [3] investigated incorporation of the question category to route questions to potential answerers. Zhou et al. [4] proposed a method that joint relevance and answer quality learning for question routing. Riahi et al. [5] adopted a Segmented Topic Model to predict who would provide the best answer. The results showed that statistical topic models can be regarded as a suitable alternative for expert recommendation. Mandal et al. [6] introduced a method using the theme in query likelihood language model for expert finding. In this case, the theme of the query was based on the parts of speech (POS) of the words in the query. Yang et al. [7] proposed a topic expertise model which joint topics and expertise by integrating textual content model and link structure analysis.

There are also some methods based on matrix completion for expert recommendation. In one study [8], the user's expertise was indicated by tags. The expertise of the users was translated to (user, tag) scores, which rely heavily on the quality of the tags. In another study [9], the user's social networks were employed to infer the expertise of user, so as to improve the performance of expert finding, but this is not a generalized method useble on all CQA sites.

3 Methodology

In this section, we describe the convolutional neural networks model we considered in detail. The questions and user profiles are represented as word embedding, and these representations are further used as features in the convolutional neural networks.

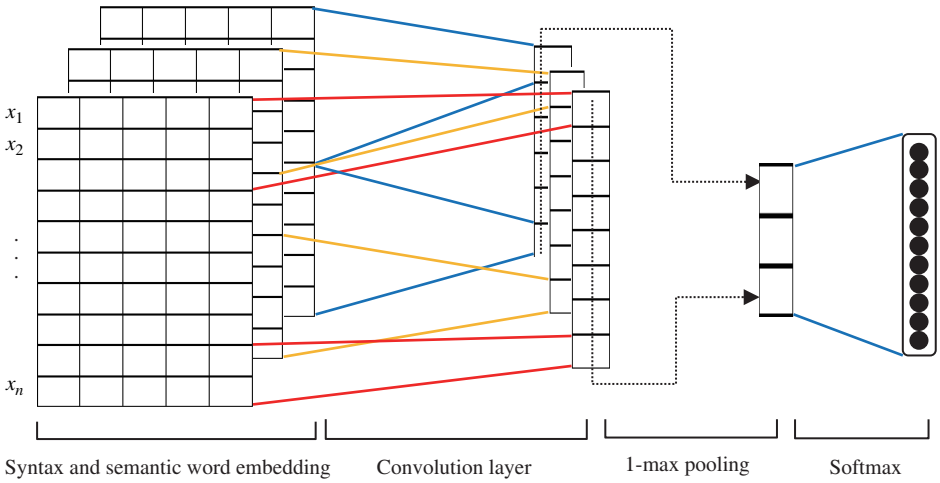

Figure 1 (Color online) The architecture of CNN used for expert recommendation.

3.1 Distributed representations of words

One-hot representation is the most intuitive and common words vector representations method in natural language processing (NLP) which may lead to the curse of dimensionality [10]. Recently, the rapid development of pre-trained word embedding showed its capacity of capturing meaningful syntactic and semantic information [11]. In this work, we performed unsupervised learning of word embedding using the word2vec tool[1)], which represents terms by means of dense, real-valued vectors [12]. Learning task-specific word embedding is known as a practical method for improving the experimental effect. We used 2010 and 2011 snapshots of the Stack Overflow corpus as a source of unlabeled data to train our task-specific word embedding. Words not present in the set of pre-trained words were initialized randomly.

3.2 Convolutional neural networks architecture

Deep learning models have achieved remarkable results in recent years [13]. Convolutional neural networks (CNN) use local connection patterns and impose constrains on the weights to incorporate knowledge [14]. CNN models were proven to be effective for NLP and made impressive results on the practically important task of sentence categorization. Yoon Kim described a series of experiments with CNN built on top of word2vec, and the model performed remarkably well for sentence-level classification tasks [15].

In NLP tasks, a word can be transformed into a vector using word embedding, therefore, a sentence can be represented as a matrix, which is then used as the input of the CNN model. We propose a variant of CNN architecture to capture the semantics of the text for expert recommendation. The model architecture is shown in Figure 1 and the details are as follows.

In the expert recommendation model, the input is the text that corresponds to the user profiles and new questions. Those sentences are first tokenized as a list of words and then converted to a matrix using word embedding to represent terms in the sentences. The rows of the matrix are the word embedding of each token, so a sentence of length n is represented as

$$x_{1:n} = x_1 + x_2 + \cdots + x_n, \tag{1}$$

where $+$ is the concatenation operator.

Convolution filters are used to extract elementary features. In NLP, the width of the convolution filter is usually the same as the dimensionality of the word embedding, we should consider the height of the filters. By applying a filter of height h, a window of h words is used to produce a higher-order feature. For instance, a higher-order feature f_i is generated from the window of words $x_{(i:i+h-1)}$ by

$$f_i = f(w \times x_{(i:i+h-1)} + b). \tag{2}$$

1) https://code.google.com/p/word2vec/.

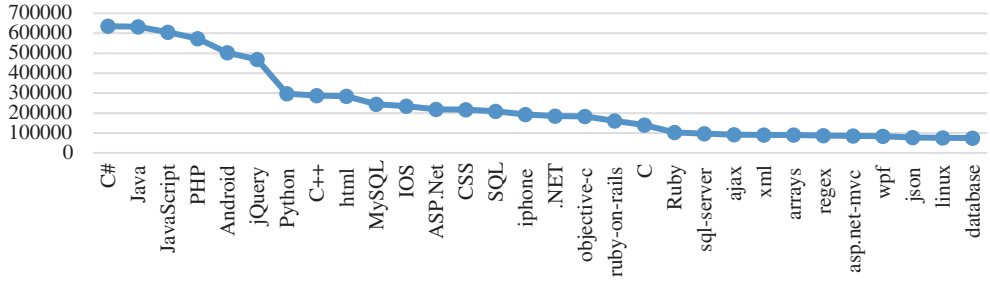

Figure 2 (Color online) Distribution of the most frequent tags in Stack Overflow.

Here $b \in R$ is a bias term and f is a non-linear activation function. In this experiment, we use the ReLU activation function which has proved to be effective. As one feature is extracted from one filter, we use multiple kinds of filters of different sizes to obtain multiple features. Those filters are applied to each possible window of words in the profile to produce a feature map

$$C = [c_1, c_2, \ldots, c_{(n-h+1)}], \quad C \in R^{n-h+1}. \tag{3}$$

The 1-max pooling operation is then applied to the feature map, in other words, the maximum value $c = \max\{C\}$ is taken as the feature. Max pooling is used to extract the highest value which is perceived as the most important feature for each feature map. Together, the outputs generated from each filter map can be concatenated into a fixed-length feature vector, which is then fed to a softmax function of which the output is the probability distribution over labels. At this point, the output labels respectively correspond to each candidate experts.

CNN is a supervised learning model, during training, the input is the candidate experts' profiles and the label of the input is the candidate experts ID. During testing, the input is the new question, the output labels respectively correspond to each candidate experts, and we can get the probability of each candidate expert for giving the best answer. According to the probability, an ordered list of candidate experts is returned.

In the training process, we chose the categorical cross-entropy as loss function, therefore, when the predicted person is not the true expert, the gradient is computed and back-propagated to the previous layers to tune all the parameters including the value of word embedding.

4 Experiments

4.1 Dataset

The experimental dataset was based on a snapshot of Stack Overflow and is available through url[2)], which the website provides for research. Stack Overflow has approximately seven million registered users and over one hundred million questions have been asked by developers. These questions constitute a huge repository and most of them are related to programming, algorithms, and software development tools. All questions are tagged by the askers to specify the categories or their subject areas, and tags make it easy to find interesting questions. To constitute a representative subset that inherits the properties of the original one, we computed the distribution of the most frequent, co-occurring tags. The line charts are shown in Figures 2 and 3. We selected tags that mostly belong to three categories: (i) tags that were very frequent and mostly co-occurred with other tags, (ii) tags that were very frequent and sometimes co-occurred with other tags, and (iii) tags that were very frequent but rarely co-occurred with other tags. Eventually, 18 tags that had different properties were manually selected and those tags are shown in Table 1. In CQA, people are considered experts only in their own specialized areas and tags are good symbols for distinguishing different topics.

2) https://archive.org/details/stackexchange.

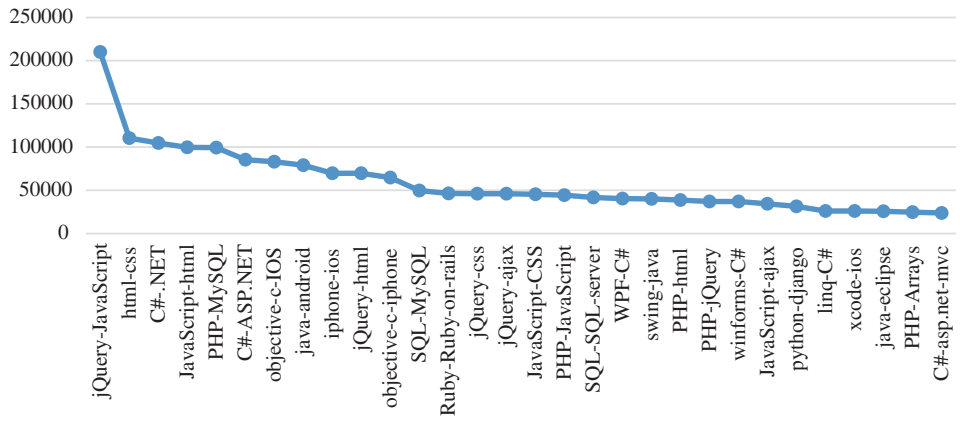

Figure 3 (Color online) Distribution of the most frequent co-occurring tags in Stack Overflow.

Table 1 Tags selected for the training set

Frequently co-occur	Partially co-occur	Rarely co-occur
C#	Python	Django
SQL	SQL-Server	CSS
Linux	Delphi	Ruby
Windows	.NET	Ruby-on-Rails
Java	JavaScript	WPF
C		iPhone
		Android

4.2 Experiment settings

A user may answer many questions related to a new question, but we can not conclude that the user is a qualified respondent considering that the quality of the answer is uncertain. Pal et al. [16] studied the evolution of experts in community question answering, the results show that the likelihood of an expert giving a best answer increased over time. A user with several answers, at least one of which was selected as the best answer, is more likely to have relevant expertise. Unlike in [2], experts are associated with the question they answered. We referred to people as candidate experts according to their past performance, more specifically, people who actually gave the best answer to a question were regarded as experts.

Depending on the truth that in community question answering only a small part of the users are responsible for most of the questions, we derived two training sets that had different requirements about the minimum number of questions for which the user has been selected as having given the best answer. The set D20 included users who had been selected as giving the best answer more than 20 times; the set D40 included those with 40 best answers. The data from 18 tags in 2010 were extracted and the data statistics are shown in Table 2. It was clearly observed that the users who provided more than 20 best answers accounted for less than 10 percent of all those giving best answers, but answered more than half of the questions asked. In the experiment, we applied our method to D20 (with 4390 candidate experts) and to D40 (with 2064 candidate experts). To verify the effect of the model, ten thousand questions asked in 2011, and associated with those tags, were randomly selected as the test set. Those 10000 test questions was associated with 56055 users.

We implemented our model based on Theano, a python library, which supports efficient symbolic differentiation and transparent use of a GPU. To benefit from the efficiency of parallel computation of the tensors, we trained the model on a GPU. The hyper-parameter settings of the convolutional neural networks depend on the dataset being used. In our experiment, hyper-parameters were tuned on dev, through continuous exploration, and we chose one set of hyper-parameters that performed well. The filer region size was set to (3, 4, 5) and the feature maps was set to 100. We trained word embedding using the

Table 2 Data statistics

Data set ID	Questions	Best answerers
All	479531	56055
D20	311857	4390
D40	248300	2064

default parameter in word2vec with the Skip-gram algorithm, and employed dropout on the penultimate layer for regularization [17]. The size of the word embedding was 400 and the dropout rate was 0.5.

We ran the experiment with several variants of CNN:

CNN-rand: In this model, all word vectors were randomly initialized and then modified during training.

CNN-static: In this model, word vectors were pre-trained vectors from word2vec and were kept static during training.

CNN-non-static: In this model word vectors were pre-trained vectors from word2vec, but these pre-trained vectors were fine-tuned during training.

According to our usage scenario, expert recommendation requires a ranked list of experts who would give high quality answers to posted questions, so we had a clear demand for high recall rather than high precision. Thus, we employed the Success-at-N ($S@N$) metric for evaluation. When the best answerer appeared in the top N among the predicted expert list, the $S@N$ value was "1", meaning that the prediction for this question was successful. The $S@N$ value of the whole test set was defined as:

$$S@N = \frac{\sum_{i=1}^{t} \sum_{j=1}^{N} 1(S)}{t}, \tag{4}$$

where t is the number of test questions, $\sum_{j=1}^{N} 1(S)$ equals "1" when the best answerer was among the top N predicted answerers.

4.3 Baseline methods

We evaluated the effectiveness of the proposed model by comparing it with several other expert recommendation methods, including the traditional information retrieval method, classification-based method and topic-based method. The TF-IDF and Language Model and Logistic Regression (LR) were realized by ourselves, while Latent Dirichlet Allocation (LDA) and Segment Topic Model (STM) were reported in [5] and the Sentence Semantic Representation Model (SSRM) was reported in [18].

TF-IDF. It is the abbreviation of term frequency-inverse document frequency, it is intended to compute importance and relevance of a word in a document. For the expert recommendation task, we used the TF-IDF value to compute the cosine distance between each question and each user profile.

Language Model. It is similar to the TF-IDF model, rare words occurs only in a set of documents have great influence. A question is represented as $q = \{w_1, w_2, \ldots, w_n\}$, where w_i is the ith word in the question. Thus, the user expertise in this question can be calculated in view of the words in the question, which are supposed to be generated from the user profiles.

SSRM. Dong et al. [18] extracted topic words based on user tags from the user profiles and added up the vectors of the extracted words as the document vector, then, the document vector was used to compute the similarity between the query and the user profile. Finally, they ranked the users based on similarity. They trained words distributed representation through the skip-gram model.

LR. The 60000 most frequent words from user profiles were selected and the count of each word was used to compute the TF-IDF value. Then the TF-IDF value was used as a feature to train an LR classifier with scikit-learn [19].

LDA. This is a three-level hierarchical Bayesian model that can model topics in text corpora [20]. LDA achieves good performance in the light of capturing the semantics of text.

STM. This is a hierarchical topic model presented by Du et al. [21]. Riahi et al. [5] used STM to conduct experiments and achieved a state of the art result for the content-based method.

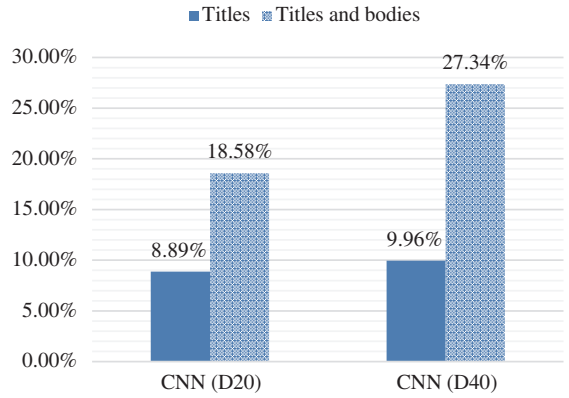

Figure 4 (Color online) $S@1$ for prediction of best answerer using CNN model with different profile configurations.

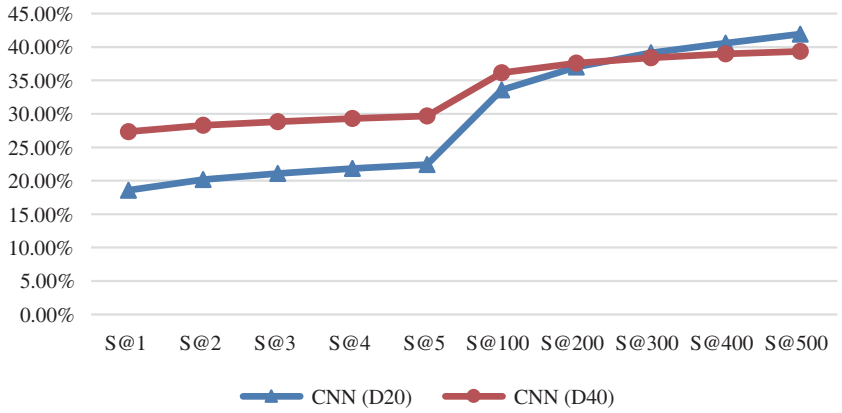

Figure 5 (Color online) $S@N$ for prediction of best answerer using CNN model based on titles and bodies.

4.4 Experiment analysis

Not all answerers are equally important for the purpose of finding experts. Likewise, not all answers are equally important either. A user may answer a great number of questions, therefore, we built representations of each candidate expert by concatenating only questions for which a user's answer was selected as the best answer, and those questions were used to represent the user's expertise. In both sets, we used question titles alone or question titles and bodies together to carry out the experiments.

Two different profile configurations and two different training sets, were used to perform the experiment to see what combination would favor expert recommendation. The CNN model was applied to the experiment and from Figure 4 we can observe that, for both training sets, use titles and Bodies together performed better than only using the titles.

When considering the different training sets, as we can see in Figure 5, we applied our method to D40 (which had 2064 candidate experts) to see if it could get better performance than D20 (which had 4390 candidate experts). The numbers of the candidate experts were the output categories, in the CNN model, fewer categories imply that much fewer parameters need to be trained. This saves considerable training time and provides better generalization. Because expert recommendation is naturally a ranking problem, we selected the candidate experts, and only the candidate experts were considered as output labels of our classification model, which reduced lots of negative samples. The D40 set got better results than did D20, which shows that fewer negative samples can lead to better results, to a certain extent.

The results from the three CNN models are shown in Figure 6. Because different sentences had different lengths, we cut every sentence to the same length to try several CNN models. The results show that to some degree, longer sentence had higher $S@1$ value. However, $S@1$ would not go up past a certain degree. The reason comes down to three points: (i) long sentence contains more useful information, therefore it

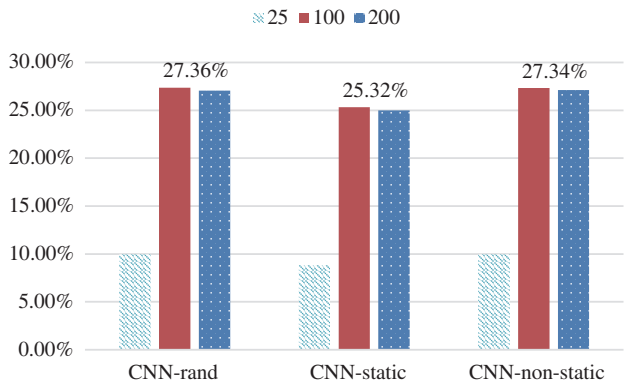

Figure 6 (Color online) Results of the prediction of best answerer based on D40: Y axis shows S@1 values and X axis shows three models of different sentence length.

Table 3 Performance comparison of proposed model and traditional methods based on D40

Method	TF-IDF	Language model	LR	LDA [5]	SSRM [18]	STM [5]	CNN-non-static
S@1	0.0320	0.0310	0.0349	0.0578	0.0578	0.1034	**0.2734**
S@2	0.0442	0.0372	0.0513	0.0765	0.0765	0.1051	**0.2830**
S@3	0.0560	0.0442	0.0625	0.0810	0.0810	0.1192	**0.2884**
S@4	0.0636	0.0478	0.0709	0.0836	0.0836	0.1200	**0.2928**
S@5	0.0714	0.0524	0.0778	0.0856	0.0856	0.1267	**0.2966**

is beneficial to CNN to classify, (ii) most sentence lengths are no more than one hundred words, and (iii) within a certain sentence length, models have learned the useful information.

From Figure 6 we can see that the S@1 values of the CNN-rand model and CNN-non-static model are similar and that both S@1 values are higher than for CNN-static. For the CNN-rand model and CNN-non-static model, the fine-tune operation allowed them to learn more meaningful representations. For example, in the original word2vec pre-trained vectors, "good" is most similar to "bad", because they are syntactically equivalent. However, for vectors in the non-static model that were fine-tuned, "good" is close to "nice" for expressing sentiment.

We compared the performance of the different models for predicting best answerers and the results are presented in Table 3. From the table, we can see that the performance of the traditional information retrieval methods was worst, because they are based on word-matching only and do not capture any textual semantics. The classification-based discriminative method typically made fewer model assumptions but the LR model could not handle problems with high complexity. The LDA-based approaches achieved good performance in terms of capturing the semantics of texts. The SSRM model is based on distributed representations of words to predict the best answerer, and achieved relatively good results. The CNN model exhibit much better performance than any of the other models. This proves that the distributed representations of words can effectively represent the semantic representation of texts, and that, in addition, neural networks can capture more contextual information through the convolutional layer and select more discriminative features through the 1-max pooling layer. This result demonstrates the effectiveness of the proposed method.

5 Conclusion

Experts are the main drivers of answer production, and expert recommendation is an important issue in CQA sites. A method based on the content was adopted to resolve this issue. The user profiles were built based on answering history. Word embedding proved to be effective for capturing the meaningful syntactic and semantic information, thus we utilized word2vec to get word distributed representations and then utilized CNN to predict those who would give the best answers. Experimental results demonstrate

that our model performs significantly better than TF-IDF, Language Model, LR, LDA, SSRM and STM.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61572098, 61632011, 61562080), National Key Research Development Program of China (Grant No. 2016YFB1001103), and Major Projects of Science and Technology Innovation in Liaoning Province (Grant No. 20151060-21).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Balog K, Fang Y, Rijke D M, et al. Expertise retrieval. Found Trends Inf Retr, 2012, 6: 127–256
2 Liu X Y, Croft W B, Koll M. Finding experts in community-based question-answering services. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, 2005. 315–316
3 Li B, King I, Lyu M R. Question routing in community question answering: putting category in its place. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, 2011. 2041–2044
4 Zhou G, Liu K, Zhao J. Joint relevance and answer quality learning for question routing in community QA. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, 2012. 1492–1496
5 Riahi F, Zolaktaf Z, Shafiei M, et al. Finding expert users in community question answering. In: Proceedings of the 21st International Conference on World Wide Web, Lyon, 2012. 791–798
6 Mandal D P, Kundu D, Maiti S. Finding experts in community question answering services: a theme based query likelihood language approach. In: Proceedings of IEEE International Conference on Advances in Computer Engineering and Applications, Ghaziabad, 2015. 423–427
7 Yang L, Qiu M, Gottipati S, et al. Cqarank: jointly model topics and expertise in community question answering. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, San Francisco, 2013. 99–108
8 Yang B, Manandhar S. Tag-based expert recommendation in community question answering. In: Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing, 2014. 960–963
9 Zhao Z, Zhang L J, He X F, et al. Expert finding for question answering via graph regularized matrix completion. IEEE Trans Knowl Data Eng, 2015, 27: 993–1004
10 Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. J Mach Learn Res, 2003, 3: 1137–1155
11 Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch. J Mach Learn Res, 2011, 12: 2493–2537
12 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of International Conference on Neural Information Processing Systems, Lake Tahoe, 2013. 3111–3119
13 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
14 LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proc IEEE, 1998, 86: 2278–2324
15 Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1746–1751
16 Pal A, Chang S, Konstan J A. Evolution of experts in question answering communities. In: Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, Dublin, 2012. 274–281
17 Gao W, Zhou Z H. Dropout Rademacher complexity of deep neural networks. Sci China Inf Sci, 2016, 59: 072104
18 Dong H L, Wang J, Lin H F, et al. Predicting best answerers for new questions: an approach leveraging distributed representations of words in community question answering. In: Proceedings of the 9th International Conference on Frontier of Computer Science and Technology, Dalian, 2015. 13–18
19 Pedregosa F, Michel V G, Gramfort A, et al. Scikit-learn: machine learning in Python. J Mach Learn Res, 2012, 12: 2825–2830
20 Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation. J Mach Learn Res, 2003, 3: 993–1022
21 Du L, Buntine W, Jin H D. A segmented topic model based on the two-parameter Poisson-Dirichlet process. Mach Learn, 2010, 8: 5–19