Expert Finding for Community-Based Question Answering via Ranking Metric Network Learning

Zhou Zhao[1], Qifan Yang[1], Deng Cai[2], Xiaofei He[2] and Yueting Zhuang[1]
[1]College of Computer Science, Zhejiang University
[2]State Key Lab of CAD&CG, Zhejiang University
{zhaozhou,yzhuang}@zju.edu.cn, {yqf.init,dengcai,xiaofeihe}@gmail.com

Abstract

Expert finding for question answering is a challenging problem in Community-based Question Answering (CQA) site, arising in many applications such as question routing and the identification of best answers. In order to provide high-quality experts, many existing approaches learn the user model mainly from their past question-answering activities in CQA sites, which suffer from the sparsity problem of CQA data. In this paper, we consider the problem of expert finding from the viewpoint of learning ranking metric embedding. We propose a novel ranking metric network learning framework for expert finding by exploiting both users' relative quality rank to given questions and their social relations. We then develop a random-walk based learning method with recurrent neural networks for ranking metric network embedding. The extensive experiments on a large-scale dataset from a real world CQA site show that our method achieves better performance than other state-of-the-art solutions to the problem.

1 Introduction

Community-based question answering (CQA) is an Internet-based web service that enables users to post their questions and obtain the answers from other users later. The benefits of CQA have been well-recognized today [Jurczyk and Agichtein, 2007]. We have witnessed the popular CQA sites such as Yahoo ! Answer and Quora. Expert finding is an essential problem in CQA sites [Wang *et al.*, 2013], which arises in many real applications such as question routing [Yang *et al.*, 2013] and the identification of best answers [Bouguessa *et al.*, 2008].

The problem of expert finding is to choose the right experts for answering the questions posted by the users, which has attracted considerable attention recently [Xu *et al.*, 2012; Zhao *et al.*, 2015; Guo *et al.*, 2008; Bouguessa *et al.*, 2008; Zhang *et al.*, 2007; Yang *et al.*, 2013]. Most of the existing works consider the expert finding problem as content-based expert recommendation task, which learns the user model from their past question-answering activities voted by the community through thumb-ups/downs, and then predicts users' performance for answering the questions. Although existing expert finding methods have achieved promising performance, most of them still suffer from the insufficiency of discriminative feature representation for question contents [Qiu and Huang, 2015] and the sparsity of CQA data [Li and King, 2010].

Currently, most of existing expert finding methods [Xu *et al.*, 2012; Zhao *et al.*, 2015; Guo *et al.*, 2008] learn the semantic representation of question contents based on the hand-crafted feature (e.g., bag-of-words). However, the word sequence information is not fully utilized on the feature representation in existing methods, which is critical for question understanding and routing [Shen *et al.*, 2015]. Recently, various embedding methods are proposed for learning the semantics of similar words [Mikolov *et al.*, 2013] and encode the word sequence into low-dimensional continuous embedding space. Thus, the semantic representation of sentences and paragraphs is further improved [Le and Mikolov, 2014; Sutskever *et al.*, 2014]. Since the question contents are always sequential data with variant length, it is natural to employ the deep recurrent neural networks [Hochreiter and Schmidhuber, 1997] to learn the semantic representation of the questions.

On the other hand, the sparsity of CQA data is also a challenging issue for expert finding. In CQA sites, the network between questions and users is constructed from their answer relations. Usually, each question is answered by a few users and thus the CQA network is very sparse [Li and King, 2010]. Fortunately, with the prevalence of online social networks today, it is not difficult to find the relation of CQA users, such as their connections in various online social networks (e.g., Facebook, Twitter, etc.). For example, more than one third of the users in Quora have a twitter account [Zhao *et al.*, 2015]. A social relation between two users provides a strong evidence for them to have common background [Jiang *et al.*, 2015]. Moreover, the performance of users for answering the questions is voted by the community, which provides the relative quality rank of different users to the same question. Thus, leveraging both users' social relations and their relative quality rank is critical for tackling the sparsity problem of CQA data.

In this paper, we consider the problem of expert finding from the viewpoint of ranking metric network embedding. Specifically, we integrate both semantic representation of the questions and heterogeneous CQA network structure learning into a unified Ranking Metric Network Learning framework, named as RMNL. We then develop a random-walk based learning method with deep recurrent neural networks to learn ranking metric embedding for questions and users in the proposed heterogeneous CQA network. When a certain question is queried, RMNL can rank the users for it based on the trained ranking metric embedding. The main contributions of this paper are as follows:

- Unlike the previous studies, we present the problem of expert finding from the viewpoint of ranking metric network embedding. We learn the semantic representation of questions and users in CQA networks simultaneously with deep recurrent neural networks.
- We adopt a random-walk based learning method with recurrent neural networks to learn ranking metric embedding from the proposed heterogeneous CQA network. Our learning method is scalable for large-scale CQA networks which is easily parallelized.
- We evaluate the performance of our method on the well-known question answering site Quora and the popular social network Twitter. The extensive experiments show that our method can outperform several state-of-the-art solutions to the problem.

The rest of this paper is organized as follows. In Section 2, we introduce the problem of expert finding from the viewpoint of ranking metric learning and propose the ranking metric network learning method. A variety of experimental results are presented in Section 3. We provide a brief review of the related work about expert finding and network embedding in Section 4. Finally, we provide some concluding remarks in Section 5.

2 Expert Finding via Ranking Metric Network Embedding

In this section, we first present the problem of expert finding from the viewpoint of ranking metric embedding, and then introduce the ranking metric network learning framework. Finally, we propose a random-walk based learning method with deep recurrent neural networks for the problem.

2.1 The Problem

Before presenting the problem, we first introduce some basic notions and terminologies. Since the questions in CQA sites are always the sequential data with variant length, we then encode the question contents into fixed length feature vectors for semantic representation using recurrent neural networks, which have shown beneficial to word sequence learning with variant length in [Le and Mikolov, 2014]. Given a set of input questions $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, we take the last hidden layer of neural networks as the semantic embedding of the questions by $Q = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$. We then employ the ranking metric function $f_{\mathbf{u}_i}(\mathbf{q}_j) = \mathbf{q}_j^T \mathbf{u}_i$ that quantifies the quality of the i-th user for answering the j-th question. We denote the set of user embeddings by $U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$ where $\mathbf{u}_i$ is the embedding vector for the latent expertise of the i-th user. The quality of users for answering the questions is voted through thumb-ups/downs, which indicates the community's long term review for users' performance.

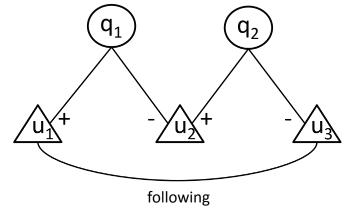


Figure 1: Ranking Metric Heterogeneous CQA Network

Currently, most of the existing methods [Yang *et al.*, 2013; Zhao *et al.*, 2015] learn the user model from the received absolute votes of their past question-answering activities. However, the absolute votes are the biased estimator in CQA learning. This is because the community votes for users are also dependent on the question popularity in CQA sites. Thus, we introduce the relative quality rank to model the performance of users for answering the questions, which is in the form of triplet constraints. We denote a triplet constraint by the ordered tuple (j, i, k), meaning that "the i-th user obtains more votes than the k-th user for answering the j-th question". Let $T = \{(j, i, k)\}$ denote the set of triplet constraints obtained from the community votes for a set of M users answering N different questions. More formally, we aim to learn the ranking metric function that for any $(j, i, k) \in T$, the inequality holds:

$$f_{\mathbf{u}_j}(\mathbf{q}_j) > f_{\mathbf{u}_k}(\mathbf{q}_j) \Longleftrightarrow \mathbf{q}_i^T \mathbf{u}_j > \mathbf{q}_i^T \mathbf{u}_k.$$

We observe that many of the CQA users also have connections on some online social networks. Thus, we denote the relation between users by $\mathbf{S} \in R^{m \times m}$. The entry $s_{ij} = 1$ if the i-th user and the j-th user are friends, otherwise, $s_{ij} = 0$. We then utilize the users' social relations to tackle the sparsity of CQA data. We propose the heterogeneous CQA network by integrating both users' social relations and their relative quality rank to for answering the questions. We denote the proposed heterogeneous CQA network by $G = (V, E)$ where the set of nodes V is composed of question contents X and users U, and the set of edges consists of relative quality rank T and social relations S. We illustrate a simple example of ranking metric heterogeneous CQA network modeling in Figure 1. We show the relative quality rank of users for answering the given questions as follows. The question $\mathbf{q}_1$ is answered by the high-quality user $\mathbf{u}_1$ (i.e., marked with $+$ on the answering relation) and low-quality user $\mathbf{u}_2$ (i.e., marked with $-$ on the answering relation). We also illustrate the following relation between users $\mathbf{u}_1$ and $\mathbf{u}_3$ (i.e., $s_{13} = 1$) in Figure 1.

Using the notations above, we define the problem of expert finding from the viewpoint of ranking metric learning as follows. Given the input questions X, the triplet constraints T derived from community vote, and the heterogeneous CQA network G, our goal is to learn the ranking metric function $f_{\mathbf{u}}(\cdot)$ for all embedding users $\mathbf{u}$ and then rank the users for answering the questions. The best users qualifying for answering the question $\mathbf{q}$ are then selected according to $f_{\mathbf{u}}(\mathbf{q})$.

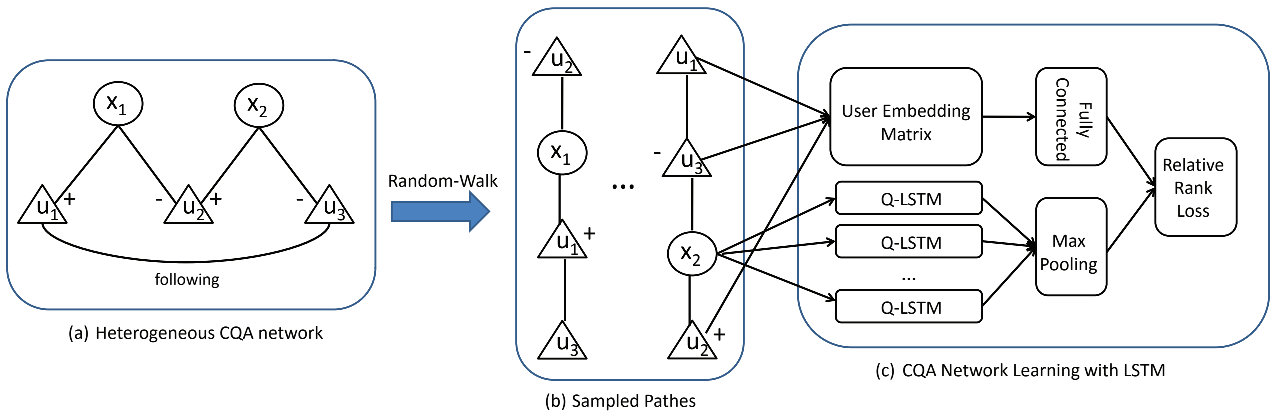


Figure 2: The Overview of Ranking Metric Heterogeneous Network Learning. (a) The heterogeneous CQA network is constructed by integrating both relative quality rank of users in CQA sites and their social network relations. b) A random walker is walking on the heterogeneous CQA networks to sample the data paths. (c) The questions and users in CQA networks are encoded into fixed feature vectors based on relative rank loss model.

2.2 Ranking Metric Network Embedding with Recurrent Neural Networks

In this section, we propose the ranking metric network embedding for expert finding in CQA network and present the learning process in Figures 2(a), 2(b) and 2(c).

Inspired by DeepWalk [Perozzi *et al.*, 2014], we first sample the paths from the heterogeneous CQA network by random-walk method, illustrated in Figure 2(b). We considered the sampled paths as the context windows for the vertex embedding in networks. For example, the context of vertex v_i with the c-th sampled window is $W_c = \{v_{i-\frac{w}{2}}, \dots, v_{i-1}, v_{i+1}, v_{i+\frac{w}{2}}\}$ where the length of sampled path is w. Following the neural language models [Mikolov *et al.*, 2013], the vertex embedding can be formulated as the optimization problem by

$$\min_{\Phi} -\log p(\{v_{i-\frac{w}{2}}, \dots, v_{i-1}, v_{i+1}, v_{i+\frac{w}{2}}\} | \Phi(v_i)),$$

where Φ is the embedding function for all the vertices V and $\Phi(v_i)$ is the embedding of v_i.

However, the goal of DeepWalk is to learn network embedding only from the network structure, which is an unsupervised learning method. We notice that the side information in our proposed CQA network are question contents and relative quality rank, which are critical for the problem of expert finding. Thus, the DeepWalk method cannot be directly applied for the CQA network embedding. To leverage the beneficial supervised information, we integrate the random-walk method in DeepWalk with recurrent neural networks based learning into a unified CQA network learning framework. We then introduce our approaches for question representation and relative quality rank learning in CQA networks below, respectively.

We first choose the proper embedding method for question representation in CQA networks. Given a sequence of words for question $\mathbf{x}_i$, we represent the t-th word by pre-training word embedding [Mikolov *et al.*, 2013] as x_{it} and then use the sequence $(x_{i1}, x_{i2}, \dots, x_{ik})$ as the input of the corresponding recurrent neural network. However, simple recurrent neural network is difficult to train because of the resulting long term dependencies [Sutskever *et al.*, 2014]. Therefore, we choose the variant recurrent neural networks called long-short term memory (LSTM) [Hochreiter and Schmidhuber, 1997] to learn the question embeddings by:

$$\begin{aligned}
\mathbf{i}_t &= \delta(\mathbf{W}_i\mathbf{x}_t + \mathbf{G}_i\mathbf{h}_{t-1} + \mathbf{b}_i), \\
\hat{\mathbf{C}}_t &= tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{G}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{f}_t &= \delta(\mathbf{W}_f\mathbf{x}_t + \mathbf{G}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{C}_t &= \mathbf{i}_t \cdot \hat{\mathbf{C}}_t + \mathbf{f}_t \cdot \mathbf{C}_t, \\
\mathbf{o}_t &= \delta(\mathbf{W}_o\mathbf{x}_t + \mathbf{G}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{C}_t + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \cdot tanh(\mathbf{C}_t).
\end{aligned} \tag{1}$$

where δ represents the sigmoid activation function; $\mathbf{W}$s, $\mathbf{G}$s and $\mathbf{V}_o$ are weight matrices, and $\mathbf{b}$s are bias vectors. The gates in LSTM cell can modulate the interactions between the memory cell itself and its environment. The input gate can allow incoming signal to alter the state of the memory cell to have an effect on other neurons or prevent it. Moreover, the forget gate can allow the cell to remember or forget its previous state. The architecture structure of LSTM can be found in [Hochreiter and Schmidhuber, 1997].

We train the LSTM networks for question embedding named Q-LSTM and then take the output of the last LSTM cell, $\mathbf{h}_k$, as the semantic embedding of question $\mathbf{q}$. Considering the fact that the questions may be in the paragraph with several sentences in real CQA sites, we split them into sentences for learning the semantic embedding by Q-LSTM and then merge the embedding by an additional max-pooling layer, illustrated in Figure 2(c).

As illustrated in Figures 2(b) and 2(c), the context windows W for network vertices are sampled by random-walk method and the question representation corresponds to LSTM, respectively. We now introduce our method for learning CQA network embedding with question representation and relative quality rank. For each vertex v_i in the context window W, we now design its loss function as follows:

Algorithm 1 Ranking Metric Network Learning

Input: Heterogeneous CQA network $G = (V, E)$, window size w, embedding size d, walks per vertex γ, number of iterations T

Output: Matrix of question representation $\mathbf{Q} \in R^{d\times n}$ and user embedding $\mathbf{U} \in R^{d\times m}$

```
for t = 1 → T do
    for j = 1 → γ do
        O = Shuffle(V)
        for each v_i ∈ O do
            W_{v_i} ← Random-walk(G, v_i, w).
            Accumulate the loss in W_{v_i} by Equation (2).
            Update the embedding by SGD.
```

$$l(v_i) = \begin{cases} \sum_{u^+,u^- \in W} \max(0, m + f_{\mathbf{u}^-}(\mathbf{v}_i) - f_{\mathbf{u}^+}(\mathbf{v}_i)), & v_i \in Q \\ \sum_{u \in W} \|u - v_i\|^2, & v_i \in U, \end{cases} \tag{2}$$

where the superscript u^+ denotes the high-quality expert (with higher votes) and u^- denotes the low-quality expert (with lower votes) for answering question v_i. We denote the hyper-parameter m $(0 < m < 1)$ controls the margin in the loss function and Q, U are the sets of questions and users, respectively.

Using the proposed approaches above, we can integrate both the question contents and users' relative quality rank into the unified the CQA network embedding framework for expert finding. The trained model can rank the users for given question directly and return the embedding for other CQA tasks such as question answering or question retrieval.

2.3 Ranking Metric Network Learning

In this section, we present the details of our ranking metric network learning method and summarize the main training process in Algorithm 1.

We first start a walker to sample the paths (i.e., context window) from the heterogenous CQA network, and then accumulate the training loss by Equation (2). We notice that the following relationship in microblogs may be directed, our walker is only allowed to walk from the followers to their following users. The reason is that the followers should have similar background with their followed users but not vice versa. We denote all the model coefficients including neural network parameters and the result embeddings by Θ. Therefore, the objective function in our learning process is given by:

$$\min_{\Theta} L(\Theta) = \sum_{W} \sum_{v_i} L(v_i) + \lambda \|\Theta\|^2 \tag{3}$$

where λ is the trade-off parameter between the training loss and regularization.

To optimization the objective, we employ the stochastic gradient descent (SGD) with the diagonal variant of AdaGrad in [Qiu and Huang, 2015]. At the t-th step, the parameters Θ is updated by:

$$\Theta_t \leftarrow \Theta_{t-1} - \frac{\rho}{\sqrt{\sum_{i=1}^{t} g_i^2}} g_t \tag{4}$$

where ρ is the initial learning rate and g_t is the subgradient at time t.

3 Experiments

In this section, we conduct several experiments on the question-answering site Quora, and the social network, Twitter to show the effectiveness of our approach RMNL for the problem of expert finding in CQA sites.

3.1 Data Preparation

We evaluate the performance of our method using the Quora dataset in [Zhao *et al.*, 2015], which is obtained from a popular question answering site, Quora. The dataset contains 444,138 questions, 95,915 users and 887,771 answers from Quora, and users' following relationship in Twitter's social network. The quality of users on answering the question is indicated through thumbs-up/down voted by the community. Following the experimental setting in [Yang *et al.*, 2013; Zhao *et al.*, 2015], we sort the questions based on their posted timestamp. We use the first 60%, 70% and 80% posted questions as training set and the remaining ones for testing. So the training and testing data do not have overlap.

3.2 Evaluation Criteria

We evaluate the performance of our proposed RMNL method based on three widely-used ranking evaluation criteria for the problem of expert finding in CQA site, i.e., normalized Discounted Cumulative Gain (nDCG) [Yang *et al.*, 2013], Precision@1 [Zhu *et al.*, 2014] and Accuracy [Guo *et al.*, 2008].

For ground truth, we consider all the corresponding answerers as the candidate user set and their received thumbs-up/down as the ground truth ranking scores. The experts for the questions tend to receive more scores. Note that our task is to predict the relative quality rank of users instead of the exact thumbs-up/down values. Given the testing question set $\mathcal{Q}$, we denote the predicted ranking of all the users for question q by R^q and the ranked user on i-th position by r_i. We now introduce the evaluation criteria below.

- **nDCG.** The nDCG for ranked users for answering question q is given by

$$nDCG = \frac{DCG}{IDCG}, DCG = rel_1 + \sum_{i=2}^{|R^q|} \frac{rel_i}{\log_2 i},$$

  where IDCG is the DCG of ideal ordering, $|R^q|$ is the number of ranked users for question q and rel_i is the relevance score between question q and the i-th in the ranking list, which is indicated by thumbs-up/down.

- **Precision@1.** The Precision@1 is used to measure the raking quality of the best answerer, given by

$$Precision@1 = \frac{|\{q \in Q | r^q_{best} = 1\}|}{|Q|}.$$

  In other words, Precision@1 computes the average number of times that the best answerer is ranked on top by a certain algorithm.

- **Accuracy.** The Accuracy is the normalized criteria of accessing the ranking quality of the best answerer, given by

$$Accuracy = \frac{1}{|Q|} \sum_{q \in Q} \frac{|R^q| - r^q_{best}}{|R^q| - 1},$$

Table 1: Experimental results on nDCG with different proportions of data for training.

Methods	nDCG		
	60%	70%	80%
AuthorityRank	0.6723	0.7014	0.7108
TSPM	0.6316	0.645	0.6627
DRM	0.6585	0.6646	0.6707
GRMC-AGM	0.7152	0.7292	0.7346
DeepWalk	0.6238	0.6296	0.6381
RMNL	**0.7191**	**0.7364**	**0.741**

Table 2: Experimental results on Precision@1 with different proportions of data for training.

Methods	Precision@1		
	60%	70%	80%
AuthorityRank	0.4441	0.4785	0.4904
TSPM	0.3781	0.3944	0.4177
DRM	0.4183	0.4242	0.4311
GRMC-AGM	0.4829	0.4971	0.5125
DeepWalk	0.3659	0.3717	0.3815
RMNL	**0.5053**	**0.5204**	**0.5405**

Table 3: Experimental results on Accuracy with different proportions of data for training.

Methods	Accuracy		
	60%	70%	80%
AuthorityRank	0.5755	0.6133	0.6152
TSPM	0.516	0.5229	0.5384
DRM	0.5579	0.5534	0.5501
GRMC-AGM	0.5455	0.5342	0.5522
DeepWalk	0.4988	0.492	0.4918
RMNL	**0.6120**	**0.6248**	**0.6411**

where $Accuracy = 1$ (best) means that the best answerer returned by a certain algorithm always ranks on top while $Accuracy = 0$ means the opposite.

In summary, nDCG is the measure for the ranking quality of all answerer by a certain method while Precision@1 and Accuracy are the measures for the ranking quality of the best answerer.

3.3 Performance Comparisons

We compare our proposed method with other five state-of-the-art methods for the problem of expert finding in CQA site as follows:

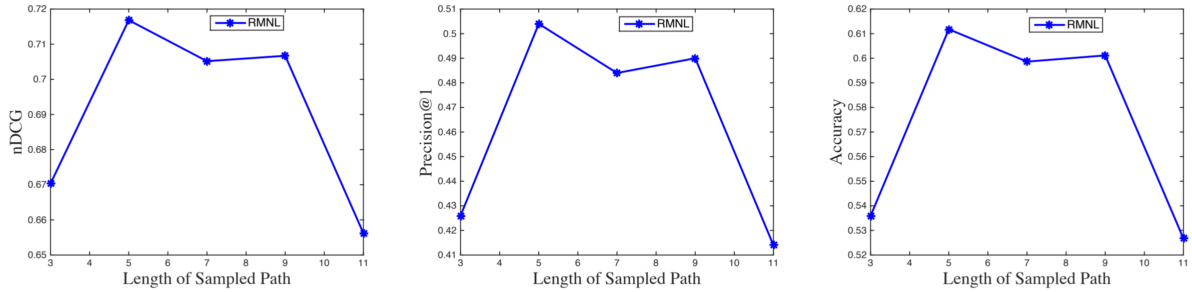
- **AuthorityRank** method [Bouguessa *et al.*, 2008] computes user authority based on the number of best answers provided, which is an in-degree method.
- **TSPM** method [Guo *et al.*, 2008] is a topic-sensitive probabilistic model for expert finding in CQA site, which learns question representation via LDA-based model.
- **DRM** method [Xu *et al.*, 2012] is also a topic-sensitive probabilistic model, which learns question representation via PLSA-based model.
- **GRMC-AGM** method [Zhao *et al.*, 2015] is a graph regularized matrix completion model, which learns user model from the viewpoint of missing value estimation.
- **DeepWalk** method [Perozzi *et al.*, 2014] learns the embedding of both questions and users based on the network structure.

Among them, the AuthorityRank and DeepWalk methods learn the user model for expert finding only based on network structure while the TSPM, DRM and GRMC-AGM methods learn the user model based on both the question contents and users' thumbs-up/downs in CQA sites. Unlike the previous studies, our method RMNL learns the user model from the proposed CQA network. The input words of our methods are initialized by pre-trained word embeddings [Mikolov *et al.*, 2013] and the weights of LSTMs are randomly chosen by a Gaussian distribution with zero mean. We then employ the random-walk based learning with LSTM networks for training our proposed RMNL model.

Tables 1, 2 and 3 show the evaluation results on nDCG, Precision@1 and Accuracy, respectively. The evaluation were conducted with different ratio of the data from 60%, 70% to 80%. The hyperparameters and parameters which achieve the best performance on the validation set are chosen to conduct the testing evaluation. We report the average value of all the methods on the three evaluation criteria. There experiments reveal a number of interesting points:

- The supervised methods, AuthorityRank, TSPM, DRM and GRMC-AGM outperform the unsupervised DeepWalk method, which suggests that the supervised information such as users' relative quality rank and question contents are critical for the problem.
- The GRMC-AGM achieves better performance than other baselines. This suggests that users' social relations can also improve the performance of expert finding.
- In all the cases, our RMNL method achieves the best performance. This fact shows that the ranking metric network learning framework that exploits both deep representation of question contents and users' relative quality rank can further improve the performance of expert finding.

In our approach, there are three essential parameters, which are the length of sampled paths (size of context window), the size of network embeddings and the number of walks. We vary the length of sampled paths from 3 to 7, the size of embedding from 100 to 1000, and the number of walks from 5 to 20. The performance trend of our method becomes stable after the size of embedding larger than 200 and the number of walks larger than 10, which is similar in [Perozzi *et al.*, 2014]. We now mainly investigate effect of the length of sampled paths on our method since the varying of the length of sampled paths is related to our proposed CQA networks. We use 60% of the data for training and then illustrate the performance of our method by varying the length of sampled paths on nDCG, Precision@1 and Accuracy in Figures 3(a), 3(b) and 3(c), respectively. The performance trend is consistent when we use 70% and 80% of data for training. We observe that our method achieve the best performance when the length



(a) nDCG (b) Precision@1 (c) Accuracy

Figure 3: Effect of Sampled Path Length on nDCG, Precision@1 and Accuracy using 60% of data for training.

of sampled paths is set to 5. This suggests that by leveraging both users' relative quality rank to the given questions and their social relations for network embedding, we can further improve the quality of expert finding.

4 Related Work

In this section, we briefly review some related work on expert finding, deep content-based recommendation and network learning.

The existing work for the problem of expert finding can be mainly categorized as authority-based approaches [Bouguessa *et al.*, 2008; Zhu *et al.*, 2014] and topic-based approaches [Xu *et al.*, 2012; Guo *et al.*, 2008; Zhang *et al.*, 2007; Yang *et al.*, 2013]. Bouguessa et al. [Bouguessa *et al.*, 2008] choose the experts to answer the questions based on the number of best answers provided by users, which is an In-degree method. Zhu et al. [Zhu *et al.*, 2014] measure the category relevance of questions and rank user authority in extended category graph. Xu et al. [Xu *et al.*, 2012] propose a probabilistic dual role model for expert finding. Yang et al. [Yang *et al.*, 2013] devise the CQArank model that learns the latent topic of questions and the user model. Guo et al. [Guo *et al.*, 2008] develop the topic sensitive probabilistic model to learn the user model. Zhang et al. [Zhang *et al.*, 2007] learn the expertise of users from their networks. Zhao et al. [Zhao *et al.*, 2015] tackle the expert finding problem via graph regularized matrix completion. Unlike previous studies, formulate the problem from the viewpoint of ranking metric network embedding, which can be solved via random-walk based learning with recurrent neural networks.

Recently, deep learning models show great potential for learning effective representation and deliver state-of-the-art performance in natural language processing [Shen *et al.*, 2015; Dong *et al.*, 2015] and data mining applications [Perozzi *et al.*, 2014; Tang *et al.*, 2015a]. We now review the two very related techniques for our problem: (1) deep content-based recommendation, (2) network learning. Deep recommendation methods employ the deep learning model to learn the content representation for various recommendation tasks. Van et al. [Van den Oord *et al.*, 2013] employ the traditional convolutional neural networks for content representation in music recommendation. Wang et al. [Wang *et al.*, 2014] propose the robust representation model based on auto-encoder for collaborative filtering. Elkahky et al. [Elkahky *et al.*, 2015] develop the deep cross-domain model for recommendation. On the other hand, the network learning methods exploit the network structure for embedding. Perozzi et al. [Perozzi *et al.*, 2014; Luo *et al.*, ; Tang *et al.*, 2015b] propose the network structure embedding method. Chang et al. [Chang *et al.*, 2015] propose the embedding method for heterogeneous networks. Yang et al. [Yang *et al.*,] develop the learning method for attributed networks. Wu et al. [Wu *et al.*, 2016] devise the embedding algorithm for multi-modal networks. Tang et al. [Tang *et al.*, 2015a] propose the predictive embedding method for heterogeneous networks. However, the objective of ranking metric network learning in our problem is different from these deep learning methods. Thus, these methods may not be suitable for our problem.

5 Conclusion

In this paper, we formulated the problem of expert finding from the viewpoint of learning ranking metric embedding. We propose the heterogenous CQA network that exploits both users' relative quality rank to the given questions and their social relations for expert finding. We then develop a novel ranking metric network learning framework that tackles both the insufficiency of question representation and the sparsity of CQA data issues. We develop a random-walk based learning method with recurrent neural networks for ranking metric embedding in heterogenous CQA networks. We evaluate the performance of our method using the dataset from the well-known question answering site Quora and the popular social network Twitter. The extensive experiments demonstrate that our method can achieve better performance than several state-of-the-art solutions to the problem.

Acknowledgements. This work was supported by National Basic Research Program of China (973 Program) under Grant 2012CB316400, Grant 2013CB336500 and Grant 2015CB352300, National Program for Special Support of Top-Notch Young Professionals, and National Natural Science Foundation of China under Grant 61233011 and Grant 61125203, and the Fundamental Research Funds for the Central Universities.

References

[Bouguessa *et al.*, 2008] Mohamed Bouguessa, Benoît Dumoulin, and Shengrui Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *SIGKDD*, pages 866–874. ACM, 2008.

[Chang *et al.*, 2015] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. Heterogeneous network embedding via deep architectures. In *SIGKDD*, pages 119–128. ACM, 2015.

[Dong *et al.*, 2015] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over freebase with multi-column convolutional neural networks. In *ACL*, volume 1, pages 260–269, 2015.

[Elkahky *et al.*, 2015] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. WWW, 2015.

[Guo *et al.*, 2008] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. Tapping on the potential of q&a community by recommending answer providers. In *CIKM*, pages 921–930. ACM, 2008.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Jiang *et al.*, 2015] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social recommendation with cross-domain transferable knowledge. 2015.

[Jurczyk and Agichtein, 2007] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *CIKM*, pages 919–922. ACM, 2007.

[Le and Mikolov, 2014] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[Li and King, 2010] Baichuan Li and Irwin King. Routing questions to appropriate answerers in community question answering services. In *CIKM*, pages 1585–1588. ACM, 2010.

[Luo *et al.*,] Yuanfei Luo, Quan Wang, Bin Wang, and Li Guo. Context-dependent knowledge graph embedding.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, pages 701–710. ACM, 2014.

[Qiu and Huang, 2015] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, pages 1305–1311, 2015.

[Shen *et al.*, 2015] Yikang Shen, Wenge Rong, Zhiwei Sun, Yuanxin Ouyang, and Zhang Xiong. Question/answer matching for cqa system via combining lexical and sequential information. In *AAAI*, 2015.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[Tang *et al.*, 2015a] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *SIGKDD*, pages 1165–1174. ACM, 2015.

[Tang *et al.*, 2015b] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. WWW, 2015.

[Van den Oord *et al.*, 2013] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, pages 2643–2651, 2013.

[Wang *et al.*, 2013] Gang Wang, Konark Gill, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. Wisdom in the social crowd: an analysis of quora. In *WWW*, pages 1341–1352. International World Wide Web Conferences Steering Committee, 2013.

[Wang *et al.*, 2014] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. *arXiv preprint arXiv:1409.2944*, 2014.

[Wu *et al.*, 2016] Fei Wu, Xinyan Lu, Jun Song, Shuicheng Yan, Zhongfei Zhang, Yong Rui, and Yueting Zhuang. Learning of multimodal representations with random walks on the click graph. 2016.

[Xu *et al.*, 2012] Fei Xu, Zongcheng Ji, and Bin Wang. Dual role model for question recommendation in community question answering. In *SIGIR*, pages 771–780. ACM, 2012.

[Yang *et al.*,] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information.

[Yang *et al.*, 2013] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. Cqarank: jointly model topics and expertise in community question answering. In *CIKM*, pages 99–108. ACM, 2013.

[Zhang *et al.*, 2007] Jun Zhang, Mark S Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *WWW*, pages 221–230. ACM, 2007.

[Zhao *et al.*, 2015] Zhou Zhao, Lijun Zhang, Xiaofei He, and Wilfred Ng. Expert finding for question answering via graph regularized matrix completion. *TKDE*, pages 993–1004, 2015.

[Zhu *et al.*, 2014] Hengshu Zhu, Enhong Chen, Hui Xiong, Huanhuan Cao, and Jilei Tian. Ranking user authority with relevant knowledge categories for expert finding. *World Wide Web*, 17(5):1081–1107, 2014.