Community-Based Question Answering via Heterogeneous Social Network Learning

Hanyin Fang, Fei Wu, Zhou Zhao,*[*] Xinyu Duan, and **Yueting Zhuang**
College of Computer Science, Zhejiang University
and the Key Lab of Big Data Intelligent Computing of Zhejiang Province, China
{fhy881229, wufei, zhaozhou, duanxinyu, yzhuang}@zju.edu.cn

Martin Ester
School of Computing Science, Simon Fraser University, Canada
ester@cs.sfu.ca

Abstract

Community-based question answering (cQA) sites have accumulated vast amount of questions and corresponding crowdsourced answers over time. How to efficiently share the underlying information and knowledge from reliable (usually highly-reputable) answerers has become an increasingly popular research topic. A major challenge in cQA tasks is the accurate matching of high-quality answers w.r.t given questions. Many of traditional approaches likely recommend corresponding answers merely depending on the content similarity between questions and answers, therefore suffer from the sparsity bottleneck of cQA data. In this paper, we propose a novel framework which encodes not only the contents of question-answer(Q-A) but also the social interaction cues in the community to boost the cQA tasks. More specifically, our framework collaboratively utilizes the rich interaction among questions, answers and answerers to learn the relative quality rank of different answers w.r.t a same question. Moreover, the information in heterogeneous social networks is comprehensively employed to enhance the quality of question-answering (QA) matching by our deep random walk learning framework. Extensive experiments on a large-scale dataset from a real world cQA site show that leveraging the heterogeneous social information indeed achieves better performance than other state-of-the-art cQA methods.

Introduction

Community-based question answering (cQA) is an Internet-based crowdsourcing service which enables users to post their questions on a cQA website and be answered by other users later. Some cQA sites are becoming more and more popular in the real world such as Yahoo! Answers and Quora. The answers in these cQA sites are highly specific for personal questions and facilitate different users to directly quest answers from complex and heterogeneous information. The benefits of cQA have been proven in (Jurczyk and Agichtein 2007; Li, Lyu, and King 2012) and cQA has attracted lots of attention in the fields of information retrieval and natural language processing research (Bilotti et al. 2010).

Although the cQA service has shown its promising applicability and the rapid increasing of crowdsourced cQA data provides an opportunity to comprehend the implicit knowledge of crowdsourced questions and answers better, a major challenge for cQA tasks comes from the sparsity of QA data (Li and King 2010). As shown in the upper part of Figure 1, the links between questions and answers are sparse (all the four questions are isolated and only answer 2 and answer 3 are connected by question 2). Due to this sparsity of cQA data, existing methods which only utilize the contents of Q-A can merely learn the global semantics from different questions. Since the questions and answers are all posted by users and a reliable user may be followed by another users, it is attractive to introduce *user context* (e.g., who posts an answer and one followed by others) into cQA in order to tackle the data sparsity. Moreover, the rating scores voted by community users provide the relative quality rank of different answers w.r.t the same question. Thus, how to leverage these valuable social information is very significant for cQA tasks. Figure 1 shows our intuition of the used heterogeneous social networks (the links between questions and askers are not presented because of the unavailability of the corresponding data in this paper).

Besides the valuable social information, the textual contents of questions and answers are also very important for cQA tasks. Most of the existing works treat the QA problem as a short text matching task, such as recommending high quality answers to questions or the retrieval of similar questions (Shen et al. 2015; Zhou et al. 2013). An important issue in short text matching task is the insufficiency of discriminative features for the textual contents. Though the traditional hand-crafted feature (e.g., bag-of-words) has been proven successful in a range of text modeling tasks (Fang et al. 2015; Qu et al. 2009; Gabrilovich and Markovitch 2007), it is unable to capture the word sequence information which is critical for short text matching task, as mentioned in (Qiu and Huang 2015). In recent years, various methods are proposed to learn the semantics of similar words (Mikolov et al. 2013b) and encode the order information of word sequence to enhance the semantic representation of sentences and paragraphs (Socher et al. 2013; Sutskever, Vinyals, and Le 2014; Hu et al. 2014) in a low-

[*]Corresponding author

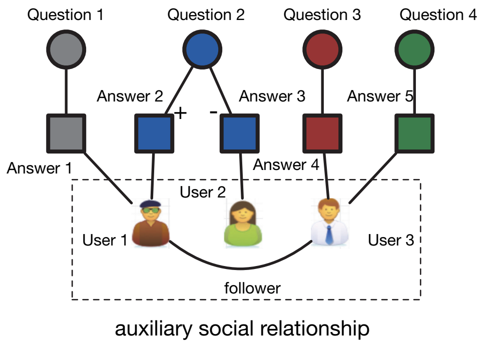

Figure 1: The intuitive illustration of the used heterogeneous social networks. There are in total three kinds of nodes(e.g., question, answer and answerer) which represent a particular theme of relationship in the heterogeneous social networks. Our approach effectively exploits the rich interactions between heterogeneous social networks, for example, the quality difference between a high-quality answer (e.g., answer 2 marked with +) delivered by a reliable answerer (e.g., User1) and a low-quality answer (e.g., answer 3 marked with -) posted by another answerer (e.g., User2), and the follower relationship (e.g., between User1 and User3). These rich interactions are arguably beneficial to boost the cQA task.

dimensional continuous embedding space.

In this paper, we adopt a random walk method to exploit the plentiful social information from heterogeneous social networks to solve the sparsity problem in cQA tasks and combine it with a deep recurrent neural network which promisingly models the textual contents of questions and answers. Our proposed framwork is named as **HSNL** (cQA via **H**eterogeneous **S**ocial **N**etwork **L**earning). In our proposed HSNL, the questions, answers and users are specifically modeled to simultaneously utilize the textual contents and the social relationships. When a certain question is queried, HSNL can retrieve the best answer for it or recommend suitable answers from the similar questions. Moreover, the expertise of users learned by our HSNL can be beneficial for other cQA tasks, such as expert finding.

It is worthwhile to highlight several contributions of our work here:

- Different from the traditional content-based methods, a novel framework HSNL is proposed in this paper to leverage the plentiful social information from heterogeneous social networks, which not only greatly mitigates the sparsity problem in cQA tasks but also utilizes the relative quality rank of different answers.
- We adopt a random walk method to exploit the relationship information from heterogeneous social networks. Our proposed approach is scalable for large-scale social networks and easily parallelized because only a little fraction of the network data is loaded at the same time in training process.
- A deep learning model is introduced in our proposed HSNL to directly calculate the matching score between questions and answers. Meanwhile, the features of questions, answers and users can be learned simultaneously for many cQA tasks, such as question retrieval or expert finding.

Related Work

Recently, some works are proposed on applications of deep neural networks to cQA tasks. In (Shen et al. 2015), the authors calculate a similarity matrix for each pair of question and answer 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 covolutional neural network used in (Shen et al. 2015), (Qiu and Huang 2015) introduce a dynamic convolutional neural network (Blunsom et al. 2014) to encode the variable-length sentences of questions and answers in semantic space and models their interactions with a tensorial top layer. Besides the CNNs, another kind of neural networks has been successfully applied in textual content analysis. In (Le and Mikolov 2014), 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 (Sutskever, Vinyals, and Le 2014), a multilayered RNN is used to map the input sentence to a fixed-dimensional vector.

Similar to the spirit of our work, some methods introduce side information to improve the quality of QA tasks. In (Iyyer et al. 2014), a recursive neural network with sentence dependency-tree is used for factoid question answering task and in (Zhou et al. 2013), the authors build a concept thesaurus based on the world knowledge of Wikipeida and then leverage these semantic relations to enhance the question retrieval task. However, the setting of factoid QA is quite different from the cQA scenario and the method in (Zhou et al. 2013) only models the semantic relation in word level by hand-crafted features.

There are a few proposed works on exploiting the social information for QA tasks. In (Zhao et al. 2015), 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 (Jiang et al. 2015). 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. Although these methods can exploit social information contained in the social link structures, they treat the items (e.g., questions and answers) and users as simple nodes and ignore the rich content information.

Modeling Q-A with Recurrent Neural Network

The first task of our cQA framework is to represent the textual contents of questions and answers with proper semantic embeddings. Since the questions and answers are always sequential data with variant length, in this paper, we construct two different recurrent neural networks, which have been proven beneficial to variant length sequential learning (Mikolov et al. 2013b), to encode the textual contents of questions and answers into fixed length feature vectors, respectively. Given a senquence of words, we firstly represent the t-th word by pre-training word embedding (Mikolov et al. 2013a) as x_t and then use the sequence $\{x_1, x_2, \cdots, x_k\}$ as the input of the corresponding recurrent neural network. As mentioned in (Sutskever, Vinyals, and Le 2014), simple recurrent neural network would be difficult to train due to the resulting long term dependencies. Therefore, we choose long-short term memory (LSTM) (Hochreiter and Schmidhuber 1997) instead of traditional recurrent neural network to learn the embeddings for questions and answers by the following equations:

$$\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\
\hat{C}_t &= tanh(W_c x_t + U_f h_{t-1} + b_f), \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\
C_t &= i_t \cdot \hat{C}_t + f_t \cdot C_{t-1}, \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_o), \\
h_t &= o_t \cdot tanh(C_t).
\end{aligned} \tag{1}$$

where σ represents the sigmoid activation function; Ws, Us and V_o are weight matrices; and bs are bias vectors. The gates in LSTM cell can modulate the interactions between the memory cell itself and its enviroment. The input gate can allow incoming signal to alter the state of the memory cell or block it and the output gate can allow 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 and implementation of LSTM can be found in public website (http://deeplearning.net/tutorial/lstm.html). We take the output of the last LSTM cell, h_k, as the semantic embedding of the input sequence $\{x_1, x_2, \cdots, x_k\}$.

Considering the fact that the questions and answers are always paragraphes with several sentences, we split them into sentences to calculate the semantic embeddings for each sentence by the LSTM described above and then merge the embeddings by an additional max-pooling layer, as shown in Figure 2(c).

Exploiting Heterogeneous Social Networks

A significant characteristic of cQA is the abundant user interaction information introduced from the heterogeneous social networks. For example, in this paper, there are two heterogeneous social networks. The first is a network which consists of the questions, answers and users from a cQA site; and the other is a pure social relationship network whose nodes are the users. This structure is shown in Figure 2(a). Although the traditional content-based approaches (Shen et al. 2015; Qiu and Huang 2015) have made some achievements in cQA, they cannot leverage the valuable social resources to improve the quality of cQA tasks. In order to overcome this defect, we introduce deep random walk (Perozzi, Al-Rfou, and Skiena 2014) method to combine the textual content analysis and social information embedding for cQA.

DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) is an approach for learning latent representations of vertices in a network and these latent representations encode network structural information in a continuous vector space. Formally speaking, for a network $G = (V, E)$, where V is the set of nodes and $E \subseteq (V \times V)$ is the set of edges in G, we want to learn the latent semantic embeddings for each node in V. Inspired by the neural language models (Bengio et al. 2003; Mikolov et al. 2013a), we firstly generate a path from the network G by random walk and then treat this path as a sentence and each node v_i of it as a word. Our goal is to learn an embedding function with which we can properly predict the context of v_i within a window $W = \{v_{i-w}, v_{i-w+1}, \cdots, v_{i+w}\}$ by SkipGram (Mikolov et al. 2013a). This problem can be formulated as an optimization problem as follows:

$$\min_{\Phi} -\log \Pr(\{v_{i-w}, \cdots, v_{i-1}, v_{i+1}, \cdots, v_{i+w}\} | \Phi(v_i)) \tag{2}$$

where Φ is the latent representation matrix of all nodes V and $\Phi(v_i)$ represents the latent embedding of v_i.

However, the original DeepWalk is an unsupervised learning schema which learns the embeddings only from the network structure and ignores the information of edge weights (rating information) and the node contents (content of questions and answers). In our cQA scenario, the textual contents and the answer rating information are very critical for question retrieval and answer recommendation, for example, the rating information can indicate the relative quality rank of different answers w.r.t the same question. To leverage these beneficial supervised information, in this paper, we combine the deep random walk method with the deep Q-A embedding model described in the previous section to boost cQA tasks.

As shown in Figure 2, we firstly generate a walk path P from the heterogeous social networks by random walk and then sample a context window W centered by a certain node v_i. The textual contents of questions and answers in W are then fed into the corresponding LSTM to calculate the latent representation $f(q), f(a)$, respectively. And the user representation $f(u)$ is generated from a learned user embedding matrix. After obtaining $f(q), f(a)$ and $f(u)$, we can calculate the matching score for a pair of question q and answer a by

$$s(q, a) = \sigma(f(q)^T M f(a)) \tag{3}$$

where M is a similarity matrix and σ is sigmoid activation function. All the weight parameters and the similarity matrix M are randomly initialized and trained by the back propogation of training loss.

Taking notice of that there are three different kinds of nodes and to utilize the rating information, for each node v_i in context window W, we design a specific loss function to simultaneously encourage the similarity within W and capture the relative quality diversity between different answers, as follows:

$$l(v_i) = \begin{cases} \sum\limits_{a^+, a^- \in W} max(0, m * \alpha + s(v_i, a^-) - s(v_i, a^+)), \; v_i \in Q; \\ \sum\limits_{q^+, q^- \in W} max(0, m + s(q^-, v_i) - s(q^+, v_i)), \; v_i \in A; \\ \sum\limits_{u \in W - v_i} ||v_i - u||^2, \; v_i \in U. \end{cases} \tag{4}$$

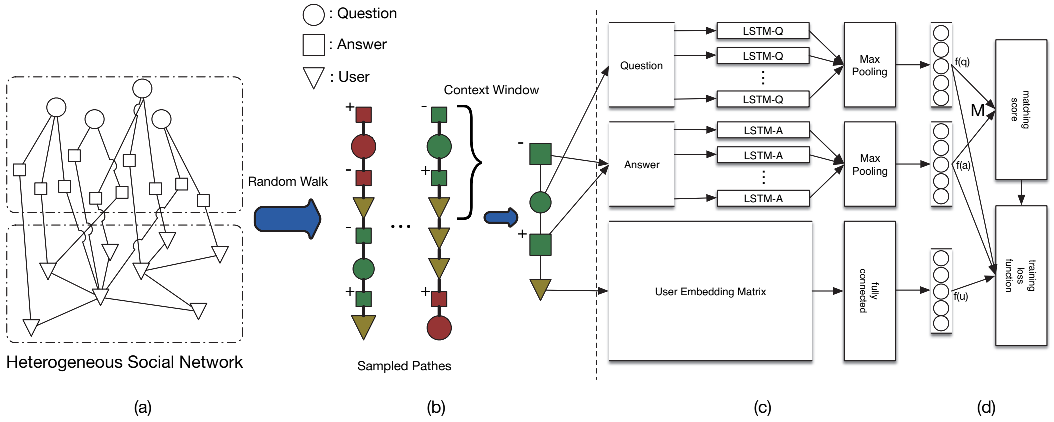

Figure 2: The overview of our proposed framework HSNL. (a) A random walker is walking on the heterogeneous social networks to sample paths of data. (b) For training process, a sliding window select context information including the textual content of questions, answers (different questions and their corresponding answers are marked by different colors) and the user interaction information(e.g., follower relationship and the relative quality rank marked by $^+$ and $^-$). (c) Different kinds of data (questions, answers and users) are encoded into fixed feature vectors by specific models. (d) These feature vectors are used to calculate the matching score between question and answer as output in testing process or obtain the training loss to update the parameters in training process. (best view in color)

where the superscript $^+$ denotes the better matching questions (with higher ratings) or answers and $^-$ denotes the worse ones (with lower ratings); the hyper-parameter $0 < m < 1$ controls the margin in training and Q, A, U are the sets of questions, answers and users, respectively. Taking the different ratings of answers into account, we use a parameter α to scale the margin in the ranking based loss in Equation (4). The α is set to 1 when a^- is a mismatching answer or $0.5 - \frac{r^-}{2*(r^+ + 0.001)}$ for a low-quality answer a^-, where r^+ and r^- are the rating score obtained from the QA community.

Using the framwork described above, we can integrate the social information of heterogeneous networks into the cQA textual content analysis. The trained deep model can calculate QA matching scores directly or output the latent embedding $f(q)$, $f(a)$ and $f(u)$ for other cQA tasks, such as question retrieval or expert finding.

Training

Before training in heterogeneous networks, we firstly pre-train the user embedding matrix (see Figure 2) by DeepWalk (Perozzi, Al-Rfou, and Skiena 2014). After that, we start a walker in the heterogeneous networks to generate paths and the losses of each node in Equation (4) are accumulated as the final training loss. Considering the directed nature of the follower relationships, our walker is only allowed to walk from the followers to their followed user. The reason is that the followers should have some similar preference with their followed users but not vice versa. Denote all the parameters in our model as Θ, the objective function in training process is:

$$\min_{\Theta} L(\Theta) = \sum_{\mathcal{P}} \sum_{\mathcal{W}} \sum_{v_i} l(v_i) + \lambda ||\Theta||_2^2 \tag{5}$$

where $\lambda > 0$ is a hyper-parameter to trade-off the training loss and regularization; $\mathcal{P}, \mathcal{W}$ are the sets of paths and context windows, respectively.

To minimize the objective, we use stochastic gradient descent (SGD) with the diagonal variant of AdaGrad as in (Qiu and Huang 2015). At time step t, the parameter Θ is updated as follows:

$$\Theta_t = \Theta_{t-1} - \frac{\rho}{\sqrt{\sum_{i=1}^{t} g_i^2}} g_t \tag{6}$$

where ρ is the initial learning rate and g_t is the subgradient at time step t. The whole training process is summarized in Algorithm 1.

Algorithm 1 Heterogeneous Social Network Learning for cQA

```
Input: heterogeneous social network $G(V, E)$, walks per node $n$,
    max walk length $t$, number of iterations $m$
Pre-train the user embedding matrix by DeepWalk
for $i = 1$ to $m$ do
    for $j = 1$ to $n$ do
        $\mathcal{O} = shuffle(V)$
        for each $v \in \mathcal{O}$ do
            $p = RandomWalk(G, v, t)$
            Calculate the loss for each node in $p$
        end for
        Accumulate the training loss in Equation (5)
        Update parameters by SGD
    end for
end for
```

Experiments

Dataset Preparation

To empirically evaluate and validate our proposed framework HSNL, a dataset is built up by collecting the crowdsourced data from a popular high-quality community-based question answering system, Quora, and the social relation information from the famous social network site, Twitter.

Quora was launched to the public in June 2010 and has become very successful in just a few years. We have crawled the questions and the corresponding answers posted between September 2012 and August 2013. The ratings (thumbs-up/down count) and user accounts of all the answers are also crawled. In total, we collect 444,138 questions, 887,771 answers and 95,915 users from Quora.

According to the Twitter IDs extracted from the Quora user account, we have crawled their follower relationship from Twitter's social network (Limited by Twitter API, only 500 followers at most can be obtained for single queried ID). After removing the unanswered questions, the user accounts without Twitter IDs and the unaccessable Twitter IDs, we construct a cQA dataset with 252,826 questions, 381,949 answers and 67,185 users. As mentioned in (Zhao et al. 2015), the thumbs-up/down count distribution is a power-law distribution and we normalize the value of ratings of answers to the range of $[0, 100]$. The dataset is split into training set, validation set and testing set without overlapping in our experiments. The size of validation set is fixed as 10% to tune the hyperparameters and the size of training set varies from 20% to 80%. This cQA dataset used in our experiments will be released later.

Evaluation Criteria

Considering that the cQA problem is similar to retrieval or ranking tasks, we evaluate the performance of our proposed HSNL based on three widely-used ranking evaluation metrics, i.e. normalized discounted cumulative gain (**nDCG**) (Shen et al. 2015), **Precision@1** (Qiu and Huang 2015) and **Accuracy** (Zhao et al. 2015).
For the ground truth, we consider all the corresponding answers as the candidate answer set for each question and their received ratings (thumbs-up/down counts) as the ground truth ranking score. The better answer tends to get higher ratings and our task is to predict the relative rank of answers but not the exact thumbs-up/down values. Given the testing question set $\mathcal{Q}$, we denote the predicted ranking order of all the answers for question q by R^q and the answer on i-th position by r_i. The evaluation criterias are introduced as follows:

- **nDCG** The nDCG for ranked answers of question q is given by

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

  where IDCG is the DCG of ideal ordering, $|R^q|$ is the number of ranked answers for question q and rel_i is the relevance between question q and answer r_i which is indicated by thumbs-up/down value. We report the average nDCG for all the questions in $\mathcal{Q}$.

- **Precision@1** This criteria is used to measure the ranking quality of the best answer, given by

$$Precision@1{=}\frac{|\{q{\in}\mathcal{Q}|r_{best}{=}1\}|}{|\mathcal{Q}|}$$

  where r_{best} is the rank of best answer. This criteria computes the average number of times that the best answer is ranked on top by a certain algorithm.

- **Accuracy** Accuracy is normalized by the number of answers for a question, which is given by

$$Accuracy{=}\frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\frac{|R^q|-r_{best}}{|R^q|-1}$$

  where r_{best} has the same meaning as in Precision@1.

In summary, nDCG is the measure for the ranking quality of all the candidate answers while Precision@1 and Accuracy are different measures for the ranking quality of the best answer selected by a certain algorithm.

Baselines and Experiment Settings

In order to demonstrate the efficiency and efffectiveness of our proposed HSNL, six popular supervised and unsupervised algorithms are used in our experiments, incuding some state-of-the-art cQA methods.

- **BOW** Bag-of-words (BOW) is a classical representation for natural language processing tasks. In our experiment, we represent the questions and answers by BOW feature vectors and then calculate the relevant score to rank the candidate answers for each question.
- **LDA** Latent Dirichlet Allocation (LDA) (Celikyilmaz, Hakkani-Tur, and Tur 2010) is a generative model which projects the questions and answers into a latent semantic space and then we can calculates their matching scores in this space.
- **Doc2Vec** Doc2Vec (Le and Mikolov 2014) is a distributed memory and distributed bag of words model which can encode the questions and answers as documents in a low-dimensional continuous feature space. The QA tasks are then conducted in this learned feature space.
- **DeepWalk** DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) learns the representation of data only depending on the graph structure information of social networks.
- **CNTN** CNTN (Qiu and Huang 2015) uses a convolutional neural tensor network with dynamic pooling layers to modeling the questions and answers and calculate the matching score between question and answer.
- **S-matrix** S-matrix (Shen et al. 2015) constructs a similarity matrix to model the complex interaction between questions and answers, then a classical convolutional neural network is introduced to calculate the suitable answer probability.

The first four baselines are unsupervised methods which focus on the representation learning to construct a semantic feature space and the latter two are supervised methods which directly calculate the matching score between questions and answers. In order to better demonstrate the impact of different components of HSNL, we evaluate a simplified

Table 1: Experimental results on nDCG with different propotions of data for training. (best scores are boldfaced)

#Training	20%	40%	60%	80%
BOW	0.6268	0.729	0.7434	0.7704
LDA	0.545	0.7344	0.7902	0.8027
Doc2Vec	0.543	0.7436	0.8039	0.8584
DeepWalk	0.5172	0.7122	0.8025	0.8397
CNTN	0.5367	0.7139	0.8643	0.9101
S-matrix	0.5387	0.7104	0.8721	0.9099
HSNL-sim	0.5721	0.7749	0.8946	0.9191
HSNL	**0.6286**	**0.8249**	**0.9263**	**0.9481**

Table 2: Experimental results on Precision@1 with different propotions of data for training (best scores are boldfaced)

#Training	20%	40%	60%	80%
BOW	0.3634	0.3669	0.445	0.4597
LDA	0.3084	0.3404	0.434	0.4507
Doc2Vec	0.32	0.363	0.4013	0.4205
DeepWalk	0.3289	0.3586	0.3957	0.4065
CNTN	0.2917	0.3704	0.4955	0.5524
S-matrix	0.2854	0.3725	0.5019	0.5755
HSNL-sim	0.325	0.4054	0.5209	0.5909
HSNL	**0.3687**	**0.4426**	**0.5927**	**0.6701**

version of HSNL without the similarity matrix M, named HSNL-sim and the DeepWalk can be regarded as a simplified HSNL without the extensions of Equation (3) and (4). The input words of supervised methods are initialized by pre-calculated word embeddings (Mikolov et al. 2013a) and the weights of neural networks are randomly initialized by a Gaussian distribution with zero mean in our experiments. The hyperparameters and parameters which achieve the best performance on the validation set are chosen to conduct the testing evaluation.

Experimental Results and Analysis

To evaluate the performance of our proposed framework, we conduct several experiments on three metrics described above.

As mentioned previously, we argue that the utilization of social information can mitigate the sparsity problem in cQA tasks. In order to verify this assumption, all the models are trained (BOW needs no training) with different size of training data and tested with the same remaining data (excluding the training and validation data) for evaluation. The content information are utilized by all the methods except DeepWalk, and the social information are introduced by DeepWalk and our proposed HSNL. Table 1, 2 and 3 show the evaluation results in terms of nDCG, Precision@1 and Accuracy, respectively. With these experimental results, we can observe several interesting points:

- With sufficient training data, the supervised methods outperform the unsupervised methods, which suggests that the supervised information, such as the matching scores between questions and answers, are very important for cQA tasks.
- Unsupervised methods are more robust to the training size variance and produce better results when only a few training data are available, because the supervised methods are easily overfitting in this situation.
- Since DeepWalk performs worse than CNTN and S-matrix in most cases, the content analysis plays a more important role than the simple utilization of social information in cQA tasks.
- Comparing with CNTN and S-matrix, our proposed HSNL makes more improvement when the training data is more sparse, which means the additional social information indeed mitigate the sparsity problem of cQA tasks.

Table 3: Experimental results on Accuracy with different propotions of data for training. (best scores are boldfaced)

#Training	20%	40%	60%	80%
BOW	0.3555	0.3839	0.4072	0.4475
LDA	0.2772	0.3384	0.4217	0.4418
Doc2Vec	0.2842	0.3265	0.409	0.4256
DeepWalk	0.2537	0.3163	0.4002	0.4179
CNTN	0.2295	0.3904	0.4531	0.5342
S-matrix	0.2338	0.3833	0.4477	0.5286
HSNL-sim	0.3428	0.4086	0.471	0.5556
HSNL	**0.3645**	**0.4953**	**0.6453**	**0.6928**

- With the increasing of training data, the superiority of HSNL-sim decreases comparing with CNTN and S-matrix, because the advantages of using similarity matrix and introducing social information are balanced out.
- The experimental results of simplified versions of HSNL demonstrate that the random walk method (capturing social information) and the deep learning framework (modeling textual content) are both advantageous for cQA task.
- In all the cases, our HSNL achieves the best performance. This fact shows that the combination of social information and content analysis can further improve the performance of QA matching task.

Conclusion and Future Works

In this paper, we explore the cQA problem from a new perspective of integrating the content analysis and social information exploiting. A novel framework called (HSNL) is presented, which utilizes a random walker to exploit the heterogeneous social networks and encodes different kinds of data with specific deep models. Our work can be easily extended to larger social networks with complicated graph structure since only a little fraction of social graphs is loaded at the same time for training. Moreover, the learned representations for different types of data (e.g., users) can benefit other tasks, such as expert finding in crowdsourcing social networks. Extensive experiments on a large-scale dataset show the effectiveness and efficiency of our proposed framework and demonstrate that the introduction of social information can mitigate the prevalent sparsity problem in cQA tasks.

Acknowledgement

This work was supported in part by the National Basic Research Program of China under Grant 2015CB352300, in part by the China Knowledge Centre for Engineering Sciences and Technology and in part by the Fundamental Research Funds for the Central Universities. The work of Martin Ester was supported in part by NSERC Discovery grant.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research* 3:1137–1155.

Bilotti, M. W.; Elsas, J.; Carbonell, J.; and Nyberg, E. 2010. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 459–468. ACM.

Blunsom, P.; Grefenstette, E.; Kalchbrenner, N.; et al. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

Celikyilmaz, A.; Hakkani-Tur, D.; and Tur, G. 2010. Lda based similarity modeling for question answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, 1–9. Association for Computational Linguistics.

Fang, H.; Lu, W.; Wu, F.; Zhang, Y.; Shang, X.; Shao, J.; and Zhuang, Y. 2015. Topic aspect-oriented summarization via group selection. *Neurocomputing* 149:1613–1619.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, 1606–1611.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*, 2042–2050.

Iyyer, M.; Boyd-Graber, J.; Claudino, L.; Socher, R.; and Daumé III, H. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 633–644.

Jiang, M.; Cui, P.; Chen, X.; Wang, F.; Zhu, W.; and Yang, S. 2015. Social recommendation with cross-domain transferable knowledge. *IEEE Trans. Knowl. Data Eng.* PP(99):1.

Jurczyk, P., and Agichtein, E. 2007. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 919–922. ACM.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1188–1196.

Li, B., and King, I. 2010. Routing questions to appropriate answerers in community question answering services. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 1585–1588. ACM.

Li, B.; Lyu, M. R.; and King, I. 2012. Communities of yahoo! answers and baidu zhidao: Complementing or competing? In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 1–8. IEEE.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 701–710.

Qiu, X., and Huang, X. 2015. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 1305–1311.

Qu, M.; Qiu, G.; He, X.; Zhang, C.; Wu, H.; Bu, J.; and Chen, C. 2009. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th international conference on World wide web*, 1229–1230. ACM.

Shen, Y.; Rong, W.; Sun, Z.; Ouyang, Y.; and Xiong, Z. 2015. Question/answer matching for CQA system via combining lexical and sequential information. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 275–281.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, 1642.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Zhao, Z.; Zhang, L.; He, X.; and Ng, W. 2015. Expert finding for question answering via graph regularized matrix completion. *IEEE Trans. Knowl. Data Eng.* 27(4):993–1004.

Zhou, G.; Liu, Y.; Liu, F.; Zeng, D.; and Zhao, J. 2013. Improving question retrieval in community question answering using world knowledge. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*.