<table>
<thead>
<tr>
<th>Title</th>
<th>Joint implicit and explicit neural networks for question recommendation in CQA services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Tu, Hongkui; Wen, Jiahui; Sun, Aixin; Wang, Xiaodong</td>
</tr>
<tr>
<td>Date</td>
<td>2018</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/47297">http://hdl.handle.net/10220/47297</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2018 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See <a href="http://www.ieee.org/publications_standards/publications/rights/index.html">http://www.ieee.org/publications_standards/publications/rights/index.html</a> for more information.</td>
</tr>
</tbody>
</table>
Joint Implicit and Explicit Neural Networks for Question Recommendation in CQA Services

HONGKUI TU1, JIAHUI WEN2,3,4, AIXIN SUN5, AND XIAODONG WANG1

1National Laboratory of Parallel and Distributed Processing, National University of Defense Technology, Changsha, China
2College of Computer, National University of Defense Technology, Changsha, China
3Airborne Troops Training Base, Guilin, China
4College of Information Science and Engineering, Guilin University of Technology, Guilin, China
5School of Computer Science and Engineering, Nanyang Technological University, Singapore

Corresponding author: Hongkui Tu (tuhkjet@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61070203, Grant 61202484, Grant 61472434, Grant 61572512, and Grant 61502510, in part by the Singapore Ministry of Education Academic Research Fund Tier-2 under Grant MOE2014-T2-2-066, and in part by the National Key Laboratory of High Performance Computing under Grant 20124307120033.

ABSTRACT
Community question answering (CQA) services have emerged as a type of popular social platforms. In the social network, experts provide knowledgeable answers to the questions in their domain of expertise, while celebrities publish influential opinions toward the topics led by some questions. Given the large amount of knowledge organized in the format of question–answers, an interesting research problem is to recommend questions to users so as to maximize their engagements with the platform. However, recommending questions in CQA services is a non-trivial task. Data sources in the CQA services are of different types. It is challenging to incorporate heterogeneous information for the recommendation task. Furthermore, data sparsity is an inherent problem in such platforms. In this paper, we propose a model that is able to jointly model both implicit and explicit information for question recommendation. The model integrates multiple data sources and addresses the problem of data heterogeneity. In the proposed model, we dynamically discover latent user groups and incorporate those hierarchical information to bridge the semantic gaps among users in the shared latent space. We evaluate the proposed model on two real-world datasets, and demonstrate that our model outperforms the state-of-the-art alternatives by a large margin. We also investigate different structures of the proposed model to study the effects of different data sources.

INDEX TERMS
Data mining, neural networks, recommender systems.

I. INTRODUCTION
Lots of users resort to the Internet for searching answers to their questions and satisfying their information needs. Community Question Answering (CQA) websites, such as Quora1 and Zhihu,2 allow users to raise questions to leverage the expertise of other users participating in the community. CQA sites have gained enormous popularity as they can fulfill users’ demands in an easier way.

In CQA, users engage with the sites by posting questions on a certain topic, providing answers to questions in their domain of expertise, or giving comments to answers. As a result, such platforms have accumulated a large amount of texts that covers a wide range of knowledge, from users’ opinions towards a certain topic to knowledge in a specific domain of expertise. Users on such platforms usually need to seek for questions and answers that they are interested in, hence leaves footprints indicating user preferences. However, users often face the problem of information overloading [1], considering the large volume of diversified questions contributed by many users. CQA sites are heavily rely on community participation. Therefore, it is of great significance to present the “right” questions to the “right” users, so as to maximize users’ engagement with the sites.

In this paper, we investigate personalized question recommendation in CQA. Different recommender systems have been shown effective on many online services. However, recommending question in CQA is a non-trivial task, and the challenges are two-folds. First, users’ interest in a question is influenced by many explicit factors. On one hand, users’ preferences to a question is highly connected with the textual content in the question body. On the other hand, the propagation of questions heavily relies on the structure of social networks. Users are more likely to be exposed to the questions

1https://www.quora.com
2https://www.zhihu.com
that their social friends are interacting with. For example, a user knows the questions initiated, upvoted, commented, and focused by her followers. Many existing studies [2]–[4] combine collaborative filtering with social information for recommendation. However, it is still an open problem how to comprehensively incorporate the aforementioned heterogeneous explicit information into a unified model, especially in deep learning models. Second, implicit factors potentially affect users’ interaction behavior. Many studies [5], [6] have demonstrated that considering user grouping improves recommendation accuracy, because users in the same group share similar interests. The grouping information alleviate the problem of data sparseness, by bridging the semantic gap among users in the same group. However, for the CQA question recommendation task, the grouping information is not explicitly available. It is a challenging task to discover the latent user groups.

To address the aforementioned challenges, in this paper, we propose a novel Joint Implicit and Explicit Neural Network (JIE-NN) model for online question recommendation in CQA. JIE-NN incorporates both implicit and explicit information to model user-question interactions.

To extract latent textual features, we employ Convolutional Neural Networks (CNNs) from a given question and the historical texts of a user, and align the user-question pair in the latent semantic space for measuring the semantic similarity between them. For the social connections, we regularize latent feature vectors for social friends, so that users linked with social connections are more likely to have similar interests. In addition to the previous explicit information, we also propose to discover latent user groups, and measure the similarity between the latent feature vector of a user group and that of a question for recommendation. We model users, questions, and user groups in an end-to-end neural network so that all parameters are collaboratively updated, towards the optimization of a pre-defined objective function. On two real-world datasets, we demonstrate the effectiveness of JIE-NN.

To conclude, the main contributions of this papers are as follows:

- We propose a personalized question recommendation model in CQA. The proposed model is flexible to incorporate heterogeneous information sources such as textual content and social connections into an end-to-end neural network. The parameters that extract features from different information sources are updated collaboratively in the network towards the optimization of the loss function.
- We further propose to discover implicit user groups for question recommendation. The process of user group clustering is dynamically iterated with the backpropagation process of the neural network. The similarity measurement between user groups and questions bridges the semantic gap among users in the same group, as they are more likely to have similar interests.
- We evaluate the effectiveness of the proposed model on two real-world datasets. Specifically, we demonstrate that the proposed model outperforms state-of-the-art models on both datasets. Furthermore, we investigate the different variants of the proposed model, and demonstrate that the proposed model is invulnerable to different configurations of hyper-parameters.

II. RELATED WORK

In this section, we review the literature most closely related to our work. Specifically, we review studies on the following: question routing, neural recommendation, joint recommendation, and community detection.

A. QUESTION ROUTING

Our work is similar to question routing [7]–[10] that routes questions to users who can potentially provide answers of high quality. The basic idea is that, in question answering communities, only a few experts amongst the large population of users [11], are likely to make provide good answers to a question. However, most of related studies consider the problem of expert identification, which is to classify experts from numerous users. Then the new questions are routed to them to answer. On the contrary, we recommend questions and their corresponding answers to meet users’ interests. Therefore, the aforementioned studies and ours reside in different layers of the CQA services, and complement each other.

B. NEURAL RECOMMENDATION

In the past decade, deep learning techniques have attracted considerable attention and have been applied in many fields including natural language processing [12]. Deep learning methods are able to extract abstract features automatically through many layers of non-linear transformation [13]. The high-level features are extracted in a way towards the optimization of some pre-defined objective functions [14], and hence the extracted features are highly representative for the data in the learning task. In the area of recommender systems, many studies have tried to combine various neural network structures with collaborative filtering to boost the recommendation accuracy [15]. For example, a few works [16], [17] combine generalized matrix factorization with multi-layer perceptron to capture the interrelations between users and items. The proposed multi-layer non-linear transformation on the user and item latent factors is more expressive than traditional dot product, and is able to discover the hidden factors underlying user-item interactions. Some other studies [18], [19] apply auto-encoders to model user-item interactions, and recommendations are made by reconstructing the user preferences over the items with pre-trained denoising auto-encoders. However, these models are inapplicable to our case as the models are learned solely based on user-item interactions.

C. JOINT RECOMMENDATION

Recently, researchers propose to jointly model the auxiliary information associated with users or items for recommendation. The underlying reason of incorporating additional data
sources is that, they are highly inter-related with user-item interactions, and can help to alleviate the data sparseness and cold-start problem. The data modalities under consideration include texts [12], [20]–[25], images [12], [24]–[28], audio [29], [30] and video [28]. Those observable attributes are processed into abstract representations with popular deep learning techniques (e.g., CNNs, RNNs), and interact them with user or item representations to compensate sparse user-item interactions. However, most of those methods customize the modeling process for data sources of different modalities and tightly couple them with the key collaborative filtering effect, which negatively affects the scalability of the methods. Moreover, when it comes to the interactions between user and item representation, they resort to latent factor model that is vulnerable to data sparsity.

Some publications consider social connections to jointly model user representations based on the idea that users with strong social connections are more likely to have similar interests. Previous studies [31], [32] utilize regularization method to force closely connected users in a social network also located closely to each other in the latent space. In this modeling, the aforementioned heterogeneous information sources are explicitly available. In our work, we make a step further by discovering implicit user group. The latent user grouping information is proved to boost recommendation accuracy, as it bridges semantic gap between users having similar interests. Zhu et al. [33] propose a transfer learning method for recommendation. They incorporate the external knowledge base for alleviating the cold start problem. However, maintaining a knowledge base is of big efforts, especially for some emerging topics.

D. COMMUNITY DETECTION

Since we dynamically cluster users into latent groups, our work is related to community detection [34]. It is the problem of discovering densely connected groups of nodes on a graph. Early works on community detection usually employ different clustering techniques on the graph to discover user cluster. For instance, Tang and Liu [35] apply spectral clustering to social networks for extracting user communities. Recently, embedding techniques is utilized to boost community detection [36], [37]. Most of these works embed nodes in social graphs into low-dimensional vectors, and then apply clustering methods on the embedding representations.

On the other hand, most of previous community detection methods perform node clustering solely based on the link information on social networks. In our case, the user grouping process is carried out collaboratively and iteratively with the semantic matching between user-question pairs. Therefore, our user grouping process is benefited from the collaborative filtering in the learning process, as it constrains users with similar interactions to have similar latent feature vectors. In return, user group information bridges the semantic gap between users in a same group, as they are assumed to have similar interests.

III. PROBLEM FORMULATION

We denote a user-question interaction matrix as \( R \in \mathbb{R}^{M \times N} \), where \( M \) and \( N \) are the number of users and questions, respectively. The non-empty entry \( R_{ij} \) refers to a positive interaction between user \( u_i \in U \) and question \( v_j \in V \). In our case, a positive interaction with a question can be in the form of up voting, down voting, commenting, or providing an answer to the question. In addition, a question \( v_j \) is associated with a text \( x_j \), the content and description of the question. The text associated with each user is the concatenation of all the texts in his/her historical interactions. A social network \( G = (U, E) \) is also available, where \( U, E \) is the set of users and social connections respectively. \((u_i, u_l) \in E \) means there is a positive connection between users \( u_i \) and \( u_l \). Given the interaction matrix \( R \), the set of users \( U \) and items \( V \), the observable texts \((x_j, j = 1, \ldots, N) \) associated with the items and the social network \( G \), our task is to predict the missing values in \( R \).

In this work, we formulate the question recommendation problem as a binary classification problem. That is, for each user-question pair \((u_i, v_j)\) we estimate the posterior probability \( y_{ij} \), which is the probability that \( u_i \) may interact with question \( v_j \). The basic idea is to calculate evidence score for each user-question pair from different information sources. The loss function is defined as the sigmoid cross entropy between the estimated scores and the ground truth.

\[
\begin{align*}
\text{score}_{ij} &= \text{score}^e_{ij} + \text{score}^l_{ij} + \text{score}^g_{ij} \\
\hat{y}_{ij} &= \frac{1}{1 + e^{-\text{score}_{ij}}} \\
\text{loss} &= -\sum (y_{ij}\log \hat{y}_{ij} + (1 - y_{ij})\log(1 - \hat{y}_{ij})) + \text{reg} \quad (1)
\end{align*}
\]

where \( \text{score}^e_{ij}, \text{score}^l_{ij}, \text{score}^g_{ij} \) is the evidence calculated against user-question latent feature vectors, textual information, and group information, respectively. We encode social information into the regularization term, \( \text{reg} \), as it only conveys information from user side. In the following sections, we present how to leverage both implicit and explicit information for question recommendation.

IV. MODEL

We employ deep learning techniques such as CNNs and embedding to model user preferences over items (i.e., questions). Fig. 1 shows the overview architecture of the proposed model. We will introduce detailed information in the next subsections.

A. TEXTUAL INFORMATION

The descriptive texts \( x_j \) associated with a question \( j \) summarizes the overall topics of the question. The underlying reason of modeling textual data sources is that a user’s preference over a question can be explained by the fact that the user is interested in the content of that question. Instead of adopting traditional text processing methods [38], [39], we employ CNNs to extract textual features from question
and user text, as they are demonstrated to be effective for modeling texts [20], [40].

Formally, we transform each text, \( x_j = \{w^n_j\}_{n=1}^{|x_j|} \) where \( w^n_j \) is the \( n \)th word in \( x_j \), into embedding vectors with publicly available tools such as Glove: \( E_j = \{e^n_j\}_{n=1}^{|x_j|} \). The embedding vectors are then fed into one convolution layer and one max pooling layer to obtain a representation for each question. Specifically, for each filter map \( K_j \in \mathbb{R}^{(k \times m)} \), we generate a feature map when moving the filter map through the embedding vectors. The feature map is a vector as shown in eq. (2), where \( k \) is the embedding size and \( m \) is the filter size.

\[
z_t = f(E_{[1:k, t:t+m-1]} \ast K_j + b_j)
\]

\[
f_j = \{z_1, z_2, \ldots, z_{L_u-m+1}\}
\]

where \( \ast \) is convolution operator and \( f \) is an activation function (i.e., ReLU). We then apply max pooling operation over each feature map to extract the most important feature, as presented in eq. (3). In practice, we apply filter maps of various sizes to the embedding vectors for extracting features of different local patterns.

\[
o_j = \max\{f_j\}
\]

\[
h'_j = \{o_1, o_2, \ldots, o_s\}
\]

where \( s \) is a hyper-parameter of our model, denoting the number of feature map after applying CNNs and max-pooling on the embedding vectors. One of the advantage of CNNs for modelling texts is that the dimension of hidden textual representations (i.e., \( h'_j \)) is determined by the number of feature maps rather than the text length. The readers are referred to [40] for more details of the text modelling process.

Similarly, we obtain a text representation, \( h'_i \), for each user, where the text associated with each user is the concatenation of all the question texts in his/her historical interactions. Note that we have textual representations for each user-question pair, we can interact \( h'_i \) with \( h'_j \) to capture the semantic similarity between each user-question pair. However, unlike [40], the representations of users and questions are from different feature spaces and not comparable [20]. Existing method [20] leverages Factorization Machine to model all nested variable interactions of the concatenated textual representation pair, but it consumes too much memory. Therefore, we propose an element-wise feature interaction method that aligns the informative parts of the interactions for measuring representation similarity.

\[
\text{score}^t_{ij} = b' + \sum_{m,n} w_{m,n} h'_i m h'_{j,n} = b' + \sum W \circ (h'_i \cdot h'_j)^T
\]
where $\circ$ is the element-wise multiplication, $W \in \mathbb{R}^{b' \times b'}$ models the element-wise interactions, and $b'$ is the bias.

### B. LATENT INFORMATION

User generated texts reflect his/her preference and texts associated with items indicate their characteristics. However, the features learned from texts are basically biased to the textual semantic and cannot be generalized to reflect users’ interests. One possible solution is to utilize embedding and deep learning techniques to model user and item latent feature vectors. In our case, each user and question is projected into a vector with dimension $D$ through indexing embedding matrices, denoted by $u_i, v_j \in \mathbb{R}^{D \times 1}$ respectively. Notice that latent feature vectors, $u_i$ and $v_j$, capture user interests and item characteristics. The interaction between them models the user preferences over items.

To deeply model the interactions between each user-question pair, we merge $u_i$ and $v_j$ into a joint vector, and then feed the vector into a multi-layer neural network. Existing works [16], [17] show that non-linear transformation in the multi-layer neural network is able to model user-item latent structure, and it outperforms the widely used matrix factorization that learns the linear interaction between user-item pairs.

The latent information in this work is modelled as follows:

$$m_{ij} = u_i \oplus v_j \oplus u_i^T v_j \oplus u_i \circ v_j$$

$$a_1 = \text{relu}(W_1^TM_{ij} + b_1)$$

$$\ldots$$

$$a_{L-1} = \text{relu}(W_{L-1}^T a_{L-2} + b_{L-1})$$

$$\text{score}_{ij} = W_L^T a_{L-1} + b_L$$

(5)

where $\oplus$ is the concatenation operation, $W_L, b_L$ are the parameters of the $L^{th}$ layer. $L$ is 3 in our case. We employ rectified linear unit $\text{relu}(x) = \max(0, x)$ as this non-linear transfer function is non-saturated [16] and simpler than others.

### C. SOCIAL INFORMATION

In CQA services, users’ interaction with questions are visible to their followers. Hence, social relations play an important role in question recommendation as users are more like to be exposed to the questions that their followers are interacted with. Therefore, users with social connections are more likely to interact with the same questions. As user latent feature vectors reflect user interests, social friends are supposed to be in a proximity closed to each other in the shared latent space. Therefore, modelling social information is to propagate users’ interests along the social links, and regularize social friends to have similar latent feature vectors. We define social matrix $F \in \mathbb{R}^{M \times M}$ as follows.

$$F_{il} = \begin{cases} 1 & \text{if } (u_i, u_l) \in E, \\ |F_{il,:}| & \text{otherwise} \end{cases}$$

(6)

where $|F_{il,:}|$ is the normalization factor for user $u_i$, and $\lambda_s$ is the weight for controlling the tradeoff between recommendation accuracy and social regularization. The basic idea of introducing the normalization factor is that the fewer friends a user has, the more centralized she is exposed to her friends’ interactions, hence the more similar is her latent feature vector to that of her friends. The social information is modelled as in eq. (7)

$$\text{reg}_{social} = \lambda_s \sum_{(u_i, u_l) \in E} (u_i - u_l)^2$$

(7)

In practice, we train the proposed model on batch basis, hence the regularization is performed with user data available in the batch. We denote $E_u \in \mathbb{R}^{D \times M}$ and $U_b \in \mathbb{R}^{\text{batch size} \times M}$ as the user embedding matrix and one-hot user encodings in a batch.

$$\text{reg}_{social} = \lambda_s (E_uU_b^T - (E_uF)U_b^T)^2$$

(8)

where $E_uU_b^T$ retrieve embeddings for the users in the batch, $(E_uF)U_b^T$ first average embeddings of respective social friends for each user, and then retrieve the averaged embeddings for the users in the batch.

### D. GROUP INFORMATION

In CQA services, some users may have the same interests and constitute a group. The group information can be regarded as a high-level representation for the users in the group. Incorporating the hierarchical structure of user and group for question recommendation can bridge the semantic gap amongst the users in the same group. However, the group information is not explicitly available. In this paper, we propose to discover group information collaboratively with the learning of user latent representations. Specifically, we cluster user latent representations into group centroids, and interact them with question latent representations to model group interests. We denote the group id for user $u_i$ as $g_i$. The clustering process aims to minimize the objective defined in eq. (9)

$$\arg\min_{\mu_k} \sum_{k=1}^{K} \sum_{g_i = k} ||u_i - \mu_k||^2$$

(9)

where $\mu_k$ is the centroid for group $k$, and $K$ is the pre-specified group number. The parameters $\{g_i\}_{i=1}^M$ and $\{\mu_k\}_{k=1}^K$ are iteratively updated as in eq. (10).

$$g_i = \arg\min_{k} ||u_i - \mu_k||^2$$

$$\mu_k = \sum_{g_i = k} u_i / I(g_i = k)$$

(10)

where $I(x) = 1$ if $x$ holds, and 0 otherwise. Instead of using $\{u_i\}_{i=1}^K$ as the group representations, we introduce an embedding matrix $E_g \in \mathbb{R}^{D \times K}$ for representing groups, and regularize the difference between $E_g$ and $\{u_i\}_{i=1}^K$, which is a commonly used relaxation method [41]. We denote the one-hot group encoding for user $u_i$ with $E_{g_i} \in \mathbb{R}^{K \times 1}$, which consists of $0$s in all dimensions with the exception of a single $1$ in $g_i^{th}$ dimension used to uniquely identify the group id for user $u_i$. Then we obtain the group latent dense representation for the user through indexing the group embedding matrix.

$$\theta_{g_i} = E_g \cdot E_{g_i}$$

(11)
where $\theta_{g_i}$ is the group representation for user $u_i$. Similar to Section IV-B, we apply deep neural network to model the interactions between user groups and items. Same as in eq. (5), $L = 3$.

$$z_{ij} = \theta_{g_i} \oplus v_j \oplus \theta_{g_i}^T v_j \oplus \theta_{g_i} \circ v_j$$

$$a_1 = \text{relu}(W_1^T z_{ij} + b_1)$$

$$\ldots \ldots$$

$$a_{L-1} = \text{relu}(W_{L-1}^T a_{L-2} + b_{L-1})$$

$$\text{score}^g_{ij} = W_L^T a_{L-1} + b_L$$  \hspace{1cm} (12)

The group latent representation captures general interests of the users in the group, and the interactions between user groups and items model user preferences over items in a high-level latent space. This hierarchical structure of user and group representations help to bridge the semantic gaps amongst the users since users in the same group share the same group representation. In our work, the group representations are obtained by indexing the group embedding matrix $E_g$ which is independent of user representations. However, group representations are high-level abstracts of user representations, and need to be related to user latent feature vectors. Therefore, we regularize the differences between group representations and user cluster centroids to enforce them to be similar to each other.

$$\text{reg}_{\text{group}_1} = \lambda_{g_1} \sum (\theta_{g_i} - \mu_{g_i})^2$$

$$\text{reg}_{\text{group}_2} = \lambda_{g_2} \sum (\theta_{g_i} - u_i)^2$$  \hspace{1cm} (13)

Combining eq. (4), eq. (5), eq. (7), eq. (12) and eq. (13) together, we have the object function.

$$\text{score}_{ij} = \text{score}^l_{ij} + \text{score}^g_{ij} + \text{score}^c_{ij}$$

$$\hat{y}_{ij} = \frac{1}{1 + e^{-\text{score}_{ij}}}$$

$$\text{loss} = -\sum_{i,j}(y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij}))$$

$$+ \lambda_{g_1} \sum_{i}(u_i - \sum_{(u_i, u_i) \in E} F_{il} u_i)^2$$

$$+ \lambda_{g_2} \sum_{i}(\theta_{g_i} - \mu_{g_i})^2$$

$$+ \lambda_{g_3} \sum_{i}(\theta_{g_i} - u_i)^2$$

$$+ \lambda_g \sum_{j}||v_j||^2 + \lambda_u \sum_{i}||u_i||^2 + \lambda_k \sum_{k}||\theta_k||^2$$  \hspace{1cm} (14)

E. INFERNECE

With the constructed objective function, its local minimum can be targeted by taking the derivative with respect to the parameters which are updated along the gradient direction. The derivatives of the objective function with respective to the parameters are shown as follows:

$$\frac{\partial \text{loss}}{\partial u_i} = \sum_{ij}(\hat{y}_{ij} - y_{ij}) \frac{\partial \text{score}^l_{ij}}{\partial u_i}$$

$$+ \lambda_s (u_i - \sum_{(u_i, u_i) \in E} F_{il} u_i)$$

$$- \lambda_s \sum_{(u_i, u_i) \in E} (u_i - \sum_{(u_i, u_i) \in E} F_{il} u_i)$$

$$- \lambda_{g_2} (\theta_{g_i} - u_i)$$

$$+ \lambda_{g} u_i$$

$$\frac{\partial \text{loss}}{\partial v_j} = \sum_{ij}(\hat{y}_{ij} - y_{ij}) (\frac{\partial \text{score}^g_{ij}}{\partial v_j} + \frac{\partial \text{score}^c_{ij}}{\partial v_j})$$

$$+ \lambda_v v_j$$

$$\frac{\partial \text{loss}}{\partial \theta_{g_i}} = \sum_{ij}(\hat{y}_{ij} - y_{ij}) \frac{\partial \text{score}^g_{ij}}{\partial \theta_{g_i}}$$

$$+ \lambda_{g_1} (\theta_{g_i} - \mu_{g_i})$$

$$+ \lambda_{g_2} (\theta_{g_i} - u_i)$$

$$+ \lambda_{g} \theta_{g_i}$$

$$\frac{\partial \text{loss}}{\partial \Theta} = \sum_{ij}(\hat{y}_{ij} - y_{ij}) \frac{\partial \text{score}^c_{ij}}{\partial \Theta_{c}}, c \in \{t, l, g\}$$  \hspace{1cm} (15)

where $\frac{\partial \text{score}^c_{ij}}{\partial \Theta_{c}}, c \in \{t, l, g\}$ is the derivatives of matching scores with respect to the parameters in the neural networks proposed in Section IV-A, Section IV-B and Section IV-D. The matching scores consist of matching signals of heterogeneous information sources in the latent space. Therefore, the derivatives are able to propagate information from one space to another, and end up with a unified end-to-end model that is able to exploit multiple data sources collaboratively for the recommendation task. The derivatives can be obtained with chain rule, and we omit them here due to space limitation.

In this work, we perform $T_2$ iterations of clustering for every $T_1$ batches of parameters learning. Unlike traditional clustering algorithm, the input user representations after every $T_1$ batches of parameters learning are different. However, we do not perform clustering after every batch of parameters updating, as this incurs high computational overhead. For the same reason, we do not cluster user representations until converges every time after model training. We study hyper parameters $T_1$ and $T_2$ later in experiments.

V. EXPERIMENT

A. DATA SET

To evaluate the effectiveness of the proposed model, two real-world datasets are collected from Quora and Zhihu. We use crawlers to collect a month’s data from July 2017 to August 2017. Inspired by the previous works [42] and [43], we filter out the users with fewer than 5 interactions. The statistics of the two datasets are reported in Table 1. On average, on the Quora dataset, we have 11.4 interactions for each
we mix it with 500 random items, and then rank the ground
leave-one-out scheme. For each ground truth item for a user,
we have 24.05 interactions for each user and 9 users for each
question. While for Zhihu dataset,
we are able to exploit explicit and implicit information from
lines with a large margin. The reason lies in the fact that
tions between user-item pairs, it suffers from data sparseness
traditional matrix factorization and deeply models the interac-
tions amongst users and items. Even though NeuMF generalizes
DeepCoNN, textual information help bridge the semantic gap
between users and items. By predicting contexts
with user/item embedding, users with similar contexts
are forced to be similar in the latent space.

DeepCoNN [20]: Instead of embedding users and items,
it uses user-generated and item-generated texts to rep-
resent users and items. It utilizes CNNs for extract-
ing user and item textual representations and employ
Factorization Machine to model first- and second-order
interactions between a user-item representation pair.
We select these state-of-the-art recommendation models
as baselines, considering item attributes and regularization.
For some baselines like NeuMF, we fine tune the parameters
(e.g., learning rates and batch normalization) to achieve their
best results, while for others we set the parameters to the
values suggested in the original papers. We reimplemented
the DeepCoNN method, and adopted the open source code
released by the authors of NeuMF and PACE.

B. SET UP
We set the hyper-parameters of the proposed model to the
following default values: for the regularization terms in
the loss function, we set the parameters $\lambda_s = \lambda_u =
\lambda_k = 0.0001, \lambda_x = \lambda_{g1} = \lambda_{g2} = 0.001$; for the texts
associated with the items, we initialize the embedding matrix with
Glove and tune it during the training process; for parameters
in CNNs processing the texts, we set the number of filters to
be 100 for each of the filter size amongst [1,2,4]; for the training
configuration, we set the initial learning rate to 0.0001.
The batch size is set to 1024, and the model is trained for a
maximum of 200 epochs. For each user, we randomly sample
50 missing interactions as negative samples for each positive
sample. In addition, for each user 70% of the respective pos-
itive and negative samples are used for training. The default
number of user group is 35. User clustering is performed for
20 iterations for every 100 batches of collaborative learning,
\textit{i.e.}, $(T_1, T_2) = (100, 20)$ in Algorithm 1. In addition to the
aforementioned default values, we also evaluate the proposed
model with different network structures.

As for the validation process, we adopt the widely used
leave-one-out scheme. For each ground truth item for a user,
we mix it with 500 random items, and then rank the ground
truth together with the 500 items and measure the recommen-
dation performance of the proposed model. Finally, we apply
two commonly used metrics, including precision@k and
recall@k. All experiments are conducted on Nvidia GPU
gtx 1070 with 8G memory.

C. BASELINES
We compare the proposed model with the following three
state-of-the-art baselines.

- NeuMF [16]: It generalizes matrix factorization with
end-to-end neural network. It embeds users and items
into latent space and employs multi-layer neural network
to model their interactions. The embedding of users and
items are learnt jointly with the objective function in the
model.

- PACE [31]: In addition to the modelling of user-item
interactions, it introduces contexts to bridge the semantic
gap between users and items. By predicting contexts
with user/item embedding, users with similar contexts
are forced to be similar in the latent space.

- DeepCoNN [20]: Instead of embedding users and items,
it uses user-generated and item-generated texts to rep-
resent users and items. It utilizes CNNs for extract-
ing user and item textual representations and employ
Factorization Machine to model first- and second-order
interactions between a user-item representation pair.

D. COMPARISON PERFORMANCE
The results of precisions and recalls on both datasets are
presented in Fig. 2 and Fig. 3. From the figures we make the
following observations.

First, both DeepCoNN and PACE outperform NeuMF
significantly, demonstrating the benefit of incorporating
additional information sources for modelling user prefer-
ences. For PACE, user representations are used to predict
their contexts, and hence users share similar contexts are
implicitly regularized to have similar representations. As for
DeepCoNN, textual information help bridge the semantic gap
amongst users and items. Even though NeuMF generalizes
traditional matrix factorization and deeply models the interac-
tions between user-item pairs, it suffers from data sparseness
without auxiliary information sources.

Second, the proposed model JIE-NN outperforms all bas-
elines with a large margin. The reason lies in the fact that
we are able to exploit explicit and implicit information from

\footnotesize{\begin{verbatim}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{Dataset} & \textbf{User} & \textbf{Question} & \textbf{Interactions} & \textbf{Sparsity} \\
\hline
\texttt{Quora} & 7761 & 19068 & 89442 & 99.940\% \\
\texttt{Zhihu} & 10928 & 29222 & 267791 & 99.917\% \\
\hline
\end{tabular}
\caption{Statistics of datasets used in our work.}
\end{table}

\end{verbatim}
\normalsize

\end{document}

heterogeneous data sources. The data sources complement each other and both benefit the recommendation task. For example, compared with PACE, we are able to incorporate textual information and discover hierarchical user preferences to model user interests over the items. Compared with DeepCoNN, we nonlinearly model the interactions between user and item latent representations. On average, our model constantly yields 10% relative improvement on dataset Quora and 18% on dataset Zhihu across the metrics.

Finally, all models achieve better results on Zhihu dataset than that on Quora dataset. One reason is that the interaction matrix of the former is denser than that of the later. Since all models are based on user-item interactions, more historical footprints help capture users’ preferences. For the same reason, the proposed model outperforms baselines with a larger margin on the first dataset.

E. STRUCTURE STUDY

We now study the performance of the different variants of the proposed model. Specifically, we evaluate the effect that each information source has on the performance of JIE-NN. Each variant is trained with one data source absent, and the variants are denoted as JIE-NN/c, where $c \in \{t, s, g\}$. For example, JIE-NN/t is the proposed model trained without textual information.

The performance of different variants on two datasets are presented in Table 2 and Table 3. Overall, information of each domain has its own contribution to the recommendation accuracy, as JIE-NN constantly yields higher precisions and recalls than the variants trained with one information.
source absent. On Zhihu dataset, group information plays the most important role as JIE-NN achieves the highest improvement over JIE-NN/g that is learned without discovering user latent group. On Quora dataset, an exception is with the model JIE-NN/s. Without social regularization, the model performs even worse than the baselines. The reason is that the model is not able to provide enough evidences for user grouping without social information, as social regularization and user grouping mutually benefit each other: social regularization enforces socially linked users to have similar representations, user grouping drives similar users into the same cluster. However, JIE-NN does not experience performance deterioration on Zhihu because richer interactions can help to model user preferences through the collaborative filtering process, and hence benefit user grouping. For validation, we visualize user clusters resulted from JIE-NN/s on Quora in Fig. 4. Observe that users in small group are cohesive, while users in larger groups are more scattered; distances between users in a same group are not necessarily shorter than that between users from different groups.

F. PARAMETER STUDY

In this subsection, we investigate the impact of hyperparameters in the proposed model. We first study the effect that the number of user groups. A group representation indicates the general preference of the users in the group, and intra-group users share more similar interests than inter-group users. However, coarse-grained user clustering may result in
loss of discriminative information among group representations, as group representations are not representative for users with different preferences. While fine-grained user clustering cannot bridge similar users in the latent space, as similar users are not necessarily in the same group in this case. The effects of group number on the recommendation performance are shown in Fig. 5 and Fig. 6. The recommendation performance varies with different group numbers, but the fluctuations are constantly within the percentage of 5%. Therefore, the proposed model JIE-NN is invulnerable to the hyper-parameter group number, and it is able to discover discriminative user groups with different structures.

In the proposed model, user clustering is iteratively performed with the collaborative filtering process. We perform $T_1$ iterations of user clustering for every $T_2$ iterations of collaborative filtering as shown in Algorithm 1. The effects of hyper-parameters $T_1$, $T_2$ on the recommendation performance are presented in Fig. 7 and Fig. 8. Observe from the figures that, different settings of $T_1$, $T_2$ have negligible influence on the recommendation performance except that low $T_1$ degrades both precision and recall of the proposed model. The reason of the performance decrement is that users are clustered frequently with respect to the collaborative filtering, and early user clustering can result in local optima for JIE-NN.

G. GROUP VISUALIZATION

The user groups discovered by the proposed model has many applications such as link prediction and group recommendation. We visualize the user clustering results on the two datasets using t-SNE [44] in Fig. 9. Different groups are indicated in different colors. We observe that user grouping on Quora is more scattered than that on Zhihu. This is consistent with the data statistics that Zhihu has richer interactions, and the proposed model has sufficient data to model user preferences and capture latent group.

FIGURE 7. Precision and recall of JIE-NN with different $T_1$ and $T_2$ on dataset Quora. (a) $T_1$, (b) $T_2$.

FIGURE 8. Precision and recall of JIE-NN with different $T_1$ and $T_2$ on dataset Zhihu. (a) $T_1$, (b) $T_2$.

FIGURE 9. Group virtualization for Quora and Zhihu. (a) Quora. (b) Zhihu.
VI. CONCLUSION
In this paper, we propose a recommender model that exploits explicit and implicit information for question recommendation in CQA services. Specifically, we leverage heterogeneous explicit information sources to collaboratively learn user and item representations. Meanwhile, we discover user latent groups for modelling hierarchical user interests. The collaborative learning and user clustering are performed iteratively, and they mutually benefit each other. Experiments on two real-world datasets demonstrate that the proposed model outperforms state-of-the-art alternatives significantly. We also study different variants of the proposed model to evaluate the effect that each information source has on the recommendation accuracy. Furthermore, we examine the impact of the hyper-parameters and demonstrate our model is able to maintain similar precisions and recalls with different settings.

In CQA services, there are many other types of information that can be incorporated for question recommendation, such as answerers and user profiles. Besides, we will try to apply some recent clustering methods [45]–[47] to discover user groups and adopt other deep learning model [48] for textual processing. We leave them for future work.

ACKNOWLEDGMENT
Hongkui Tu and Jiahui Wen contributed equally to this work. The authors would like to thank Prof. Xiaohui Cheng for his constructive suggestions to revise the paper.

REFERENCES


