2013

CQARank: Jointly Model Topics and Expertise in Community Question Answering

Liu YANG
Singapore Management University, liuyang@smu.edu.sg

Minghui QIU
Singapore Management University, minghui.qiu.2010@smu.edu.sg

Swapna GOTTOPATI
Singapore Management University, SWAPNAG@smu.edu.sg

Feida ZHU
Singapore Management University, fdzhu@smu.edu.sg

Jing JIANG
Singapore Management University, jingjiang@smu.edu.sg

See next page for additional authors

DOI: https://doi.org/10.1145/2505515.2505720

CQARank: Jointly Model Topics and Expertise in Community Question Answering

Liu Yang[†,‡], Minghui Qiu[‡], Swapna Gottipati[‡], Feida Zhu[‡], Jing Jiang[‡], Huiping Sun[†], Zhong Chen[†]
[†] School of Software and Microelectronics, Peking University, China
[‡] School of Information Systems, Singapore Management University, Singapore
yang.liu@pku.edu.cn,{minghui.qiu.2010, swapnag.2010, fdzhu, jingjiang}@smu.edu.sg
{sunhp,chen}@ss.pku.edu.cn

ABSTRACT

Community Question Answering (CQA) websites, where people share expertise on open platforms, have become large repositories of valuable knowledge. To bring the best value out of these knowledge repositories, it is critically important for CQA services to know how to find the right experts, retrieve archived similar questions and recommend best answers to new questions. To tackle this cluster of closely related problems in a principled approach, we proposed Topic Expertise Model (TEM), a novel probabilistic generative model with GMM hybrid, to jointly model topics and expertise by integrating textual content model and link structure analysis. Based on TEM results, we proposed CQARank to measure user interests and expertise score under different topics. Leveraging the question answering history based on long-term community reviews and voting, our method could find experts with both similar topical preference and high topical expertise. Experiments carried out on Stack Overflow data, the largest CQA focused on computer programming, show that our method achieves significant improvement over existing methods on multiple metrics.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering, selection process, retrieval models* ; H.3.5 [**Information Storage and Retrieval**]: On-line Information Services—*Web-based services*

General Terms

Algorithms, Experimentation, Performance

Keywords

Community Question Answering; Latent Topic Modelling; Gaussian Mixture Model; Expert Recommendation; Link Analysis

1. INTRODUCTION

The recent boom of Web 2.0 has seen the emergence and flourishing of many knowledge sharing community services such as Wikipedia[1], Stack Overflow[2] and Quora[3]. The huge success of the concept of Community Question Answering(CQA), which enables people to post questions and answers in various domains, drives home the enormous power of online community activities in satisfying users' professional and personal knowledge quest.

However, existing question answering mechanism in CQA sites still falls short of users' expectation for several reasons: (1) **Poor expertise matching**: A new question, in many cases, may not find its way to the right people with the best-matching interest and ability to answer it, resulting in suboptimal answers and prolonged latency. (2) **Low-quality answers:** CQA sites may contain low-quality answers such as mischievous answers and spams [16]. These answers often receive low ratings or voting from community members. (3) **Under-utilized archived questions:** Many questions from different users are in fact similar. Before posting a new question, a user may benefit from browsing related archived questions and their answers first. Not surprisingly, these issues are closely related. In fact, a common fundamental question underlying all these tasks is how to model *topics* and *expertise* in CQA sites.

Previous research efforts along this line include expert user mining [35; 5], relevant answer retrieval [2; 16] and similar question finding [33; 28; 15]. In this paper, our contribution is to push the research frontier along two dimensions: (1) Horizontally, we propose to jointly model topics and expertise in a unified framework; and (2) Vertically, we achieve better understanding of both user topical interest and expertise by leveraging tagging and voting information, important pieces of information that have so far been neglected in the modeling.

Our Contributions

First, to the best of our knowledge, we propose the first extensive study to jointly model topics and expertise. Traditionally, topics and expertise have been modeled separately. On one hand, for topics, latent topic models such as LDA[4], when applied to CQA, can measure the semantic similarity between questions and answers, and thus help find relevant answers or related questions given a new question. They can also model a user's topical interests based on the user's posting history, and hence match users and questions based on their topical similarity. On the other hand, for expertise, each user's ability in answering questions can be modeled as an expertise level, by which we can better recommend candidate answerers. By modeling the relationships among users, questions and answers in CQA as a linked network, existing work often relies on link analysis techniques such as PageRank[24] and HITS[19] to find authoritative users.

[1] http://www.wikipedia.org/
[2] http://www.stackoverflow.com/
[3] http://www.quora.com/

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505720.

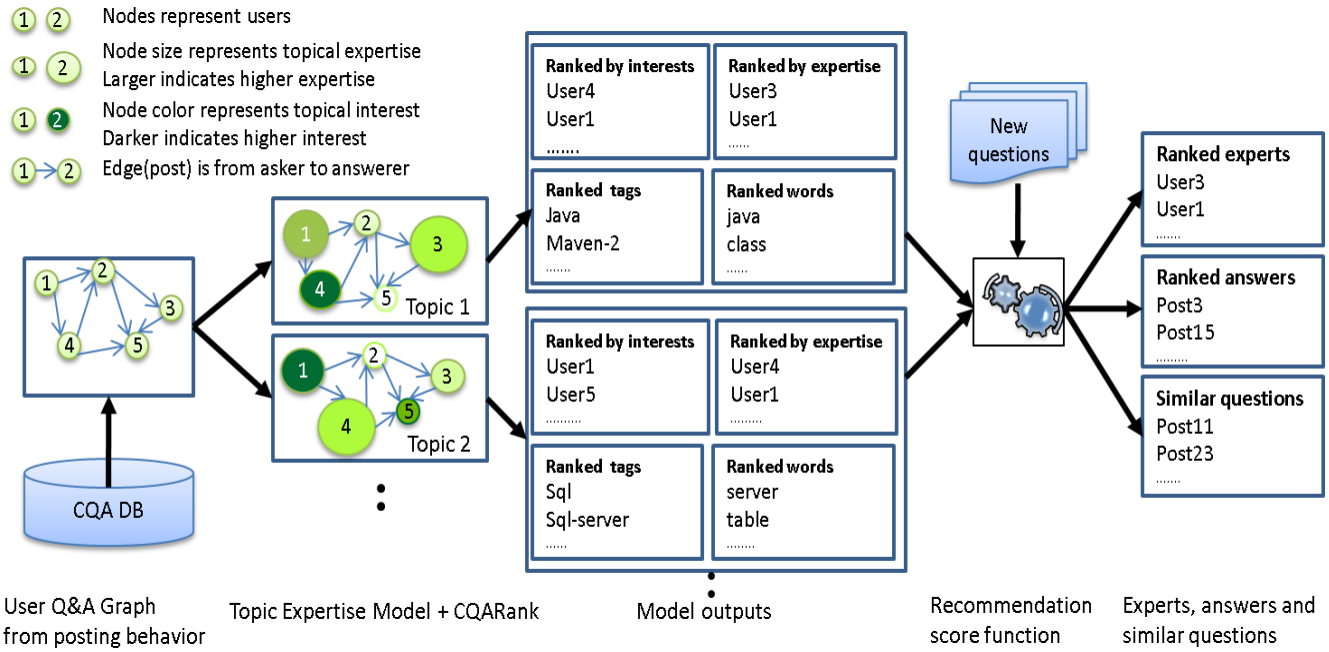


Figure 1: CQARank Recommendation Application Framework. TEM models text with tagging and voting information for user followed by CQARank which combines learning results from TEM with link structure analysis for Q&A graph to discover topical interest and expertise. For new questions, the outputs from the method are processed along with the question to generate ranked experts, answers and similar questions.

Despite their success in each aspect, there is evidently a strong need in real-life CQA services to integrate these two aspects together to enhance user experience. After all, no one is expert in all topical interests, which means one's expertise level should be evaluated with respect to the corresponding topics. On other hand, every new question falls into some particular topics, and they should be routed to answerers interested in those particular topics with the right level of expertise. We take both user topical interest and expertise evaluation into our model, enabling our method to find experts with both similar topical preference and matching topical expertise.

Secondly, we achieve better understanding of both user topical interest and expertise by leveraging tagging and voting information. Since both topics and expertise are latent factors, i.e. we do not directly observe their values from CQA sites, existing work solve their inference based on their textual content and the linkage structure among them. However, we notice that two important types of information have not been well utilized: (I) *Tagging information* — Tags are important user-generated category information for many Q&A communities, e.g., technical forums, that achieves fine-grained and dynamic topic representation. Users who use a particular tag when posting questions or answers might prefer topic summaries most relevant to that tag[29]. Consequently, incorporating tags of questions and answers into textual content aids in better discovery of user topical interest. (II) *Voting information* — Votes indicate a CQA community's long term review for a given user's expertise level under a specific topic [1]. Users with high expertise tend to receive high votes for their Q&A posts. This motivates us to exploit the votes for a user given specific topics to model user topical expertise.

We propose a probabilistic Topic Expertise Model (TEM) which uses *tagging information* to help learn topics and a Gaussian mixture hybrid to model *voting information*. Based on the model results of TEM, we propose CQARank, an extension of PageRank algorithm, to aggregate user topical expertise base on Q&A link structures, combining both textual content model results and link structure to simultaneously measure user topical expertise and interests.

Finally, we perform a thorough experimental study on a large real data set from Stack Overflow, the largest CQA focused on computer programming. The evaluation results show that CQARank achieves significant improvement over existing methods on multiple metrics.

Roadmap. The rest of our paper is organized as follows. We give our method overview in Section 2. In Section 3 we define several important notations and present our Topic Expertise Model for jointly modelling user topical interest and expertise. In Section 4 we propose CQARank to combine both textual content model results in Section 3 and link structure to estimate user expertise and interests under various topics. Section 5 is a systematic experimental analysis using real data from Stack Overflow. Section 6 is on related work and we conclude our study in Section 7.

2. METHOD OVERVIEW

In this section, we provide an overview of our method referred as CQARank, which is shown in Figure 1. We first introduce some concepts.

User: We use *user* to refer to the askers and answerers in CQA. Table 1 shows a snapshot of typical Q&A posts with votes and tags in Stack Overflow. Every question has a tag set assigned by the asker. Both questions and answers have vote scores given by users in CQA. Users can vote-up or vote-down posts. The value of vote score equals the difference between times of vote-up and vote-down.

Topical Interest: We use *Topical Interest* to refer to user preference for specific topics in CQA. For example, some users prefer to post content related to "Java", while others are more interested in "database".

Topical Expertise: We use *Topical Expertise* to refer to their level of expertise on specific topics in CQA. Different users have dif-

Questions: **What statistics should computer scientists know?** *Tags*: **Statistics, Computer Scientist, NLP, R, Performance** *Votes*: **45** *I've tried to look into learning more statistics, but I've gotten a bit lost. What kind of problems in programming and computer science are statistical methods well suited for? I've found some lists of books. Where should I start?*
Answers
User A: **Answer**: Interesting question. As a statistician whose interest is more and more aligned with computer science perhaps I could provide a few thoughts ... *Votes*: **99** User B: **Answer**: Just as a point, not as a critic, your question should be formulated in a different way: "what statistics should any person know?" ... *Votes*: **54** User C: **Answer**: I have not much to add. What caught my attention is the preface, where the author refers to a common dissatisfaction to those who approach the study of statistics: ... *Votes*: **15** User D: **Answer**: My short answer is this: latent variable statistics, including both structural equation modelling and finite mixture modelling. These cover an impressive number of statistical models.... *Votes*: **0**

Table 1: Sample Q&A posts with tags and votes in Stack Overflow.

ferent topical expertise. Moreover, one user could have different expertise levels for different topics. For example, a user may be a guru for the "Java" topic but a novice for "Matlab".

Q&A Graph: We use *Q&A Graph* to refer to the network based on user posting behavior in CQA. Nodes denote users and a directed edge exists between two users if one of them has answered questions by the other, where the edge direction is from the asker to the answerer.

We first construct a Q&A graph from user posting behavior in CQA corpus. We then jointly model Q&A textual content with votes and tags using our probabilistic Topic Expertise Model. Finally, we apply our CQARank to combine learning results from TEM with link analysis of Q&A graph to discover user topical interests and expertise. For each topic, different users exhibit different topical interests and expertise in Q&A graph, so we get user lists ranked by their interests and expertise. We also have top tags and words for each topic as model results. For new questions, using recommendation score functions, we process the model outputs along with the question to generate ranked experts, answers and similar questions.

In Section 3, we will explain in detail how we jointly model topics and expertise in CQA with a generate probabilistic model with GMM hybrid. In Section 4, we will present CQARank which combines learning results of TEM with link structure analysis to make recommendations for given new questions. The recommendation score function for each output is explained in Section 5.3.

3. TOPIC EXPERTISE MODEL

3.1 Model

We now present Topic Expertise Model(TEM) to jointly model user topical interests and expertise. Table 2 shows the set of notations and descriptions of our model parameters.

In our model, the user "topical expertise" e is the level of knowledge and ability of a user u under a topic z. To model this information, we assume there exist E expertise levels, each with a Gaussian distribution on vote scores. The reason why we choose Gaussian distribution is that it is with a high range of scores, and the expertise level can be reflected by looking at mean of its corresponding Gaussian distribution. Specifically, a high expertise level is often associated with high vote scores which can be modeled by a Gaussian distribution with high mean. On the contrary, a low expertise level is with a Gaussian distribution with low mean. To model user topical expertise, we assume each user u has an expertise level distribution on each topic z, denoted as $\phi_{z,u}$. In this case, if this user is an expert in topic z, the probability proportions $\phi_{z,u}$ will have high values for expertise levels which correspond to Gaussian distributions with high mean.

Notations	Descriptions
U	the total number of users
N_u	the total number of Q&A posts of user u
$L_{u,n}$	the total number of words in u's n-th post
$P_{u,n}$	the total number of tags in u's n-th post
K	the total number of topics
E	the total number of expertise levels
T	the total number of unique tags
V	the total number of unique words
μ	mean of Gaussian distribution
Σ	precision of Gaussian distribution
w, t, v, e, z	label for word, tag, vote, expertise, topic
$\mathbf{W}, \mathbf{T}, \mathbf{V}, \mathbf{E}, \mathbf{Z}$	vector for words, tags, votes, expertise, topics
θ_u	user specific topic distribution
$\mathcal{N}(\mu_e, \Sigma_e)$	expertise specific vote distribution
ψ_k	topic specific tag distribution
φ_k	topic specific word distribution
$\phi_{k,u}$	user topical expertise distribution
$\alpha, \beta, \eta, \gamma$	Dirichlet priors
$\alpha_0, \beta_0, \mu_0, \kappa_0$	Normal-Gamma parameters
$\mathcal{NG}(\alpha_0, \beta_0, \mu_0, \kappa_0)$	Normal-Gamma distribution

Table 2: Notations and descriptions.

For each Q&A post, we observe its vote, multiple words and tags. We assume that each post has latent variables e and z, which denote the expertise and topic of this post respectively. For each Q&A post of a given user u_i, topics are generated from a user specific topic distribution θ_u and its expertise is generated from the user topical expertise distribution $\phi_{z,u}$. For each topic z, words are generated from a topic specific word distribution φ_z and tags are generated from a topic specific tag distribution ψ_z. Note that we assume tags of answers are the same with the corresponding question. For each expertise e, votes are generated from an expertise specific Gaussian distribution $\mathcal{N}(\mu_e, \Sigma_e)$ with Normal-Gamma distribution priors. The E expertise specific Gaussian distributions compose a Gaussian Mixture Model (GMM) component for modeling the generation of votes. The other distributions are Multinomial distributions with symmetric Dirichlet priors. The plate notation is in Figure 2.

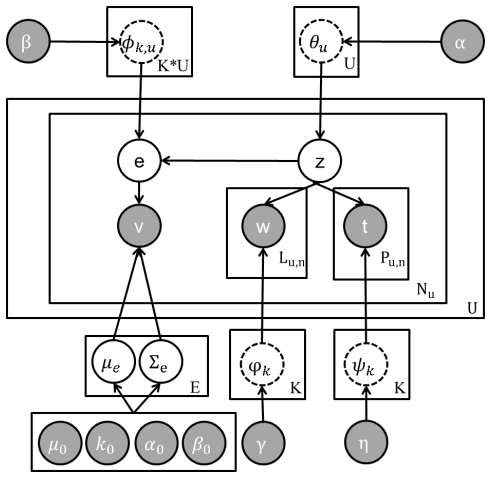


Figure 2: The plate notation of Topic Expertise Model for user topical interests and expertise discovery in CQA. Dashed variables will be collapsed out in Gibbs Sampling.

The generative process of Q&A posts of users can be described as follows:

- For the u-th user, $(u = 1, 2, \cdots, U)$
  - Draw a user specific topic distribution $\theta_u \sim \text{Dir}(\alpha)$
- For the e-th expertise, $(e = 1, 2, \cdots, E)$
  - Draw an expertise specific vote distribution $\mathcal{N}(\mu_e, \Sigma_e) \sim \mathcal{NG}(\alpha_0, \beta_0, \mu_0, k_0)$
- For the k-th topic, $(k = 1, 2, \cdots, K)$
  - Draw a topic specific tag distribution $\psi_k \sim \text{Dir}(\eta)$
  - Draw a topic specific word distribution $\varphi_k \sim \text{Dir}(\gamma)$
    - For the u-th user, $(u = 1, 2, \cdots, U)$
      - Draw a user topical expertise distribution $\phi_{k,u} \sim \text{Dir}(\beta)$
- For the u-th user $(u = 1, 2, \cdots, U)$
  - For the n-th post $(n = 1, 2, \cdots, N_u)$
    - Draw topic $z \sim \text{Multi}(\theta_u)$
    - Draw expertise $e \sim \text{Multi}(\phi_{z,u})$
    - Draw vote $v \sim \mathcal{N}(\mu_e, \Sigma_e)$
    - For the l-th word $(l = 1, \cdots, L_{u,n})$
      - Draw word $w \sim \text{Multi}(\varphi_z)$
    - For the p-th tag $(p = 1, \cdots, P_{u,n})$
      - Draw tag $t \sim \text{Multi}(\psi_z)$

3.2 Learning and Parameter Estimation

We use collapsed Gibbs sampling to obtain samples of the hidden variable assignment and estimate the model parameters of TEM. The Gibbs Sampling process is described in Algorithm 1.

Algorithm 1 Gibbs Sampling for TEM.

```
procedure GibbsSampling
    Initialize Z and E by assigning random values
    for each Gibbs Sampling iteration do
        for each user u = 1, ..., U do
            for u's n-th QA post, n = 1, ..., N_u do
                Let c denotes {u, n}
                Update (μ_{e_c,¬_c}, Σ_{e_c,¬_c}) according to Eqn. 4
                Draw z_c and e_c according to Eqn. 1
                Update (μ_{e_c}, Σ_{e_c}) according to Eqn. 4
            end for
        end for
    end for
    Estimate model parameters θ, ψ, φ and ϕ
end procedure
```

We jointly sample topic $z_{u,n}$ and expertise $e_{u,n}$ for each user u and post n, where we assume $(\mu, \mathbf{\Sigma})$ for all the expertise levels are known. Let c denotes $\{u, n\}$, Θ denotes all the Dirichlet priors and Normal-Gamma priors, we can drive the Gibbs update rule for $z_{u,n}$ and $e_{u,n}$ as follows:

$$
\begin{aligned}
& p(z_c = z, e_c = e | \mathbf{Z}_{\neg_c}, \mathbf{W}, \mathbf{E}_{\neg_c}, \mathbf{V}, \mathbf{T}, \Theta) \\
\propto\ & \frac{p(\mathbf{Z}, \mathbf{W}, \mathbf{E}, \mathbf{V}, \mathbf{T} | \Theta)}{p(\mathbf{Z}_{\neg_c}, \mathbf{W}, \mathbf{E}_{\neg_c}, \mathbf{V}, \mathbf{T} | \Theta)} \\
=\ & \frac{\Delta(C^{\mathbf{k}}_u + \alpha)}{\Delta(C^{\mathbf{k}}_{u,\neg_c} + \alpha)} \cdot \frac{\Delta(C^{\mathbf{w}}_z + \gamma)}{\Delta(C^{\mathbf{w}}_{z,\neg_c} + \gamma)} \cdot \frac{\Delta(C^{\mathbf{t}}_z + \eta)}{\Delta(C^{\mathbf{t}}_{z,\neg_c} + \eta)} \\
& \cdot \frac{\Delta(C^{\mathbf{e}}_{z,u} + \beta)}{\Delta(C^{\mathbf{e}}_{z,u,\neg_c} + \beta)} \cdot \mathcal{N}(v_c | \mu_e, \Sigma_e) \\
=\ & \frac{C^z_{u,\neg_c} + \alpha}{\sum_{k=1}^{K} C^k_{u,\neg_c} + K\alpha} \cdot \frac{\prod_{w=1}^{V} \prod_{i=1}^{n^w_c} (C^w_{z,\neg_c} + \gamma + i - 1)}{\prod_{j=1}^{n^{\mathbf{w}}_c} \sum_{w=1}^{V} (C^w_{z,\neg_c} + V\gamma + j - 1)} \\
& \cdot \frac{\prod_{t=1}^{T} \prod_{p=1}^{n^t_c} (C^t_{z,\neg_c} + \eta + p - 1)}{\prod_{q=1}^{n^{\mathbf{t}}_c} \sum_{t=1}^{T} (C^t_{z,\neg_c} + T\eta + q - 1)} \\
& \cdot \frac{C^e_{z,u,\neg_c} + \beta}{\sum_{e=1}^{E} C^e_{z,u,\neg_c} + E\beta} \cdot \mathcal{N}(v_c | \mu_e, \Sigma_e), \qquad (1)
\end{aligned}
$$

where $\Delta(\cdot)$ is a "Dirichlet delta function" which can be seen as a multidimensional extension to beta function [14], $\mathcal{N}(\cdot)$ is Gaussian distribution.

To estimate parameters (μ_e, Σ_e) for an expertise level e, we need to consider all the votes associated with e and derive the posterior distribution. We report the derived formula in the following, one can refer to [9; 23] for the detailed derivations.

$$
\begin{aligned}
& p(\mu_e, \Sigma_e | \mathbf{v_i}_{e_i=e}, \Theta) \\
\propto\ & p(\{\mathbf{v_i}\}_{e_i=e} | \mu_e, \Sigma_e) \cdot \mathcal{NG}(\mu_k, \Sigma_k | \mu_0, \kappa_0, \alpha_0, \beta_0) \\
=\ & \prod_{v:\{\mathbf{v_i}\}_{e_i=e}} \mathcal{N}(v | \mu_e, \Sigma_e) \cdot \mathcal{NG}(\mu_k, \Sigma_k | \mu_0, \kappa_0, \alpha_0, \beta_0) \\
=\ & \mathcal{NG}(\mu_e, \Sigma_e | \mu'_e, \kappa'_e, \alpha'_e, \beta'_e), \qquad (2)
\end{aligned}
$$

where $\mu'_e, \kappa'_e, \alpha'_e, \beta'_e$ are defined as follows:

$$
\begin{aligned}
\mu'_e &= \frac{\kappa_0 \mu_0 + n_e \bar{\mathbf{v}}_e}{\kappa_0 + n_e}. \\
\kappa'_e &= \kappa_0 + n_e. \\
\alpha'_e &= \alpha_0 + \frac{n_e}{2}. \\
\beta'_e &= \beta_0 + \frac{1}{2} \sum_{v:\{\mathbf{v_i}\}_{e_i=e}} (v - \bar{\mathbf{v}}_e)^2 + \frac{\kappa_0 n_e (\bar{\mathbf{v}}_e - \mu_0)^2}{\kappa_0 + n_e}. \qquad (3)
\end{aligned}
$$

where $\bar{\mathbf{v}}_e$ is the average vote score for expertise e, n_e is the total number of votes with expertise level e.

Given Eqn. 2 and Eqn. 3, we can update (μ_e, Σ_e) as follows:

$$
\begin{aligned}
\mu_e &= \mu'_e. \\
\Sigma_e &= \frac{\alpha'_e}{\beta'_e}. \qquad (4)
\end{aligned}
$$

With Gibbs Sampling, we can make the following parameter estimation:

$$\theta_{u,k} = \frac{C^k_u + \alpha}{\sum_{k=1}^{K} C^k_u + K\alpha}. \quad \text{user-topic distribution} \qquad (5)$$

$$\psi_{k,t} = \frac{C^t_k + \eta}{\sum_{t=1}^{T} C^t_k + T\eta}. \quad \text{topic-tag distribution} \qquad (6)$$

$$\varphi_{k,w} = \frac{C^w_k + \gamma}{\sum_{w=1}^{V} C^w_k + V\gamma}. \quad \text{topic-word distribution} \qquad (7)$$

$$\phi_{k,u,e} = \frac{C^e_{k,u} + \beta}{\sum_{e=1}^{E} C^e_{k,u} + E\beta}. \quad \text{user topical expertise distribution} \qquad (8)$$

4. CQARANK FOR TOPICAL EXPERTISE MEASURE

TEM is a latent variable model for modeling textual contents and voting information to discover user topical interests and expertise. It does not make use of user network structure built from user Q&A graph. However, user network structure will be helpful for topical expertise learning because users who provide answers to high expertise level users tend to also be with a high expertise. Inspired by this intuition, we consider to extend PageRank to measure user topical expertise. The expertise of users under a specific topic in CQA can be interpreted as the "authority" of web pages in hyperlink environment. We propose CQARank to combine user topical interests and expertise learning results in TEM with link structure to enforce user topical expertise learning. CQARank could find experts not only with similar topical interests, but also with high topical expertise based on Q&A voting history in communities.

First of all, we construct Q&A graph $G = (V, E)$ in CQA. V is a set of vertex representing all users. E is a set of directed edges.

An edge exists between two users if one of them answers questions of the other. The direction is from the asker to the answerer. For edge $e = (u_i, u_j)$ where $u_i \in V, u_j \in V$. The weight $w_{i,j}$ is the number of all answers provided by u_j for questions of u_i.

A random surfer on Q&A graph G visits each user vertex with random walk and teleportation operation, which results in a unique distribution of steady-state visiting probabilities. To let the random surfer visits user nodes with higher topical expertise and interest with larger probability, we incorporate the results from TEM into the transition matrix and teleportation vector computation of CQARank. Given a topic z, the transition probability of a random surfer from asker u_i to answer u_j is defined as:

$$P_z(i \to j) = \begin{cases} \frac{w_{i,j} \cdot sim_z(i \to j)}{\sum_{k=1}^{|V|} w_{i,k} \cdot sim_z(i \to k)} & \text{if } \sum_m w_{i,m} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $sim_z(i \to j)$ is the similarity between u_i and u_j under topic z, which is defined as:

$$sim_z(i \to j) = 1 - \left|\theta'_{i,z} - \theta'_{j,z}\right| \quad (10)$$

θ' is row normalized $U \times K$ matrix learnt as user specific topic distribution in TEM.

The transition matrix $\mathbf{M}$ is defined as:

$$\mathbf{M}_{i,j} = P_z(i \to j) \quad (11)$$

In this definition, the more u_j answer questions of u_i, the higher expertise u_j will gain, which corresponds to a higher transition probability from u_i to u_j. Also u_j is more likely to answer questions of u_i if they share similar topical interests.

Given topic z, the CQARank saliency score $\mathbf{R}_z(u_i)$ of u_i can be formulated in a recursive manner as follows:

$$\mathbf{R}_z(u_i) = \lambda \sum_{j:u_j \to u_i} \mathbf{R}_z(u_j) \cdot \mathbf{M}_{i,j} + (1-\lambda) \cdot \theta_{u_i,z} \cdot \mathbb{E}_{z,u_i}(\mu) \quad (12)$$

where $\lambda \in [0, 1]$ is a damping factor to control the probability of teleportation and random walk. $\theta_{u_i,z}$ is the estimated user topical interest score of u_i under topic z, which is learnt as user specific topic distribution in TEM. μ is the mean of the expertise specific vote Gaussian distribution. $\mathbb{E}_{z,u_i}(\mu)$ is the estimated user topical expertise score of u_i under topic z, which is defined as the expectation of user topical expertise distribution as follows:

$$\mathbb{E}_{z,u_i}(\mu) = \sum_{e=1}^{E} \phi_{z,u_i,e} \cdot \mu_e \quad (13)$$

Thus $\theta_{u_i,z} \cdot \mathbb{E}_{z,u_i}(\mu)$ defines the teleportation vector of the random surfer under topic z in CQARank. In original PageRank algorithm, the random surfer teleport to all nodes with the equivalent probability $1/V$ where V is total number of vertex in graph. [35] propose a Topic-Sensitive PageRank for expert finding method which incorporates user topical interest into teleportation vector computation. Tapping the value of excellent Q&A performance based on community voting information, we take both user topical interest and expertise into definition of teleportation vector, which enable the random surfer tend to teleport to user nodes with both similar topic preference and professional topic expertise.

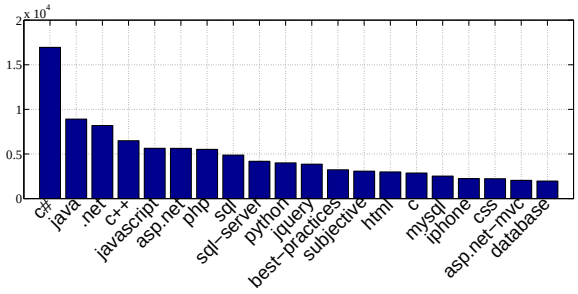
Note that we can estimate each user's topical expertise score by just using TEM results with Eqn. 13. However, for CQARank, we measure user topical expertise by the final saliency score $\mathbf{R}_z(u_i)$ when the iterated algorithm converges. The advantage of which is to combines results from TEM and link analysis of Q&A graph to further improve the user topical expertise discovery. We design recommendation experiments in Section 5.3 to compare performance of CQARank and TEM to reveal the effectiveness of incorporating Q&A graph information.

5. EXPERIMENTS

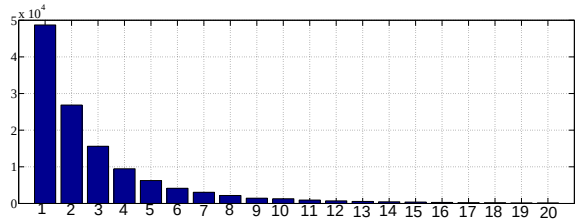
5.1 Data Set and Experiment Settings

We use real data from Stack Overflow for experiments. Stack Overflow is the most popular question answering community focusing on computer programming. The data of Stack Overflow is publicly available through Creative Commons Data Dump Service[4]. We download the complete dataset of two years which is from its launch in August 2008 to August 2010. We select all posts in three months from May 1^{st} 2009 to August 1^{st} 2009 and then use all the posts of users who have asked and answered no fewer than 80 times for the training of TEM. In training data, we have $8,904$ questions and $96,629$ answers posted by 663 users. The data set contains $85,527$ unique words, $10,689$ unique tags and 135 unique votes. Our testing data for expert users and answers recommendation experiments in Section 5.3 is all posts of the same set of users in training data from August 2^{nd} 2009 to April 29^{th} 2010. So training and testing data do not have overlap. We remove testing questions which have no, or only one, answer. The testing data set contains $1,173$ questions and $9,883$ answers. For data preprocessing, we tokenize text and discard all code snippets. Then we remove the stop words and HTML tags in text.

The most frequent tags and votes with their counts in training data set are shown in Figure 3. We observe votes count distribution is a power-law distribution which means most votes are relatively small.



(a) Top frequent tags.



(b) Top frequent votes.

Figure 3: The most frequent tags and votes and their counts in training data set. We only visualize positive votes in this figure.

[4] http://blog.stackoverflow.com/category/cc-wiki-dump/

For all experiments, we empirically set Dirichlet hyperparameters $\alpha = 50/K, \beta = 0.01, \gamma = 0.01, \eta = 0.001$ according to suggestions in [10]. For Norma-Gamma parameters, we set μ_0 as the mean of votes from our data set, κ_0 as 1, α_0 as 1, and β_0 as the mean distance between randomly sampled 1000 votes. We run TEM with 500 iterations of Gibbs sampling. With some trails on the number of topics and expertise, we set topic number $K = 15$, expertise number $E = 10$ as they provide meaningful topics and vote Gaussian distributions for our data set. For damping factor in CQARank, we set $\lambda = 0.2$ after we conduct multiple experiments to determine the best value of it from 0.1 to 0.9.

5.2 TEM Results

5.2.1 Topic Analysis

In this section, we illustrate top tags and words for 10 randomly selected topics discovered by TEM in Table 3 and Table 4. We observe clean top words and tags for each topic. Moreover, top words have strong correlation with top tags under the same topic. For example, top tags in topic 6 are about "version control", corresponding to which TEM discovered topic words like "git", "repository", "branch", "version", "control", "commit", etc. which are frequently mentioned by users when they talk about this topic. Furthermore, top tags like "career-development", "best-practices", "iphone-sdk", "memory-management", etc. provide phrase level instead of bag-of-words features to distill richer and better interpreted topic information from Q&A text.

5.2.2 Expertise Analysis

One of our motivations in this work is to model user topical expertise. Recall that TEM learns different user expertise levels by clustering votes using GMM component. The mean and precision of different expertise specific vote Gaussian distributions learnt by TEM are shown in Table 5. First, we observe 10 Gaussian distributions with various means ranging from 0.40 to 40.17 for the generation of votes in data. The mean of each Gaussian distribution can be used to denote expertise score for each expertise level. Based on this notation, we can estimate user topical expertise score according to Eqn. 13. Secondly, the higher the mean, the lower the precision. The variance becomes larger when the mean goes higher, which aligns with the power-law vote count distribution in Figure 3.

5.3 Recommendation for New Questions

One important task in CQA sites is to make "recommendations" for new questions, the idea of which is to either direct questions to the right expert users or answers, or to find similar questions for the asker to further explore similar answers. In particular, the three important tasks studied in CQA sites are the following: (1)To recommend expert users [35; 11; 5; 8](2)To find answers [2; 16], and (3)To find similar questions to new questions [33; 28; 15; 31]. In this section, we discuss how our model tackles these tasks.

5.3.1 Recommend Expert Users

The first task we consider is to recommend expert users where our aim is to find users who can provide answers with high vote scores for a given question, i.e. users with high expertise for the question. Note that this setting is different from related works like [11] which treats all actual answerers for questions in testing data as the ground-truth since they mainly model user topical interests and try to recommend responders for questions. For our experiments, we want to recommend users who not only would like to respond to the question, but also have real expertise to provide high quality answers. So in our experiments, all methods evaluated would find expert users and give the rank for each user in the recommendation list. We evaluate the rank list with ground truth from the answerer rank list ordered by votes in testing data.

Task: Given a question q and a set of test users $\mathcal{U}$, the target is to rank all these users by their interests and expertise to answer the question q. We score each user u by considering user topic similarity with the question $\text{Sim}(u, q)$ and user expertise in the question $\text{Expert}(u, q)$, where the intuition is that if the user is interested and have a high expertise for the question, then the user tends to provide a good answer wining high votes. The recommendation score function is defined as follows:

$$\begin{aligned} S(u,q) &= \text{Sim}(u,q) \cdot \text{Expert}(u,q) \\ &= (1 - \text{JS}(\theta_u, \theta_q)) \cdot \sum_{z=1}^{Z} \theta_{q,z} \cdot \text{Expert}(u,z) \quad (14) \end{aligned}$$

where $\text{Expert}(u, z)$ is the expertise of user u under topic z. For TEM, we compute it according to Eqn. 13. For CQARank, we set it as the final saliency score $\mathbf{R}_z(u_i)$ when CQARank achieves convergence. θ_q is the question's topic distribution and $\text{JS}(\cdot)$ is JS-divergence distance.

Note that θ_u and $\phi_{z,u}$ can be obtained from our model results. θ_q need to be estimated by computing its posterior probabilities. Specifically, we compute $\theta_{q,z}$ as follows:

$$\begin{aligned} \theta_{q,z} &\propto p(z|\mathbf{w}_q, \mathbf{t}_q, u) \\ &= p(z|u)p(\mathbf{w}_q|z)p(\mathbf{t}_q|z) \\ &= \theta_{u,z} \sum_{w:\mathbf{w}_q} p(w|z) \sum_{t:\mathbf{t}_q} p(t|z) \\ &= \theta_{u,z} \sum_{w:\mathbf{w}_q} \varphi(z,w) \sum_{t:\mathbf{t}_q} \psi(z,t) \quad (15) \end{aligned}$$

where $\mathbf{w}$ and $\mathbf{t}$ are the set of all the words and tags in question q. Here $\theta_{u,z}$, $\varphi(z, w)$ and $\psi(z, t)$ can be obtained from our model results. After we score each user in $\mathcal{U}$, we rank them in decreasing order of the score.

Baselines: To evaluate the effectiveness of CQARank, we compare against some previous related works including probabilistic topic models, link analysis techniques and mixture methods combining both as follows:

- **TEM:** This is our method without link analysis part.
- **TSPR:** [35] proposed a Topic-Sensitive PageRank method for expert finding in CQA. They consider link structures and topical similarity between users. $\text{Sim}(u, q)$ is based on the result of LDA and $\text{Expert}(u, q)$ is the TSPR saliency score.
- **UQA:** [11] proposed a User-Question-Answer Model for modeling of Q&A text. The category in their model is similar to tags in TEM. However, their model sets single category variable for words in each post while TEM permits multiple tags for each post. $\text{Sim}(u, q)$ is based on result of UQA model and $\text{Expert}(u, q)$ is set as 1 since UQA does not model user topical expertise.
- **PageRank(PR):** This method finds expert users with only link structure analysis using standard PageRank algrithms[24]. $\text{Expert}(u, q)$ is the PR saliency score and $\text{Sim}(u, q)$ is set as 1 since PR does not model latent topics.
- **InDegree(ID):** This method rank users by the number of best answers provided by them, as described by [5].

Evaluation Criteria: For ground truth, we consider all the answerers for each question q as the target user set $\mathcal{U}$, and their averaged votes for each question are the ground truth vote scores -

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
java	c++	sql	python	css	svn	subjective	iphone	security	c#
eclipse	c	sql-server	linux	html	version-control	career-development	objective-c	c#	.net
spring	windows	mysql	windows	javascript	git	best-practices	cocoa-touch	encryption	visual-studio
maven-2	visual-c++	tsql	bash	jquery	mercurial	language-agnostic	iphone-sdk	php	asp.net
ant	visual-studio	database	perl	internet-explorer	tortoisesvn	project-management	cocoa	.net	visual-studio-2008
tomcat	linux	sql-server-2005	beginner	web-development	best-practices	learning	uikit	asp.net	sharepoint
jar	c#	php	unix	asp.net	visual-studio	design	xcode	cryptography	windows
jsp	delphi	database-design	vim	xhtml	tfs	jobs	uitableview	email	vb.net
j2ee	winapi	query	php	div	visual-sourcesafe	java	memory-management	authentication	c++
hibernate	gcc	oracle	java	best-practices	beginner	software-engineering	core-animation	java	iis

Table 3: Top tags for different topics discovered by TEM.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
java	library	table	command	css	git	time	view	user	project
class	files	query	line	html	files	software	method	server	application
spring	dll	sql	files	element	repository	make	object	password	web
jar	compiler	data	script	div	svn	project	class	key	files
eclipse	function	index	run	page	branch	programming	controller	data	visual
project	windows	column	directory	text	version	design	set	address	net
application	header	key	windows	width	control	development	make	hash	studio
files	make	rows	python	browser	changes	find	methods	security	windows
maven	source	database	output	image	source	job	objects	client	server
build	functions	tables	shell	elements	commit	problem	data	users	version

Table 4: Top words for different topics discovered by TEM.

expert answerers tend to get more votes. Note that out task is not to predict the exact vote of each user but rank them in terms of votes.

We use the commonly used nDCG measure to evaluate model results, which is defined as follows

$$n\text{DCG@}K = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\frac{\sum_{j=1}^{K}\frac{1}{\log_2(j+1)}score(\mathcal{M}_{q,j})}{IdealScore(K,q)},$$

where $\mathcal{Q}$ is the set of questions, $\mathcal{M}_{q,j}$ is the j-th expert generated by method $\mathcal{M}$ for question q, $score(\mathcal{M}_{q,j}) = 2^{v(\mathcal{M}_{q,j})} - 1$, where $v(\mathcal{M}_{q,j})$ is the ground truth score for the expert $\mathcal{M}_{q,j}$, and $IdealScore(K,q)$ is the ideal ranking score of the top K experts of question q.

We also adopt Pearson and Kendall rank correlation coefficients which are two of the most frequently used correlation measures between ranked variables as metrics. We compare rank lists of expert users by all methods with rank list in ground truth and then use correlation coefficients to measure the strength of correlation between the two rank lists.

*Results:*The results of expert user recommendation for new questions are presented in Table 6. We summarize our observations as follows: (1) CQARank and TEM perform well in the task. Especially looking at nDCG, both methods achieve at least a score of 0.89. (2) In terms of correlation based criteria, CQARank can provide a user rank list with higher correlation coefficient with the ground truth rank list than all the other rivaling methods. (3) CQARank significantly outperforms TSPR, which shows the advantage of considering vote and tag information for user topical expertise discovery. (4) The result of CQARank is better than TEM, which proves the effectiveness of considering Q&A link structure to enforce the expertise learning. Overall, in this expert users recommendation task, our method significantly outperforms all baseline methods, with at least 10%significance level by Wilcoxon signed rank test.

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
CQARank	**0.5858**‡	**0.7991**†	**0.8941**†	**0.1905**†	**0.1738**†
TEM	0.5757	0.7826	0.8920	0.1720	0.1429
UQA	0.4650	0.7548	0.8547	-0.0606	-0.0498
TSPR	0.4790	0.7551	0.8611	-0.0136	-0.0138
PR	0.5078	0.7875	0.8729	0.0575	0.0621
ID	0.5492	0.7710	0.8727	0.0920	0.0858

Table 6: Results on expert user recommendation for new questions. ‡ means the result is better than others except TEM in the same column at 5% significance level measured by Wilcoxon signed rank test and † is at 10% level.

5.3.2 Recommend Answers

The second task we consider is to recommend answers for a given question. Our task is defined as follows.

Task: For a given question q and a set of answers $\mathcal{A}$, each method needs to rank all the answers in $\mathcal{A}$. Similar to expert ranking task, we score each answer by considering its similarity to the question and the expertise of the answerer. Similar to Eqn. 14, we define the recommendation score function as:

$$\begin{aligned} S(a,q) &= \text{Sim}(a,q)\cdot\text{Expert}(u,q) \\ &= (1-\text{JS}(\theta_a,\theta_q))\cdot\sum_{z=1}^{Z}\theta_{q,z}\cdot\text{Expert}(u,z) \quad (16)\end{aligned}$$

	Expertise 1	Expertise 2	Expertise 3	Expertise 4	Expertise 5	Expertise 6	Expertise 7	Expertise 8	Expertise 9	Expertise 10
Mean	40.17	10.42	6.07	4.39	3.25	2.38	1.75	1.46	1.14	0.40
Precision	3.03E-04	1.97E-02	4.48E-02	1.07E-01	1.11E-01	2.43E-01	4.57E-01	5.92E-01	6.51E-01	3.14E+00

Table 5: Mean and precision of different expertise specific vote Gaussian distributions learnt by TEM.

Note that θ_a and θ_q can be learnt from Eqn. 15.

Baselines: The baselines we consider for this task is the same as the task in Section 5.3.1.

Evaluation Criteria: We use each answer's vote as its ground truth score. The metrics used here are the same as in Section 5.3.1.

Results: We present the results in Table 7. We observe similar trends as in expert recommendation. (1) CQARank and TEM show good results in the task, in terms of the correlation based criteria. CQARank provides an answer rank list with higher correlation coefficient with the ground truth rank list than all the comparing methods. (2) CQARank significantly outperforms TSPR in terms of all criteria, at least 10%significance level by Wilcoxon signed rank test, which shows the advantage of considering vote and tag information for user topical expertise discovery. (3) We find in this task, to consider Q&A link structure is important as link analysis based approaches achieve better results than topic analysis based approach. Our method also shows clear advantage over TEM. Overall, in this task, CQARank outperforms all baseline methods.

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
CQARank	0.4748	**0.7857**†	**0.8194**†	**0.1644**‡	**0.1421**‡
TEM	0.4253	0.7830	0.8080	0.1289	0.1131
UQA	0.4010	0.7293	0.7661	-0.0840	-0.0709
TSPR	0.4007	0.7576	0.7924	0.0186	0.0091
PR	**0.5196**†	0.7791	0.8107	0.0718	0.0536
ID	0.4578	0.7756	0.8048	0.0572	0.0495

Table 7: Results on answers recommendation for new questions. ‡ means the result is better than others except TEM in the same column at 5% significance level measured by Wilcoxon signed rank test and † is at 10% level.

5.3.3 Recommend Similar Questions

The third task we consider is to find similar questions for a given new question, which is defined as follows.

Task: We observe that in CQA forum, when a user asks a new question (referred as query question hereafter), the user will often get replies from other users who provide links to other similar questions. These query questions serve as an ideal question set with ground truth similar questions. We crawl $1,000$ questions to form our query question set whose similar questions exist in the training data set and serve as the ground truth. For each query question with n similar questions, we randomly select another m ($m = 1,000$) questions from our training data set to form $m + n$ candidate similar questions. Each comparing method would generate a rank list of these $m+n$ candidate similar questions according to their topic similarity to the query question. Among these candidate questions, the higher the similar questions are ranked, the better the performance of the method. The recommendation score is defined by $1 - JS(.)$, where $JS(.)$ is the JS-divergence between topic distributions of two questions. Note that CQARank uses topic distributions learnt by TEM. Hence in this task, they are equivalent.

Baselines: Any topic analysis based approach can be used as baselines because the main task here is to find those questions topical similar with the query question. We consider TSPR [35] and UQA [11] discussed in Section 5.3.1 as our baselines. Note that topics learnt by TSPR [35] are equivalent to compare with LDA [4], as TSPR uses LDA to learn topics by aggregating all posts of a user to form a "document". To measure the usefulness of tags, we consider a simple baseline, *SimTag*, which recommends questions by looking at tag similarity. We use Jaccard Index (recommendation score) to measure tag similarity, where the idea is the more tags two questions share the more similar they are.

Evaluation Criteria: We compare CQARank with baselines in terms of four criteria: precision, the average rank of the similar questions $\bar{r}$, mean reciprocal rank (MRR) and cumulative distribution of ranks (CDR). Let q be the query question and $\mathcal{Q}_s$ be the ground truth similar questions. The average rank of the similar questions is defined as: $\bar{r}(q) = \frac{1}{|\mathcal{Q}_s|}\sum_{q_s \in \mathcal{Q}_s} r(q_s)$, where $r(q_s)$ is the rank of the similar question q_s from q's $m + n$ candidate questions. The MRR and CDR are defined as follows.

$$\text{MRR} = \frac{1}{|\mathcal{Q}|}\sum_{q \in \{\mathcal{Q}\}} \frac{1}{\bar{r}(q)}$$

$$\text{CDR@}p = \frac{|\{q \in \mathcal{Q} | \bar{r}(q) \leq p\}|}{|\mathcal{Q}|}$$

where CDR@p is the percentage of users whose similar questions are ranked at least at rank p. For example, CDR@$2 = 10\%$ means 10% of query questions whose similar questions are ranked at least the second. Thus a higher value means a more successful recommendation at top p rank.

Results: We present the results in Table 8. CQARank shows a better performance than all baselines in terms of all measures. As CDR score is a single numeric value, we cannot perform significant test on it. For the rest criteria, significant test shows a very low p-value, which are all less than 1E-10. This result indicates that our method significantly outperforms the comparing methods. We would like to stress that the task of recommending similar questions is difficult as the candidate question set is very large. Another challenge is that most of time, we only observe one similar question in our data set which is one question that appears as a link in the post replying to the query question. In this case, the task is essentially to rank this question among more than 1000 candidate question set. It is therefore not surprising to observe that the precision of all the methods are not high. Yet, our method shows much better precision among all. Furthermore, all the comparing methods show a low CDR score. In CDR@50, less than 5% of query questions are observed with similar questions being ranked at least in top-50 position. Our method performs significantly better, with more than 40% of query questions. Moreover, the results show the effectiveness of considering tags to measure topics as the *SimTag* baseline has a better performance than TSPR and UQA. Our method outperforms all the baselines mainly because of two factors: (1) using tags to help learn topics; (2) jointly models topics and expertise, where the interplay between them can affect the formation of topics.

5.4 Parameter Sensitivity Analysis

We further give parameter sensitivity analysis for our proposed CQARank and Topic Expertise Model. CQARank is based on the topics and expertise model results of TEM, hence the choice of

	$\bar{r}$	MRR	P@50	P@100	CDR@50	CDR@100
CQARank(TEM)	**161**$^{\diamond}$	**0.0713**$^{\diamond}$	**0.0089**$^{\diamond}$	**0.0061**$^{\diamond}$	**0.443**	**0.611**
TSPR(LDA)	547	0.0077	0.0009	0.0009	0.049	0.093
UQA	577	0.0069	0.0009	0.0009	0.044	0.091
SimTag	386	0.1143	0.0051	0.0028	0.257	0.285

Table 8: Results on similar question recommendation for new questions. $^{\diamond}$ means the result is better than other methods at 0.0001% significance level measured by Wilcoxon signed rank test.

parameters settings such as topic number, K and expertise number, E in TEM will also influence the performance of CQARank. We perform an analysis by varying the number of expertise and topics in TEM and observing the change of CQARank performance in expert users recommendation experiments. Figure 4 and Figure 5 demonstrate the change of multiple metrics with the number of expertise and topics varied.

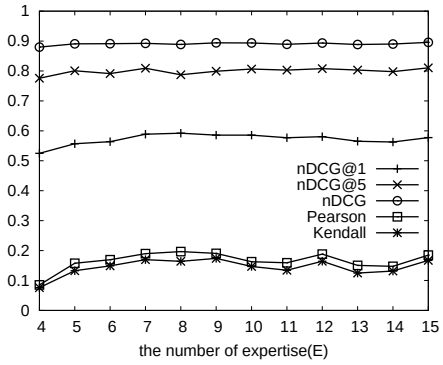


Figure 4: Performance in expert users recommendation of CQARank by varying the number of expertise (E).

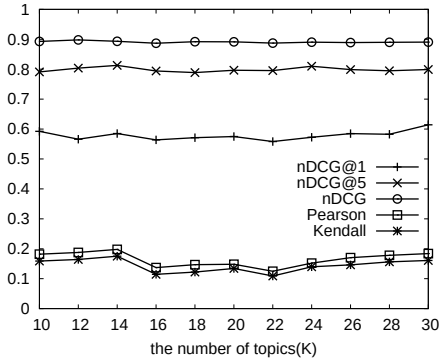


Figure 5: Performance in expert users recommendation of CQARank by varying the number of topics (K).

We can see that for nDCG metrics, the performance of CQARank is stable when the number of expertise or topics varies, which demonstrates the robustness and stability of CQARank with respect to the expertise or topic number when recommending expert users for new questions. For statistical correlation coefficients, we can see slight fluctuations with the increasing number of expertise or topics. Overall, CQARank has stable performance with varying setting of parameters.

6. RELATED WORK

Expert Identification. Current methods for expert finding in CQA are mainly based on link analysis and latent topic modeling techniques. Bouguessa et. al. [5] proposed a model based on Indegree which is the number of best answers provided by users to discover experts. Jurczyk and Agichtein [18] applied HITS[19] algorithm on the underlying graph of CQA for estimating user ranking scores. Zhang et. al. [34] proposed expertise ranking and evaluated link algorithms on a specific domain dataset. They also proposed Z-score to measure the relative expertise of a user. Although link analysis technique helps find authoritative users, a given new question on some specific topics might not interest these global experts or match their expertise and skills.

To find topic-level experts, topic-model-based methods are proposed for user topical interests analysis. Guo et. al. [11] proposed a generative model for questions and answers by exploring the category information to discover latent interests of users and recommend question answerers for new arrival questions. Liu et. al. [22] used mixture of language model and LDA for best answerer prediction. While latent topic analysis could find users interested in a given new question, these approaches fail to capture to what extent these users' expertise and skills match the questions with similar topical interest.

Furthermore, some approaches try to combine topical similarity and link analysis techniques for finding authoritative users. A typical work is [32] which proposed TwitterRank, an extension of PageRank algorithm to measure the influence of users in Twitter. Zhou et. al. [35] proposed Topic-Sensitive PageRank(TSPR) for expert finding. They also proposed a User-Topic Model, where they aggregate all posts of a user as a document. These approaches are inspired by the pioneering work of [13] which proposed original TSPR approach. Instead of computing a single global PageRank value for every page, this method computes multiple TSPR scores on topic level. Zhao et. al. [30] modeled user roles using topic models that can incorporate users contribution dynamically for generating experts and topics simultaneously. [7] modeled the user reputation in comment rating environment and proposed a latent factor model for multi-context rating prediction. Their work studied the rating information towards user comments, which is different from our problem setting focusing on community question answering that includes factual contents without opinions. Competition technique based on pair-wise approach is prosed by [21]. Techniques based on gaussian mixture models are used in similar studies such as [27; 25; 26].

Our study differs from these works in that we jointly model topics and expertise, taking in consideration both user topical interest and expertise evaluation. We also better integrate data components of CQA into our proposed model. Tagging information helps learn clean and rich topics and a Gaussian mixture hybrid can model voting information from community members which has not been well utilized in CQA for user topical expertise discovery.

Relevant Answers Retrieval. For answer retrieval, Berger et. al. [2] proposed a lexicon correlation method to build an answer finding system from FAQ whereas, Jeon et. al. [16] evaluated semantic features of answers such as author activity, number of clicks, and average length of posts to find the best answers for a given question using maximum entropy. Both methods are built on supervised techniques whereas, we propose unsupervised approach based on probabilistic generative models to find answers. Topical modeling approach for question retrieval has been proposed by [17] where the lexical gap between question-answer pairs is reduced using the topics and proved the advantage of topic models over translation-based techniques. Similar to us, Cai et. al. [6] incorporated the category information for better learning of latent topics. Apart from tags, we incorporated the topical expertise information to aid the ranking of the answers. [3] proposed a semi-supervised cou-

pled mutual reinforcement framework for simultaneously calculating the quality scores of Q&A posts, which requires relatively few labeled examples to initialize the training process. User profile information such as pictures, levels and points has been exploited by [36] for ranking answers in CQA, whereas we exploited the user expertise using the voting information which can aid in detecting more detailed user topical expertise. Our study also differs from these works in that we consider both topical expertise of authors of answers and topic similarity between questions and answers for finding answers for new questions.

Similar Question Recommendation. For similar question recommendation, Jeon et. al. [15] proposed a statistical approach that explores the semantic features to measure the question similarity. Pattern-based approach [12] depends on seed patterns with a semi-supervised approach. [20] proposed an approach based on machine translation that goes beyond the simple cosine similarity approaches and Wu et. al. [33; 28] proposed probabilistic latent semantic analysis approach that exploits both user interest and feedback, using historical data for deriving user interests. Their work showed the benefits of PSLA over the translation methods. Our study is similar to some of these works in that we explore the semantics of the questions using topic models. However, in our method, we also consider the tagging information associated with the question that aids in effective similar question retrieval task. Furthermore, to alleviate the problem of lexical gap, we jointly model topics and expertise as the interplay between them can affect the formation of topics.

7. CONCLUSION AND FUTURE WORK

In this paper, we proposed Topic Expertise Model to jointly model topics and expertise in CQA services. Based on its model results we proposed CQARank to combine textual content learning with link analysis for deriving the user expertise and interests score under various topics. Our model is generalized and applicable for various CQA tasks including expert finding, relevant answers retrieval , and similar questions recommendation. Our extensive experimental studies on Stack Overflow data sets demonstrates the effectiveness of our model when compared to other existing methods.

In the future we expect to further study the temporal aspect of users in CQA. In real world, the interests and expertise of users change with time. Capturing such temporal information could be more beneficial in recommendation tasks of CQA. Another interesting aspect is the social influence of users on CQA. The answerer profile might influence the voting behavior of users and hence impacts the recommendation methods. It is an interesting problem to analyze the correlated components in CQA for adaptive recommendation systems.

8. ACKNOWLEDGMENTS

This work was done during Liu Yang's visit to Singapore Management University. The authors would like to thank the reviewers for their valuable comments on this work.

References

[1] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *KDD '12*, pages 850–858, 2012.

[2] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *SIGIR '00*, pages 192–199, 2000.

[3] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *WWW '09*, pages 51–60, 2009.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] M. Bouguessa, B. Dumoulin, and S. Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *KDD '08*, pages 866–874, 2008.

[6] L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the latent topics for question retrieval in community qa. In *IJCNLP '11*, pages 273–281, 2011.

[7] B.-C. Chen, J. Guo, B. Tseng, and J. Yang. User reputation in a comment rating environment. In *KDD '11*, pages 159–167, 2011.

[8] G. Dror, Y. Koren, Y. Maarek, and I. Szpektor. I want to answer; who has a question?: Yahoo! answers recommender system. In *KDD '11*, pages 1109–1117, 2011.

[9] T. Elguebaly and N. Bouguila. Bayesian learning of finite generalized gaussian mixture models on images. *Signal Process.*, 91(4):801–820, Apr. 2011.

[10] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(suppl. 1):5228–5235, 2004.

[11] J. Guo, S. Xu, S. Bao, and Y. Yu. Tapping on the potential of qa community by recommending answer providers. In *CIKM '08*, pages 921–930, 2008.

[12] T. Hao and E. Agichtein. Finding similar questions in collaborative question answering archives: toward bootstrapping-based equivalent pattern learning. *Inf. Retr.*, 15(3-4):332–353, 2012.

[13] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW '02*, pages 517–526, 2002.

[14] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[15] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *CIKM '05*, pages 84–90, 2005.

[16] J. Jeon, W. B. Croft, J. H. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. In *SIGIR '06*, pages 228–235, 2006.

[17] Z. Ji, F. Xu, B. Wang, and B. He. Question-answer topic model for question retrieval in community question answering. In *CIKM '12*, pages 2471–2474, 2012.

[18] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. In *CIKM '07*, pages 919–922, 2007.

[19] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.

[20] S. Li and S. Manandhar. Improving question recommendation by exploiting information need. In *HLT '11*, pages 1425–1434, 2011.

[21] J. Liu, Y.-I. Song, and C.-Y. Lin. Competition-based user expertise score estimation. In *SIGIR '11*, pages 425–434, 2011.

[22] X. Liu, W. B. Croft, and M. Koll. Finding experts in community-based question-answering services. In *CIKM '05*, pages 315–316, 2005.

[23] K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, 2007.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, November 1999.

[25] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM '11*, pages 45–54, 2011.

[26] A. Pal, F. M. Harper, and J. A. Konstan. Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Trans. Inf. Syst.*, 30(2):10:1–10:28, May 2012.

[27] A. Pal and J. A. Konstan. Expert identification in community question answering: exploring question selection bias. In *CIKM '10*, pages 1505–1508, 2010.

[28] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *WWW '09*, pages 1229–1230, 2009.

[29] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP '09*, pages 248–256, 2009.

[30] Z. Tong, B. Naiwen, L. Chunping, and L. Mengya. Topic-level expert modeling in community question answering. In *SDM '13*, 2013.

[31] K. Wang, Z.-Y. Ming, X. Hu, and T.-S. Chua. Segmentation of multi-sentence questions: towards effective question retrieval in cqa services. In *SIGIR '10*, pages 387–394, 2010.

[32] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM '10*, pages 261–270, 2010.

[33] H. Wu, Y. Wang, and X. Cheng. Incremental probabilistic latent semantic analysis for automatic question recommendation. In *RecSys '08*, pages 99–106, 2008.

[34] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *WWW '07*, pages 221–230, 2007.

[35] G. Zhou, S. Lai, K. Liu, and J. Zhao. Topic-sensitive probabilistic model for expert finding in question answer communities. In *CIKM '12*, pages 1662–1666, 2012.

[36] Z.-M. Zhou, M. Lan, Z.-Y. Niu, and Y. Lu. Exploiting user profile information for answer ranking in cqa. In *WWW '12 Companion*, pages 767–774, 2012.