A Tri-Role Topic Model for Domain-Specific Question Answering

Zongyang Ma **Aixin Sun** **Quan Yuan** **Gao Cong**
School of Computer Engineering, Nanyang Technological University, Singapore 639798
{zma4, qyuan1}@e.ntu.edu.sg {axsun, gaocong}@ntu.edu.sg

Abstract

Stack Overflow and MedHelp are examples of domain-specific community-based question answering (CQA) systems. Different from CQA systems for general topics (*e.g.,* Yahoo! Answers, Baidu Knows), questions and answers in domain-specific CQA systems are mostly in the same topical domain, enabling more comprehensive interaction between users on fine-grained topics. In such systems, users are more likely to ask questions on unfamiliar topics and to answer questions matching their expertise. Users can also vote answers based on their judgements. In this paper, we propose a Tri-Role Topic Model (TRTM) to model the tri-roles of users (*i.e.,* as askers, answerers, and voters, respectively) and the activities of each role including composing question, selecting question to answer, contributing and voting answers. The proposed model can be used to enhance CQA systems from many perspectives. As a case study, we conducted experiments on ranking answers for questions on Stack Overflow, a CQA system for professional and enthusiast programmers. Experimental results show that TRTM is effective in facilitating users getting ideal rankings of answers, particularly for new and less popular questions. Evaluated on nDCG, TRTM outperforms state-of-the-art methods.

Introduction

In community-based question answering (CQA) systems for general topics (*e.g.,* Yahoo! Answers, Baidu Knows), users may ask questions of any topic, *e.g.,* "*What is Paris famous for?*". To answer this question, no much professional knowledge is needed. In domain-specific CQA systems (*e.g.,* Stack Overflow, MedHelp), professional knowledge is required to answer questions like "*How does ruby on rails handle requests?*" from Stack Overflow, and "*How do I know when skipped heart beats are dangerous?*" from MedHelp. While a lot of studies have been carried out on CQA for general topics (Xu, Ji, and Wang 2012; Li, Shen, and Grant 2012; Dror et al. 2011). There are limited studies on domain-specific CQA systems.

Domain-specific CQA systems have recently attracted a lot of users and accumulated a large number of domain-specific questions and answers. For example, Stack Overflow, for professional and enthusiast programmers, has over 2 million registered users and more than 7 million questions as at Apr 2014, according to Wikipedia. Users who have questions which require professional knowledge are more likely to source for help in domain-specific CQA systems, because these systems enable comprehensive interaction between users on fine-grained domain-specific topics. However, there is a lack of study on the formal modeling of users' activities in a unified framework. In this study, we consider and model users' activities with their tri-roles (*i.e., asker*, *answerer*, and *voter*) in a probabilistic framework at topic level. A topic-level modeling for users' activities benefits many applications in CQA such as ranking answers for questions, and expert finding.

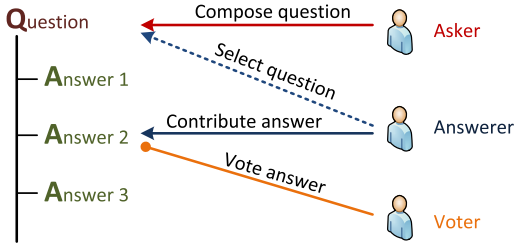

Figure 1: The tri-roles of a user and the activities

Illustrated in Figure 1, in a typical domain-specific CQA system like Stack Overflow, users may perform activities as three roles.

- **Asker**. To ask a question of unfamiliar topics, a user composes the question and waits for answers to this question.
- **Answerer**. If a user believes that she has the knowledge to answer this question, she contributes an answer. Note that, there is an *implicit* question selection activity where an answerer performs a self-assessment whether she has the knowledge to answer this question. The question selection activity is indicated by the dotted line in Figure 1.
- **Voter**. In CQA systems, users are often allowed to vote for the answers to a question, based on their judgements.

Note that, a user may perform the three roles (*i.e.,* asker, answerer, or voter) simultaneously across different questions in a CQA system.

In this paper, we propose a probabilistic Tri-Role Topic Modeling (TRTM) which makes use of the three roles of users for modeling users' activities and for mining fine-grained topics in domain-specific CQA systems. Our model is an extension of Probabilistic Latent Semantic Analysis (PLSA), which assumes that each document has a mixture of topics and each word of the document is sampled from its related topic (Hofmann 1999). Here, a document can be a question or an answer.

As aforementioned, a user composes questions in her unfamiliar topics, and contributes answers if she believes that she has the right knowledge. Users also vote positively for answers that well address the questions. We therefore argue that the topic distributions of the asker role and the answerer role of the same user could be very different (*e.g.,* unfamiliar topics vs familiar topics). Moreover, if an answer receives a large number of positive votes, then the answer is believed to be of similar topic distribution with the question. TRTM therefore assigns each user an asker role and an answerer role; each role has its own topic distribution. TRTM also implicitly models the voter role of users, which has no explicit topic distributions but contributes to constraining topical matching between questions and answerers. As the result, TRTM generates not only topic distributions of questions and answers, but also topic distributions of askers and answerers. TRTM makes three assumptions: (1) An asker and all the questions composed by her share similar topic distributions; (2) An answerer and all the answers contributed by her share similar topic distributions; and (3) An answerer's topic distribution is more similar to that of the questions answered by her, if her answers to these questions receive many positive votes.

To evaluate the effectiveness of our model, we apply TRTM to the application of ranking answers for questions. A popular question may receive many answers within a short period. The task of ranking answers is to rank the best answers to top positions before waiting for users' votes as the latter might take a long time. TRTM outperforms two state-of-the-art baselines on real data collected from Stack Overflow on this task. TRTM can also be utilized in many applications such as expert finding and similar question searching.

Related Work

We briefly overview the studies on Stack Overflow, and then survey topic models on CQA systems.

Stack Overflow. Anderson *et al.* (2012) found that in Stack Overflow, expert users are likely to answer questions more quickly, and a higher activity level of a question benefits all answerers of this question to increase their reputation level. Based on some extracted features from Stack Overflow, they attempted to predict the long-term value of a question and whether a question has been sufficiently answered. Their results show that votes indicate a user's expertise level on a specific topic. This is consistent with the modeling of votes in our proposed TRTM. Subsequently, Anderson *et al.* (2013) observed that badge mechanism in Stack Overflow steers users to increase their participation in answering questions. Question deletion in Stack Overflow was studied in (Correa and Sureka 2014), where 47 features were used to predict whether a question will be deleted. The quality of question content is found to be the main factor.

Dalip *et al.* (2013) proposed to rank answers of a question in Stack Overflow using Learning to Rank (L2R), a supervised approach. The L2R model is learned from the feature vector representations of question-answer pairs. Each pair is represented by features in 8 groups (*e.g.,* user features, structure features, style features). Answers of new coming questions are then predicted by the trained L2R model. Note that, L2R is a supervised approach. In our work, we focus on a generative probabilistic model, which is unsupervised in nature, to model the three roles of users and their activities.

Most germane to our work is the Topic Expertise Model (TEM) proposed in (Yang et al. 2013). TEM is a LDA-based model to jointly model topics and expertise of users. Gaussian mixture hybrid is used to model votes. TRTM is significantly different from TEM because TRTM considers and models the three roles of users. Our experimental results confirm that modeling three roles of users benefits the mining fine-grained topics of users. TEM was evaluated on three applications: expert finding, similar question searching, and ranking answers for questions. Nevertheless, the problem definition of ranking answers is different from ours. In (Yang et al. 2013), only answers from answerers that appear in training data can be ranked. In our problem definition, answers from new answerers can also be ranked.

Topic models on CQA systems. A body of literature exists on topic modeling in CQA context. Existing work can be classified into PLSA-based models (Xu, Ji, and Wang 2012; Wu, Wang, and Cheng 2008) and LDA-based models (Ji et al. 2012; Zhou et al. 2012; Guo et al. 2008).

Xu *et al.* (2012) proposed Dual-Role Model (DRM) to recommend questions to matching users in Yahoo! Answers. DRM is a PLSA-based model and it models the asker role and answerer role of users, but not the voter role. Through DRM, the authors showed clear distinctions between the two roles of users. Compared to TRTM, other than ignoring the voter role, DRM cannot directly generate topic distributions of askers and answerers although the two roles are considered in the modeling. An Incremental PLSA is proposed in (Wu, Wang, and Cheng 2008) for question recommendation in Wenda. The model considers users' long-term interests and short-term interests for question matching.

Ji *et al.* (2012) presented a LDA-based model for question retrieval. The model outperforms translation models on the task by using topics discovered from Yahoo! Answers. A user-topic model treating all documents of a user (*i.e.,* questions and answers) as an aggregated document was proposed in (Zhou et al. 2012). The user-topic model was applied for finding expert users in CQA and it outperforms PageRank (Page et al. 1999). Guo *et al.* (2008) extended LDA using category information in Yahoo! Answers to return expert lists for new questions. Their generative model is able to discover latent topics in the content of questions and answers and the latent interests of users. The experimental results evidence the capability of topic models for recommendation problems in CQA systems.

Table 1: Notations

Symbol	Description
U	Collection of askers $u \in U$
V	Collection of answerers $v \in V$
Q	Collection of questions $q \in Q$
A	Collection of answers $a \in A$
Z	Collection of topics $z \in Z$
W	Word vocabulary of questions
E	Word vocabulary of answers
Q_u	Set of questions composed by u
Q_v	Set of questions answered by v
A_v	Set of answers contributed by v
V_q	Set of answerers to question q
s_q^a	Voting score of answer a to question q
$n(q,w)$	Number of occurrences of $w \in W$ in q
$n(a,e)$	Number of occurrences of $e \in E$ in a

Tri-Role Topic Model

We start with the notations used in our model, summarized in Table 1. We then present the Tri-Role Topic Model (TRTM) and its inference algorithm.

Notations

Let Q and A be the set of questions and the set of answers respectively. W is the word vocabulary of questions and E is the word vocabulary of answers. Let U be the set of askers (*i.e.,* users who ever asked questions), and let V denote the set of answerers (*i.e.,* users who ever answered questions). Note that, a user can be an asker for one question and be an answerer of another question, *i.e.,* $U \cap V \neq \Phi$. We further use Q_u to denote the set of questions composed by asker u, and use Q_v to denote the set of questions answered by answerer v. The set of answers contributed by v is denoted by A_v.

An answer a, contributed by answerer v to question q, may receive zero or more votes. As a vote can be either positive or negative, the aggregated number of votes to an answer may be negative. For easy processing, we compute a *voting score* for answer a to question q as $s_q^a = x - x_{min} + 0.5$, where x is raw aggregated vote for this answer and x_{min} is the lowest vote of an answer in our data collection, and 0.5 is a constant to ensure $s_q^a > 0$. In other word, s_q^a is the aggregated votes of an answer shifted to positive region.

Model Description

In TRTM, each user has three roles: an asker, an answerer, and a voter. As discussed earlier, different from a question or an answer, a vote is not associated with any textual content. Moreover, votes become meaningful only when the number of votes is large. We therefore do not model the voter role explicitly in TRTM. Instead, voting is used to constrain the topic distributions in our model.

TRTM models four types of topic distributions. Let z denote topic. The four types of topic distributions are: (i) $p(z|u)$, topic distribution of asker u, (ii) $p(z|v)$, topic distribution of answerer v, (iii) $p(z|q)$, topic distribution of question q, and (iv) $p(z|a)$, topic distribution of answer a. Note that, TRTM assumes that each question or answer has multiple topics and each word is sampled from its corresponding topic with probability $p(w|z)$ or $p(e|z)$. Regarding topic distributions, we make the following three assumptions (**A1**, **A2**, and **A3**) in TRTM. Note that **A1** and **A3** are also adopted in (Xu, Ji, and Wang 2012).

- **A1**: An asker u and all the questions composed by her Q_u share similar topic distributions.
- **A2**: An answerer v and all the questions answered by her Q_v, share similar topic distributions. The degree of similarity in the topic distributions between v and Q_v is reflected by the voting scores of her answers. If an answer to question q by v receives a large voting score, then most users believe that this answer well addresses the question; hence answerer v has the expertise in answering this question q. v and q therefore share more similar topic distributions. Here, we use the *voter* role of users (*i.e.,* those who are not answerers or the asker) to constrain the topic distributions between the answerer and her questions. In simple words, an answerer's topic distribution is more similar to that of the questions answered by her, if her answers to these questions receive more positive votes.
- **A3**: An answerer v and all the answers contributed by her A_v share similar topic distributions.

In TRTM, we adopt the *exponential KL-divergence (eKL)* function to model the relationship between two topic distributions. Proposed in (Kim, Park, and Shim 2013), the eKL function is the combination of exponential probability densities and KullbackLeibler divergence. For two k-dimensional probability distributions μ and θ and a given scalar λ, $eKL(\theta, \lambda, \mu)$ is defined as:

$$eKL(\theta, \lambda, \mu) = \lambda e^{-\lambda KL(\mu||\theta)}$$

where $KL(\mu||\theta)$ is $\sum_k \mu_k \log(\mu_k/\theta_k)$. The properties of the $eKL(\theta, \lambda, \mu)$ function include: (i) with a fixed λ, the eKL value increases with the degree of similarity between μ and θ, and (ii) with a larger λ, the exponential probability densities decrease faster when increasing the value of $KL(\mu||\theta)$.

The generative process of TRTM is divided into three subprocedures, namely (i) composing question, (ii) selecting question to answer, and (iii) contributing answer.

Composing question. An asker u composes a question $q \in Q_u$ with probability $eKL(u, \alpha, q)$, where α is a scalar. When topic distributions of u and q are more similar, u is more likely to compose question q because of interest matching (Assumption **A1**). Next, for each word w in q, a topic z is sampled with probability $p(z|q)$, and then w is generated based on $p(w|z)$.

Selecting question to answer. The probability of an answerer v choosing to answer a question $q \in Q_v$ is modeled as $eKL(v, \beta \cdot s_q^a, q)$, where s_q^a is the voting score and β is a scalar. Recall that in Assumption **A2**, the degree of similarity in the topic distributions between v and Q_v is reflected by the voting scores of her answers. The larger the voting score s_q^a, the sharper the curve of $eKL(v, \beta \cdot s_q^a, q)$, which means that the eKL assigns a higher probability when the distance between v and q gets smaller.

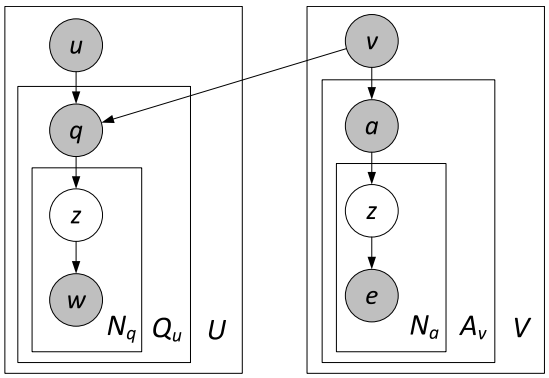

Figure 2: Tri-Role Topic Model. Note that the voter role of user is modeled implicitly, and the plate model does not show s_q^a.

Contributing answer. An answerer v contributes an answer $a \in A_v$ with probability $eKL(v, \tau, a)$, where τ is a scalar. Based on Assumption **A3**, v prefers to contribute a if the topical similarity between v and a is high. Next, for each word e in a, a topic z is sampled with probability $p(z|a)$, and then e is generated with $p(e|z)$.

The graphical representation of the TRTM model is shown in Figure 2 and the generative process is summarized as follows:

- For each question $q \in Q_u$
  - For each answer $a \in A_v$ of question q
    - Asker u composes question q with probability $eKL(p(z|q), \alpha, p(z|u))$
    - For each word w in q
      - Draw a topic z from $p(z|q)$
      - Draw a word w from $p(w|z)$
    - Answerer v selects to answer question q with probability $eKL(p(z|q), \beta \cdot s_q^a, p(z|v))$
    - Answerer v contributes answer a with probability $eKL(p(z|a), \tau, p(z|v))$
    - For each word e in a
      - Draw a topic z from $p(z|a)$
      - Draw a word e from $p(e|z)$

The Inference Algorithm of TRTM

Given question set Q composed by U and answer set A by answerers from V, we obtain the likelihood of the data in Equation 1.

$$
\begin{aligned}
L = & \prod_{u \in U} \prod_{q \in Q_u} eKL\Big(p(z|q), \alpha, p(z|u)\Big) \\
& \prod_{u \in U} \prod_{q \in Q_u} \prod_{w \in W} \left[\sum_z p(w|z)p(z|q)\right]^{n(q,w)} \\
& \prod_{v \in V} \prod_{q \in Q_v} eKL\Big(p(z|q), \beta \cdot s_q^a, p(z|v)\Big) \\
& \prod_{v \in V} \prod_{a \in A_v} eKL\Big(p(z|a), \tau, p(z|v)\Big) \\
& \prod_{v \in V} \prod_{a \in A_v} \prod_{e \in E} \left[\sum_z p(e|z)p(z|a)\right]^{n(a,e)}
\end{aligned} \tag{1}
$$

The exact inference of Equation 1 is intractable. We propose an Expectation-Maximization (EM) algorithm to appropriately infer TRTM. The EM algorithm has two steps: E-step and M-step. The E-step calculates the expectation of the hidden variables *i.e.,* $p(z|q, w)$ and $p(z|a, e)$ in TRTM.

- **E-step:**

$$
\begin{aligned}
p^{k+1}(z|q, w) &= \frac{p^k(w|z)p^k(z|q)}{\sum_{z' \in Z} p^k(w|z')p^k(z'|q)} \\
p^{k+1}(z|a, e) &= \frac{p^k(e|z)p^k(z|a)}{\sum_{z' \in Z} p^k(e|z')p^k(z'|a)}
\end{aligned}
$$

The M-step maximizes the log-likelihood (see Equation 1). The following probabilities are calculated: $p(w|z)$, $p(e|z)$, $p(z|u)$, $p(z|q)$, $p(z|a)$, and $p(z|v)$.

- **M-step:**

$$
\begin{aligned}
p^{k+1}(w|z) &= \frac{\sum_{q \in Q} n(q, w)p^{k+1}(z|q, w)}{\sum_{w' \in W} \sum_{q \in Q} n(q, w')p^{k+1}(z|q, w')} \\
p^{k+1}(e|z) &= \frac{\sum_{a \in A} n(a, e)p^{k+1}(z|a, e)}{\sum_{e' \in E} \sum_{a \in A} n(a, e')p^{k+1}(z|a, e')} \\
p^{k+1}(z|u) &= \frac{[\prod_{q \in Q_u} p^{k+1}(z|q)]^{1/|Q_u|}}{\sum_{z' \in Z} [\prod_{q \in Q_u} p^{k+1}(z'|q)]^{1/|Q_u|}}
\end{aligned}
$$

The calculation of the probabilities $p(z|q)$, $p(z|a)$, and $p(z|v)$ are shown in Table 2 for presentation clarity. We iteratively compute probabilities of **E-step** and **M-step** until achieving convergent log-likelihood (see Equation 1). k represents the kth iteration of EM algorithm. Note that, in **M-step**, we first calculate $p(z|q)$ and $p(z|a)$, then calculate $p(z|u)$ and $p(z|v)$.

Experiment

We evaluate the proposed TRTM model on Stack Overflow data[1]. Although the model can be used to enable multiple applications, due to page limit, we report one case study for the application of ranking answers for questions.

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

Table 2: **M-step** (continued) for computing $p(z|v)$, $p(z|q)$, and $p(z|a)$

$$p^{k+1}(z|v) = \frac{\left[\prod_{q\in Q_v} p^{k+1}(z|q)^{\beta\cdot s_q^a} \prod_{a\in A_v} p^{k+1}(z|a)^{\tau}\right]^{1/(\sum_{q\in Q_v}\beta\cdot s_q^a+\tau|A_v|)}}{\sum_{z'\in Z}\left[\prod_{q\in Q_v} p^{k+1}(z'|q)^{\beta\cdot s_q^a} \prod_{a\in A_v} p^{k+1}(z'|a)^{\tau}\right]^{1/(\sum_{q\in Q_v}\beta\cdot s_q^a+\tau|A_v|)}}$$

$$p^{k+1}(z|q) = \frac{\sum_{w\in W} n(q,w)p^{k+1}(z|q,w) + \sum_{u\in U_q}\alpha p^k(z|u) + \sum_{v\in V_q}\beta\cdot s_q^a\cdot p^k(z|v)}{\sum_{z'\in Z}\left\{\sum_{w\in W} n(q,w)p^{k+1}(z'|q,w) + \sum_{u\in U_q}\alpha p^k(z'|u) + \sum_{v\in V_q}\beta\cdot s_q^a\cdot p^k(z'|v)\right\}}$$

$$p^{k+1}(z|a) = \frac{\sum_{e\in E} n(a,e)p^{k+1}(z|a,e) + \sum_{v\in V_a}\tau p^k(z|v)}{\sum_{z'\in Z}\left\{\sum_{e\in E} n(a,e)p^{k+1}(z'|a,e) + \sum_{v\in V_a}\tau p^k(z'|v)\right\}}$$

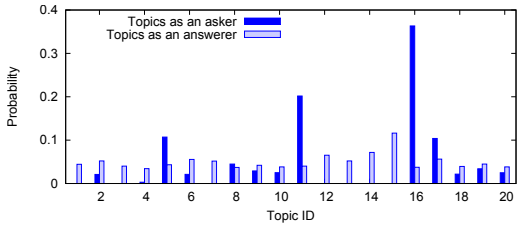

Figure 3: Topic distributions of asker role and answerer role from the same example user

Data Set. Questions and answers from Stack Overflow posted between Jan 01, 2011 and Mar 31, 2011 are used as training data; questions and answers published from Apr 01, 2011 to Sep 06, 2013 are used as test data.

We preprocess the training data by removing questions and answers from inactive users. More specifically, a user is inactive if the total number of questions and answers posted by her is smaller than 80, as defined in (Yang et al. 2013). After preprocessing, the training data contains 16,141 questions from 868 askers and 180,394 answers from 1,184 answerers. Note that a user could play a single role (*i.e.,* an asker or an answerer). The vocabulary size for questions is 21,760 and that for answers is 85,889. The raw aggregated vote is in the range of -10 to 359.

For test data, questions with fewer than 5 answers are removed, because answer ranking is more meaningful if a question has a large number of answers. As the result, the test data contains 20,834 questions and 150,320 answers. Note that, the askers and answerers in the test data may not appear in the training data.

We experimentally set the hyperparameters of TRTM: $\alpha = 100$, $\beta = 100$, $\tau = 100$. We evaluated different number of topics $|Z|$ = 10, 20, and 40.

Topic Discovery

Discovered by TRTM, we randomly select 5 topics of questions and 5 topics of answers as examples, shown in Tables 3(a) and 3(b) respectively. The top-10 words based on $p(w|z)$ and $p(e|z)$ respectively are listed for each topic. Observe that TRTM captures some major topics of questions and answers in Stack Overflow. Questions related to Java programming (Topic 8) and Web development (Topic 14) are frequently asked. Process and thread (Topic 13) and Server client programming (Topic 20) are prevalent in answers. Note that, the vocabulary sets of the questions and of the answers are significantly different. Naturally, the number of words in all answers to a question is much larger than the words in the question itself. More importantly, there are more technical terminologies in answers than that in questions.

Table 3: Example topics by TRTM from Stack Overflow

(a) Example question topics with topic ID and words

ID	Top-10 words with highest generative probability
6	file error window install reference build directory js header include
8	app java process compile org template apache module map default
12	image project script photo folder null get collection assembly generate
14	php div tag run html load link content element use
18	data create database select custom ve use answer please size

(b) Example answer topics with topic ID and words

ID	Top-10 words with highest generative probability
1	change memory compile program language address python stack git branch
5	string page example html look jquery instead url javascript content
8	application java request access cache compile load log map api
13	thread bit process result field template loop message format single
20	name server service net project client source standard connection header

We also show topic distributions of a randomly selected user for her asker and answerer roles in Figure 3. Observe that the topic distributions of the two roles are significantly different. For instance, topics 11 and 16 have large $p(z|u)$ in her question topic distribution. For the answer topic distribution, topic 15 has the largest probability $p(z|v)$. This indicates that this user is less familiar with questions of topics 11 and 16 but she has the expertise in providing answers for topic 15. TRTM is capable of distinguishing question and

answer topic distributions for users as askers and answerers.

Ranking Answers for Questions

A popular question could receive many answers within a very short period. However, given the short time period, there might be lack of enough votes to help the asker to select the high quality answers, because the answers may be from a domain the asker is unfamiliar with. Timely ranking answers for questions benefits askers in quickly getting high-quality answers.

Problem definition. Given a question q and its answer set A_q, the task of *ranking answers* is to rank answers $a \in A_q$ such that the top-ranked answers best address q. In this sense, we assume that the best answers for a question are the ones sharing most similar topic distributions with the question. The answers are then ranked by topical similarities to the question, and the topics of a's and q are learned using topic models, TRTM or other baseline models.

The *topical similarity* (TS) between a question q and an answer a is evaluated using Jensen-Shannon divergence,

$$TS(q, a) = JSD(\theta_q, \theta_a)$$

where θ_q and θ_a represent the topic distributions of question q and answer a respectively.

$$\theta_q \approx p(\boldsymbol{w}_q|z) = \sum_{w \in \boldsymbol{w}_q} p(w|z)$$

$$\theta_a \approx p(\boldsymbol{e}_a|z) = \sum_{e \in \boldsymbol{e}_q} p(e|z)$$

In above equations, $\boldsymbol{w}_q$ and $\boldsymbol{e}_q$ are word vectors for question q and answer a respectively.

Baseline methods. Latent Dirichlet Allocation (**LDA**) is a standard technique for topic analysis in document collections (Blei, Ng, and Jordan 2003). Here, a virtual document is created for each user by aggregating all her questions and answers, and then LDA is employed to learn the hidden topics (*i.e.,* $p(w|z)$). Topic Expertise Model (**TEM**) is a very recent model proposed in (Yang et al. 2013). Considered as a state-of-the-art baseline, TEM jointly models user topical interests and expertise in a probabilistic model. TEM has been evaluated on Stack Overflow data and has been applied for the task of answer ranking but with a different problem setting in (Yang et al. 2013)[2]. For both LDA and TEM, the topical similarity is computed in a similar way as in TRTM.

Evaluation measure. We use normalized discounted cumulative gain (nDCG) to evaluate the list of ranked answers, following (Yang et al. 2013). Here, the ground-truth ranking of the answers to a question is the ranking by the number of aggregated votes of the answers. The number of aggregated votes is also used to define the degree of relevance of each item in a rank, required by the nDCG measure. $nDCG@M$ for the top-M ranked answers of test question q computed as follows:

$$nDCG(q, M) = \frac{1}{IDCG(q, M)} \sum_{i=1}^{M} \frac{2^{rv_{q,i}} - 1}{\log_2(i+1)}$$

[2]In TEM, only the answers from the answerers that appear in training data are ranked. In our proposed solution, we utilize the words in an answer (*i.e.,* $p(z|w)$) where the answerer may not appear in the training data.

Table 4: nDCG of the three models; the best result for each topic number setting ($|Z|$=10, 20, 40) is in boldface.

$\|Z\|$	Model	$nDCG@1$	$nDCG@5$	$nDCG@10$	$nDCG$
10	TRTM	**0.3273**	**0.6448**	**0.6759**	**0.6762**
	TEM	0.3005	0.6281	0.6607	0.6611
	LDA	0.3026	0.6296	0.6618	0.6622
20	TRTM	**0.3405**	**0.6518**	**0.6824**	**0.6828**
	TEM	0.3052	0.6303	0.6630	0.6633
	LDA	0.3093	0.6331	0.6651	0.6654
40	TRTM	**0.3380**	**0.6506**	**0.6806**	**0.6810**
	TEM	0.3106	0.6333	0.6657	0.6660
	LDA	0.3195	0.6411	0.6719	0.6722

where $rv_{q,i}$ is the number of aggregated votes received by the answer ranked at the i-th position; $IDCG(M, q)$ is the normalization factor for the discounted cumulative gain of the ideal ranking of the top-M answers for question q. Then $nDCG@M$ is the average of $nDCG(q, M)$ over all questions in the test data.

Experimental results The $nDCG@M$'s of the three models with $|Z| = 10$, 20, and 40 topics are reported in Table 4 where M=1, 5, 10, and all answers. The following three observations are made from the results.

- TRTM performs better than both baseline methods TEM and LDA, for all different settings on number of topics and on all M settings. Particularly, on $nDCG@1$, TRTM outperforms TEM by 11.6% and LDA by 10.1% respectively. The results evidence the effectiveness of our proposed model.
- All models with 20 topics yield best results, which suggests that 20 is a more appropriate number of topics on this dataset. On the other hand, all the three models are relatively not very sensitive to topic number setting.
- LDA slightly outperforms TEM. One possible reason is that TEM assumes each question (resp. each answer) has only one unique topic, which is not appropriate in modeling Stack Overflow data, where some questions and answers are fairly long and may cover multiple topics.

Conclusion and Future Work

In this paper, we propose a Tri-Role Topic Model to model the tri-roles of users (*i.e.,* askers, answerers, and voters) in CQA systems and the activities of each role (*i.e.,* composing question, selecting question to answer, contributing, and voting answers). Our model is capable of mining four topic distributions of asker, answerer, question, and answer, respectively. We demonstrate the effectiveness of our model in discovering topics from Stack Overflow and also in addressing the problem of ranking answers for questions on the same dataset.

As a part of future work, we expect to model the rich temporal patterns of users in CQA systems. We observe that different users prefer to answer questions at different time point of a day and different day of a week. Incorporating temporal patterns of users has great potential in modeling users' activities more accurately.

References

Anderson, A.; Huttenlocher, D.; Kleinberg, J.; and Leskovec, J. 2012. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *KDD*, 850–858. ACM.

Anderson, A.; Huttenlocher, D.; Kleinberg, J.; and Leskovec, J. 2013. Steering user behavior with badges. In *WWW*, 95–106.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

Correa, D., and Sureka, A. 2014. Chaff from the wheat: Characterization and modeling of deleted questions on stack overflow. In *WWW*, 631–642.

Dalip, D. H.; Gonçalves, M. A.; Cristo, M.; and Calado, P. 2013. Exploiting user feedback to learn to rank answers in Q&A forums: A case study with stack overflow. In *SIGIR*, 543–552. ACM.

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

Guo, J.; Xu, S.; Bao, S.; and Yu, Y. 2008. Tapping on the potential of Q&A community by recommending answer providers. In *CIKM*, 921–930. ACM.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *SIGIR*, 50–57. ACM.

Ji, Z.; Xu, F.; Wang, B.; and He, B. 2012. Question-answer topic model for question retrieval in community question answering. In *CIKM*, 2471–2474. ACM.

Kim, Y.; Park, Y.; and Shim, K. 2013. DIGTOBI: A recommendation system for digg articles using probabilistic modeling. In *WWW*, 691–702.

Li, Z.; Shen, H.; and Grant, J. E. 2012. Collective intelligence in the online social network of Yahoo! answers and its implications. In *CIKM*, 455–464. ACM.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Wu, H.; Wang, Y.; and Cheng, X. 2008. Incremental probabilistic latent semantic analysis for automatic question recommendation. In *RecSys*, 99–106. ACM.

Xu, F.; Ji, Z.; and Wang, B. 2012. Dual role model for question recommendation in community question answering. In *SIGIR*, 771–780. ACM.

Yang, L.; Qiu, M.; Gottipati, S.; Zhu, F.; Jiang, J.; Sun, H.; and Chen, Z. 2013. CQArank: Jointly model topics and expertise in community question answering. In *CIKM*, 99–108. ACM.

Zhou, G.; Lai, S.; Liu, K.; and Zhao, J. 2012. Topic-sensitive probabilistic model for expert finding in question answer communities. In *CIKM*, 1662–1666. ACM.