



HAL
open science

Joint model of topics, expertises, activities and trends for question answering Web applications

Zide Meng, Fabien Gandon, Catherine Faron Zucker

► To cite this version:

Zide Meng, Fabien Gandon, Catherine Faron Zucker. Joint model of topics, expertises, activities and trends for question answering Web applications. 2016 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Oct 2016, USA, Oct 2016, Omaha, United States. hal-01293968

HAL Id: hal-01293968

<https://inria.hal.science/hal-01293968>

Submitted on 25 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Joint model of topics, expertises, activities and trends for question answering Web applications

Zide Meng
INRIA Sophia Antipolis
Méditerranée
France
zide.meng@inria.fr

Fabien Gandon
INRIA Sophia Antipolis
Méditerranée
France
fabien.gandon@inria.fr

Catherine Faron Zucker
University Nice Sophia
Antipolis, CNRS, I3S
France
faron@unice.fr

ABSTRACT

Users in question-answer sites generate huge amounts of high quality and highly reusable information. This information can be categorized by topics but since users' interests change with time, it's important to uncover the temporal patterns and trends in their activity to detect their current expertise. These temporal variations remained unexplored in question-answer sites while detecting them enables us to improve tasks such as: question routing, expert recommending and community life-cycle management. In this paper, we proposed a generative model of such a community and its dynamics, and we performed experiments with real-world data to confirm the effectiveness of our joint model, studying the users' behaviors and topics dynamics on a dataset extracted from the popular question-answer site StackOverflow.

Keywords

User behavior analysis, Topic evolution, Question Answering

1. INTRODUCTION

Community Question Answering(CQA) applications provide a social media where users can ask expert for help. A large number of people are very active on these sites and keep contributing answers. Most of them are more likely to answer questions about topics in which they are interested and specialized. In addition, since questions and answers can be viewed and searched afterward, people with similar questions can also directly find solutions provided in the past. Gradually, CQA sites have become huge repositories which provide highly reusable and highly valuable knowledge. Yet their size, heterogeneity and continuous changes raise many challenges to ensure a sustainable use and maintenance: *Can we automatically identify the topics of a CQA? Which topics are the most interesting to a given user? Which users are the most active ones on a given topic? Which topics a user has the most expertise in? How do the topics change with time? How do a user's interests change with time?* If we could extract this information from the dataset of a CQA, we could support many of the tasks of the life-cycle of the community ; for example, recommending a question to a user who is active in the corresponding topic and has the needed expertise.

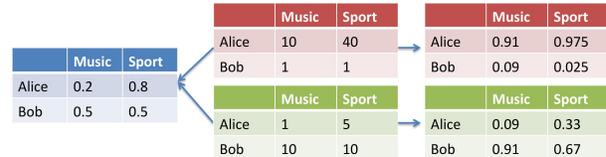


Figure 1: Different ways to estimate probabilities with results of Gibbs Sampling

In previous works, [17] only considered modeling the expertise, and [9] only considered modeling the topic trends at a group level. Compared with them, We jointly model both expertise and dynamics and we model topic trends also at the level of the users. This provides more insight on the changes of interest of each user.

In addition, the LDA based models have a common problem. Let us explain this in detail. If we use a three-layer LDA model (user-topic-word), this generates two kinds of distributions (user-topic distribution and topic-word distribution) which describe to what extent a user is interested in different topics and to what extent a keyword or a tag is related to a topic.

However, As shown in figure 1 the same user-topic distribution could be generated by different training data (Assume that the hidden variable topic is generated by Gibbs Sampling[6]), which means the user-topic distribution is incomparable among users. For the upper one, Alice is more active in topic *music*, but for the lower one, *Bob* is more active. In order to avoid such problem, we proposed a post-process method for the original LDA model to extract such information.

So the main contributions of this paper are: (1) we propose a joint model which can capture topic, expertise, activity and trends. Traditionally, these information have been modeled separately. (2) we propose a post-process method to solve a common problem in LDA based model, which is normally ignored. When LDA based models are applied, although they generate user-topic distributions, these outputs are actually incomparable among users and therefore less useful. Our proposed method can fix this problem and experiments show it can improve both our model and other LDA variants models' performance when it comes to the task of routing questions in a CQA site.

The rest of the paper is organized as follows. Firstly, we survey the state of the art of expertise modeling and temporal modeling approaches in CQA site, and point out the differences among these works. Secondly, we propose our TTEA model and explain it in

detail. Thirdly, we report the results of several experiments. They show that our approach can improve the question routing task. Finally, we discuss the performances of these methods and conclude on our contribution. We also provide details on our experiments in the Appendix.

2. RELATED WORK

Research related on expert identification in CQA are mainly based on link analysis and topic modeling techniques. The general purpose of expert detection is normally to support the question routing task which essentially consists in finding the most relevant experts to answer a newly submitted question. [19] proposed a modified PageRank algorithm to rank users in a specific domain. Besides, they proposed Z-score measure to evaluate expertise. Compared with simple statistic measures, for instance the number of best answers user provided, the Z-score measure uses both the number of questions and the number of answers a user posted. Similarly, [11] use HITS algorithm to discover authorities users. [3] proposed a model based on Indegree which is the number of best answers provide by users to discovery experts. [12] proposed a probability model to estimate users' expertise for question routing task.

Rather than detecting global experts, another kind of works uses topic models to detect topic level experts. [7] proposed a generative model by leveraging the category information of questions on certain CQA sites. [17] jointly model topics and expertise by integrating a Gaussian Mixture Model to capture vote information. [5] proposed a spectral clustering based topic model. [14] proposed a generative model to model the triple role of users (i.e., as askers, answerers, and voters, respectively). Our contribution extends that kind of works.

There are also works applying machine learning techniques to perform expert detection. [10] combine topic models outputs and statistic features and apply a pair-wised learning to obtain a ranked model and recommend expert users for a question. [15] apply machine learning algorithms to identify experts from their early behavior. [1] performed an in-depth study of Stack Overflow¹. They found that expert users tend to answer questions more quickly and gain high reputation by higher activity. Their work extract features and apply machine learning algorithms to predict whether a question has a long-term value and whether a question has been sufficiently answered. Their results show that votes information can indicate a user's expertise level while currently, these kind of work normally rely on the outputs of topic models.

Finally, there is an increasing research interest for the temporal modeling of online communities and several methods have been proposed. [16] jointly model topics and time label by assuming that words and time stamps are both generated by latent topics. [18] proposed a PLSA-based[8] model to separate temporary topics from stable topics. [9] jointly model latent user groups and temporal topics to detect group-level temporal topics.

Compared with these latest works, our model not only captures topics and expertise, it also can detect topic dynamics both at the global community level and at the individual user level. Besides, we proposed a post-process method to solve a common problem in LDA based models.

3. MODEL AND SOLUTION

¹<http://www.stackoverflow.com/>

3.1 LDA Model

The LDA model is originally applied for document-topic modeling. It trains a three layer documents-topics-words model to mine topics from documents. Some work such as [17] and [13] replace the documents by the users, and the document words by the contents generated by the user to construct a users-topics-words model which can then detect the topic distribution for each user. Let us take a look at a simple example of that. Suppose we have the results of a three-layer user-topic-word LDA model. It will mainly generate two kinds of distributions (user-topic distribution, and topic-word distribution) which separately describe to what extent a user is interested in different topics and to what extent a word is related to a topic.

However, there exist a problem. As shown in figure 1, we can find that the same user-topic distribution could be generated by different training data (Assume that the hidden variable topic is generated by Gibbs Sampling[6]), which means the user-topic distribution is incomparable among users. For the upper one, Alice is more active in *music*, but for the lower one, Bob is more active. In order to extract such information, we propose to add a post-process when generating the model output. The idea is quite simple, just normalizing the hidden variable in another dimension.

As shown in figure 1, if we normalize the data in the user dimension (left direction), we obtain a user-topic distribution, which is used to describe to what extent a user is interested in different topics. However, if we normalize the data in the topic dimension (right direction), we obtain a topic-user distribution, which is used to describe to what extent a topic is interested by different users. We use this post-process, which will be detailed explained in section 3.6 to extract the topic-user distribution.

3.2 Problem Definition

Let us consider StackOverflow for an example of the problem we address. In StackOverflow, as already explained, a user submits a question, then assigns between 1~5 tags to indicate the key domains of the question. Other users who are interested in the question may provide answers to the question. Both questions and answers will get votes from other users. For instance, Alice posts a question and assigns it the tags {*html*, *css*, *height*}. Her question then gets 30 votes, and Bob gives an answer to this question at 10/11/2015, that gets a voting score of 35.

The Temporal Topic Expertise Activity (TTEA) model we propose aims at jointly modeling topics, topic trends, user expertise, and user activities. More precisely, we aim at extracting the information listed in table 1.

Table 1: Output distributions of our model and their functionality

Notation	Functionality of distribution
θ_{uk}	detect a user's most interested topic
θ_{ku}	detect the most active users in a topic
θ_{kv}/θ_{kw}	detect the most relevant tags/words in a topic
θ_{kt}	detect the trends of a topic
θ_{tk}	detect the most popular topic at point in time
θ_{ukt}	detect a user's activity pattern in a topic
θ_{uke}	detect a user's most expertise topic

3.3 Basic Notions

Here are the basic notions later used in the description of TTEA:

Topic (θ_{kw}/θ_{kv}): A bag of words or tags which are closely related.

Words are the content of questions or answers, tags are attached to questions. For example, the topic-tag distribution $Database: \{mysql: 0.5, sql: 0.3, query: 0.2\}$ expresses that topic *Database* is related to tags *mysql*, *sql*, and *query*.

User Topical Interest(θ_{uk}): A user is interested in different topics with different levels. For example, the user-topic distribution $Alice: \{Database: 0.8, Java: 0.2\}$ expresses that *Alice* prefers to answer questions related to *Database*, but rather not about *Java*.

User Topical Activity(θ_{ku}): Different users are interested in the same topic with different levels. For example, the topic-user distribution $Database: \{Alice: 0.8, Bob: 0.2\}$ expresses that *Alice* prefers to answer question related to *Database*, while *Bob* is not willing to contribute answers to it.

Topic Trend(θ_{kt}): A topic is popular at different points in time with different levels. For example, the topic-time distribution $Database: \{May/2013: 0.2, June/2013: 0.3, July/2013: 0.5\}$ expresses that the topic *Database* is increasingly popular.

Topic Temporal Activity(θ_{tk}): Topics are active at a point in time with different levels. For example, the time-topic distribution $Sept/2013: \{Ios: 0.8, Database: 0.2\}$ expresses that *ios* related questions are popular in Sept. 2013, while *Database* related questions are not specially popular.

User Topic Temporal Dynamics(θ_{ukt}): A user is interested in different topics at different points in time with different levels. For example, the topic-time distribution for *Alice ios*: $\{May/2013: 0.2, June/2013: 0.3, July/2013: 0.5\}$ expresses that *Alice*'s interest to topic *ios* is increasing.

User Topical Expertise(θ_{uke}): A user has expertise in different topics with different levels. For example, the topic-expertise distribution for *Alice ios*: $\{High: 0.2, Medium: 0.7, Low: 0.1\}$ expresses that *Alice*'s expertise on topic *ios* is probably in medium level.

3.4 TTEA Model Structure

TTEA is an LDA-based model. Figure 2 represents it using the plate notation.

Let $u_i \in \{1, 2, \dots, U\}$ be the set of users, $p_i \in \{1, 2, \dots, P\}$ the set of answer posts, which are generated by these users, $w_i \in \{1, 2, \dots, W\}$ the set of words in answers posts, $ta_i \in \{1, 2, \dots, Ta\}$ the set of tags which are attached to posts, $v_i \in \{1, 2, \dots, V\}$ the set of votes for each answer posts, $ti_i \in \{1, 2, \dots, Ti\}$ the set of points in time which could be months or days depending on the requirements, and $z_i \in \{1, 2, \dots, K\}$ the set of topics for the posts. Here, U, P, W, Ta, V, Ti and K denote the total number of users, posts, words, tags, votes, points in time, and topics. $\alpha, \beta, \delta, \gamma, \eta$, and λ are Dirichlet priors. The notation and description of distributions $\theta_{uk}, \theta_{kv}, \theta_{kw}, \theta_{kt}$, and θ_{uke} are listed in Table 1.

Contrary to [2] who applied LDA model on long documents such as news articles and assumed that each word has a latent topic, we assume in TTEA that each answer post has one topic: like in other short social media, e.g. Twitter, an answer post is normally short, each answer post is therefore suitable to be assigned with one single latent topic, and all the words in that post are considered to be generated by this topic.

For expertise modeling, we do not use votes directly because (a)

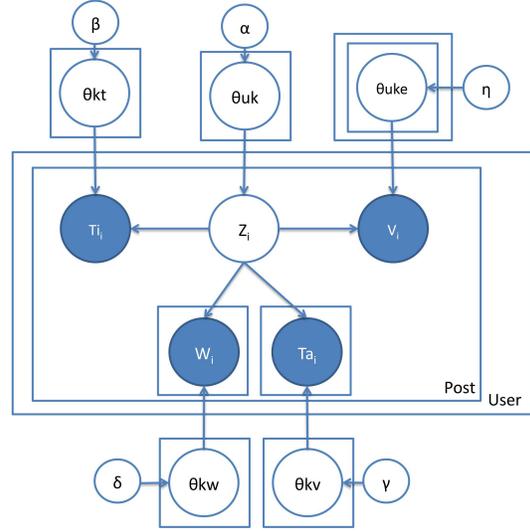


Figure 2: TTEA Model

the vote scores are sparse and noncontinuous, and (b) it is not reasonable to tell that a vote score 55 is better than a vote score 50 if the vote scores are ranging from 0 to 3000. Since the vote scores' counts distribution follows a log distribution[17], we use the logarithmic value of vote score, and separate them into several expertise levels, which is one of the parameters: the expertise level.

For temporal modeling, like [16] [9], we use time stamps directly. In order to model time at different levels, we simply split time stamps into different parts (month, day, and hour) and use them separately depending on the demands.

Let us consider a user u who wants to answer a question. She first selects a topic k according to her user-topic distribution θ_{uk} . Then she writes an answer post p . The words of p are generated from topic k 's topic-word distribution θ_{kw} . Since only the questions have tags, we consider the answers automatically acquire all the tags of the question they respond to. Then the answer post p acquires its tags according to the topic-tag distribution θ_{kv} of topic k . Meanwhile, the answer post p gets a time-stamp ti according to the topic-time distribution θ_{kt} of topic k . The generative process of TTEA model is described as follows.

- For the u -th user, $u \in U$
-draw user topic distribution $\theta_{uk} \sim \text{Dir}(\alpha)$
- For the k -th topic, $k \in K$
-draw topic tag distribution $\theta_{kv} \sim \text{Dir}(\gamma)$
-draw topic word distribution $\theta_{kw} \sim \text{Dir}(\delta)$
-draw topic time distribution $\theta_{kt} \sim \text{Dir}(\beta)$
- For the u -th user, $u \in U$
-for the k -th topic, $k \in K$
—draw user topic expertise distribution $\theta_{uke} \sim \text{Dir}(\eta)$
- For the u -th user, $u \in U$
-for the n -th q&a post, $p \in P$

- draw topic $z \sim \text{Multi}(\theta_{uk})$
- draw time point $t \sim \text{Multi}(\theta_{kt})$
- for the i -th word, $w \in W$
- draw word $w \sim \text{Multi}(\theta_{kw})$
- for the j -th tag, $ta \in Ta$
- draw tag $t \sim \text{Multi}(\theta_{kv})$
- draw expertise level $v \sim \text{Multi}(\theta_{uke})$

3.5 TTEA Model Inference

Like [9], we use the collapsed Gibbs Sampling algorithm [6] to sample the hidden variable z , based on which the unknown probabilities $\{\theta_{uk}, \theta_{kv}, \theta_{kw}, \theta_{kt}, \text{ and } \theta_{uke}\}$ can be estimated. For simplicity we set the hyper parameters to $\{\alpha, \beta, \delta, \gamma, \eta, \lambda\}$.

The TTEA inference process is as follows. We iteratively sample the topic indicator z_i for each answer post p_i according to equation 1. As explained before, each answer post will have one topic assignment.

$$\begin{aligned}
& p(z_i = k | z_{-i}, \mathbf{U}, \mathbf{Ti}, \mathbf{Ta}, \mathbf{W}) \\
& \propto \frac{C_{u,-i}^k + \alpha}{\sum_{k=1}^K C_{u,-i}^k + K * \alpha} \\
& \cdot \frac{\prod_{ta=1}^{Ta} \prod_{q=0}^{C_{ta}-1} (C_{k,-i}^{ta} + q + \gamma)}{\prod_{p=0}^{\sum C_{ta}-1} \sum_{ta=1}^{Ta} (C_{k,-i}^{ta} + p + Ta * \gamma)} \\
& \cdot \frac{\prod_{w=1}^W \prod_{s=0}^{C_w-1} (C_{k,-i}^w + s + \delta)}{\prod_{t=0}^{\sum C_w-1} \sum_{w=1}^W (C_{k,-i}^w + t + W * \delta)} \\
& \cdot \frac{C_{k,-i}^{t_i} + \beta}{\sum_{ti=1}^{T_i} C_{k,-i}^{ti} + T_i * \beta} \\
& \cdot \frac{C_{u,k,-i}^e + \eta}{\sum_{e=1}^E C_{u,k,-i}^e + E * \eta}
\end{aligned} \tag{1}$$

where $-i$ enforces that all the counters used are calculated with the answer post p_i excluded. $C_{u,-i}^k$ is the number of posts by user u assigned to topic k , C_{ta} is the number of tags ta in p_i , therefore, $\sum C_{ta}$ is the total number of tags in p_i , $C_{k,-i}^{ta}$ is the number of tags ta assigned to topic k . Similarly, C_w is the number of words w in p_i , $\sum C_w$ is the number of words in p_i , $C_{k,-i}^w$ is the number of words w assigned to topic k . $C_{k,-i}^{t_i}$ is the number of posts assigned to topic k and posted at time t_i . $C_{u,k,-i}^e$ is the number of posts which are assigned to topic k and got a vote score in the range of expertise level e .

Then, with the result of the Gibbs sampling algorithm, we can make the following parameter estimation:

$$\theta_{uk} = \frac{C_{u,-i}^k + \alpha}{\sum_{k=1}^K C_{u,-i}^k + K * \alpha} \tag{2}$$

$$\theta_{kv} = \frac{C_k^{t_a} + \gamma}{\sum_{ta=1}^{Ta} C_k^{ta} + Ta * \gamma} \tag{3}$$

$$\theta_{kw} = \frac{C_k^w + \delta}{\sum_{w=1}^W C_k^w + W * \delta} \tag{4}$$

$$\theta_{kt} = \frac{C_k^{t_i} + \beta}{\sum_{ti=1}^{T_i} C_k^{ti} + T_i * \beta} \tag{5}$$

$$\theta_{uke} = \frac{C_{u,k}^e + \eta}{\sum_{e=1}^E C_{u,k}^e + E * \eta} \tag{6}$$

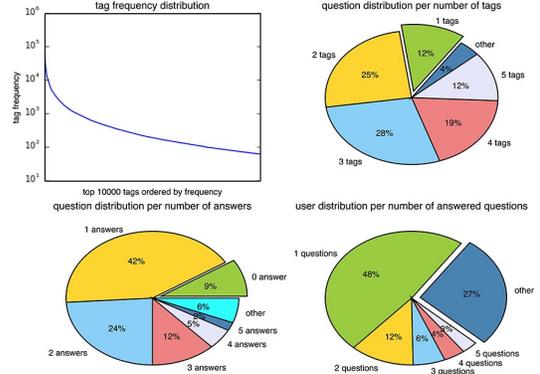


Figure 3: Basic perspectives of the dataset

3.6 Post Processing

The above model can only generate the distributions $\{\theta_{uk}, \theta_{kv}, \theta_{kw}, \theta_{kt}, \text{ and } \theta_{uke}\}$. To generate the other distributions, e.g. θ_{ku}, θ_{tk} and θ_{ukt} , we directly use the sample results at each iteration and keep recording the corresponding counters. Therefore, C_k^u is the number of posts assigned to topic k and posted by user u , C_{ti}^k is the number of posts posted at time ti and assigned to topic k . $C_{u,k}^{ti}$ is the number of posts by user u , assigned to topic k and posted at time ti . Then, we estimate $\theta_{ku}, \theta_{tk}, \theta_{ukt}$ according to the following equations:

$$\theta_{ku} = \frac{C_k^u + \alpha_1}{\sum_{u=1}^U C_k^u + U * \alpha_1} \tag{7}$$

$$\theta_{tk} = \frac{C_{ti}^k + \beta_1}{\sum_{k=1}^K C_{ti}^k + K * \beta_1} \tag{8}$$

$$\theta_{ukt} = \frac{C_{u,k}^{ti} + \lambda}{\sum_{ti=1}^T C_{u,k}^{ti} + T * \lambda} \tag{9}$$

4. EXPERIMENT

4.1 Dataset Description

We conducted experiments on a dataset from StackOverflow. This site releases its whole content every three month. For our experiments, we used the data dump from July 2008 to March 2013. Table 2 and figure 3 provide basic statistics on the dataset.

Table 2: Basic statistics on the dataset

number of tags	32,379
number of questions	4,592,961
number of users asking questions	833,041
number of users providing answers	8,585,113
number of questions having accepted answers	2,808,825

Here are some general observations about the dataset: (1) nearly half of the questions do not have accepted answers; (2) nearly half of the questions only have one answer and it maybe inadequate; (3) more than a third of the questions only have one or two tags; (4) nearly half of the users only answer one question so question routing and incentives are important problems; (5) nearly 10% percent of the questions do not have answers.

Due to the large volume of the dataset over 3 years, the processing time is extremely long. To simplify the processing, for the following experiments, we randomly chose several continuous months from the dataset, with no bias to the selections.

4.2 Compared Methods

To evaluate the effectiveness of our model, we compared it with several related works:

- TTEA is our method for modeling user, topic, temporal and expertise in Q&A sites. Besides, we also model activities by adding virtual nodes. We can generate the user-topic distribution and topic-activity distribution simultaneously.
- TEM: [17] proposed a model for user, topic and expertise in Q&A sites. It integrates a Gaussian Mixture Model to model expertise, which is time consuming. We simplify this process by directly modeling votes information. Besides, it does not model temporal information and user topic activities.
- UQA: [7] proposed a User-Question-Answer model for modeling users and topics in Q&A sites. In certain Q&A sites, questions have category information which have proved to be very useful. The category in their model is similar to tags in TTEA model and TEM model. However we allow multiple tags for each posts while they can only set a single category.
- GrosToT: [9] proposed a User-Group-Topic-Time model for modeling users, groups, topics and time in social media sites. It introduces a group level between user and topic compared with other models. It does not directly generate user-topic distribution, so we compute it with the user-group distribution and group-topic distribution.
- LDA: based on [2] we apply LDA model to create a User-Topic-Post model for modeling users and topics. It can generate the user-topic distribution and topic-words distribution.

We choose the same number of topics $K=30$ as [5] and the same number of expertises $E=10$ as [17], which have proved to be a reasonable setting for the Stackoverflow dataset. We empirical set Dirichlet hyper parameters $\alpha=\alpha_1=50/K$, $\beta=\beta_1=0.01$, $\delta=\lambda=\eta=0.01$, $\gamma=0.001$ according to suggestions in [6].

4.3 Performance of Topic Extraction

Table 3 and Table 4 show the top tags and words detected by our model. We use again the Perplexity [2] metric as a quantitative way to measure the performance of topic extraction.

We include in our training dataset all the posts in the two months from August 1st 2011 to October 1st 2011, from users having more than 80 posts (as in [17]). The resulting training dataset contains 87516 q&a posts by 674 users. For data preprocessing, we tokenize text and removed the stop words. For the testing dataset, we use all the posts of the same set of users than the training data but this time from October 1th 2011 to January 1th 2012. So training and testing datasets have no overlap but concern the same community. We vary the number of topics: 10, 30, 50, and 100. For a testing set of M posts, N_i denotes the number of words in the i^{th} post and the Perplexity score is computed according to equation 10.

$$Perplexity(D_{test}) = \exp \left\{ - \frac{\sum_{i=1}^M \log p(W_i)}{\sum_{i=1}^M N_i} \right\} \quad (10)$$

where $p(W_i)$ is the probability of the words in the test document d_i . In our model, $p(W_i)$ is computed according to equation 11.:

$$P(W_i) = \sum_k \theta_{u_i k} \prod_w \theta_{kw_i} \quad (11)$$

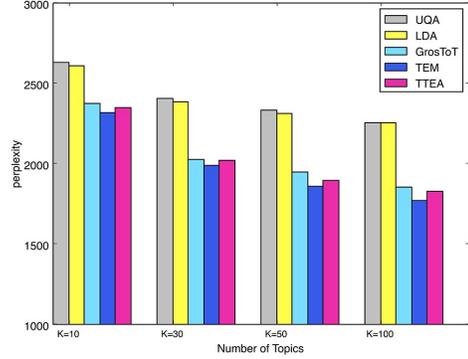


Figure 4: Comparison of topic extraction performances

Figure 4 shows the perplexity results for our TTEA method and other state-of-the-art methods. TTEA is almost as good as TEM. But TEM integrates a Gaussian Mixture Model, which is time consuming. The training process of TEM is nearly three times longer than the other models.

[4] suggested that topic models should focus on evaluations on real-world task performance rather than on optimizing likelihood-based measures. So, in addition to the perplexity-based evaluation, we used the results of TTEA to perform real-word tasks and we evaluated them. This is described in the following subsections.

4.4 Question Routing

Given a question q and a set of users U , the task is to rank all these users by their interests to answer question q . We score each user u by considering the similarity between his topics of interest and the topics of the question ($Sim(u, q)$). The intuition behind equation 12 is that the more a user is interested in the topic of a question, the more likely he is to provide an answer to that question.

$$Sim(u, q) = (1 - JS(\theta_{uk}, \theta_{qk})) \quad (12)$$

where θ_{uk} is the user topic interest distribution, θ_{qk} is the question topic distribution, and $JS(\cdot)$ is the Jensen-Shannon divergence distance. We obtain θ_{uk} directly from model results. For θ_{qk} , we apply equation 13.

$$\begin{aligned} \theta_{q, k} &\propto p(k|w_q, t_q, u) \\ &= p(k|u)p(w_q|k)p(t_q|k) \\ &= \theta_{uk} \sum_{w_i \in w_q} \theta_{kw_i} \sum_{t_i \in t_q} \theta_{kt_i} \end{aligned} \quad (13)$$

where w_q and t_q are the sets of all the words and tags in question q and θ_{kw} , θ_{kt} are the topic-word distribution and topic-tag distribution obtained directly from the model result. Then for question q , we compute the Sim score for user set U and rank them in decreasing order.

We used all the posts from July 1th 2011 to October 1th 2011 from users having more than 50 q&a posts for the training dataset.

Table 3: Top tags for different topics generated by the TTEA model

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
php	c	iphone	c++	javascript	android	sql	java	jquery	git
xslt	.net	objective-c	c	jquery	java	mysql	spring	javascript	svn
xml	linq	ios	pointers	php	android-layout	sql-server	eclipse	html	version-control
xpath	generics	xcode	templates	ajax	listview	php	jsp	css	github
mysql	asp.net	cocoa-touch	stl	html	activity	query	.htaccess	jquery-selectors	mercurial
html	vb.net	ipad	arrays	json	android-intent	tsql	servlets	jquery-ui	eclipse
arrays	c-4.0	uitableview	vector	asp.net	sqlite	sql-server-2008	jsf	dom	tortoisenvn
jquery	reflection	iphone-sdk-4.0	string	jquery-ajax	layout	join	mod-rewrite	php	linux
javascript	entity-framework	cocoa	function	forms	android-widget	select	maven	javascript-events	clearcase
foreach	list	xcode4	c++11	asp.net-mvc-3	xml	sql-server-2005	apache	ajax	ssh

Table 4: Top words for different topics generated by the TTEA model

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
xsl	aspx	view	std	jquery	android	select	html	jquery	git
td	msdn	reference	const	ajax	activity	join	java	div	branch
tr	microsoft	nsstring	pointer	script	html	group	file	click	commit
template	library	apple	char	javascript	view	order	spring	element	file
select	select	html	template	page	developer	table	jar	event	svn
row	linq	library	vector	html	intent	key	apache	input	repo
echo	system	documentation	operator	form	reference	count	eclipse	document	repository
table	dictionary	developer	compiler	url	layout	row	docs	text	files
match	ienumerable	ios	memory	document	try	inner	servlet	html	master
node	expression	release	struct	json	button	query	web	api	github

Rather than using threshold of 80 post like in [17], we empirically set it to 50 posts to get enough users for recommendation. The resulting training set contains 297881 posts by 2555 users. For the testing dataset, we use all the questions posted by the same set of users as in the training set but this time from October 1th 2011 to January 1th 2012. Therefore the training and testing datasets have no overlaps. We removed testing questions which have no, or only one, answer. The resulting test dataset contains 6044 questions, 18077 answers and 7888 involved users.

We also chose another period for this experiment. Besides, we vary the number of topics by 15 and 50, we vary the filter limit by 40 and 80. These experiment results are shown in section appendix.

In order to evaluate different models, we consider precision at position N (Precision@N or simply P@N) and recall at position N (Recall@N or simply R@N), which are widely used measures in the Information Retrieval community. Let R_q be the recommendations of users for a question q and U_q be the actual set of users who posted for question q . Then Precision@N is defined in equation 14 and Recall@N is defined in equation 15.

$$P@N = \frac{1}{|Q|} \sum_{q \in Q} \frac{|R_q \cap U_q|}{|R_q|} \quad (14)$$

$$R@N = \frac{1}{|Q|} \sum_{q \in Q} \frac{|R_q \cap U_q|}{|U_q|} \quad (15)$$

where Q is the set of testing questions. Like in [5], we use the Matching Set Count (MSC) which is defined in equation 16. The idea is to count the number of successful recommendations, i.e., for which at least one of the recommended users answered the ques-

tion.

$$MSC@N = \frac{1}{|Q|} \sum_{q \in Q} 1[R_q \cap U_q \neq \emptyset] \quad (16)$$

where $1[condition]$ is equal to 1 if *condition* is true, otherwise 0.

In addition, our model can capture activity and we believe this information improves question routing. The intuition is that even if a user has a high *Sim* score for a question, the less he is active, the less likely he is to provide an answer to that question. Therefore, we define a score *SimAct* to combine both topic similarity and activity level as shown in equation 17, where $Act(u, q)$ is the computed activity score for user u to question q . A high value of the *Act* score indicates a high probability of activity on a question. We use TTEA to denote the method using only the similarity information, that is to say, ranking users by *Sim* score. We use TTEA-ACT to denote the method using both similarity and activity, that is to say, ranking users by *SimAct* score. We also integrated our activity model to the TEM model and we refer to it as TEM-ACT.

$$\begin{aligned} SimAct(u, q) &= (1 - JS(\theta_{uk}, \theta_{qk})) * Act(u, q) \\ &= (1 - JS(\theta_{uk}, \theta_{qk})) * \sum_{k=1}^K \theta_{qk} * \theta_{ku} \end{aligned} \quad (17)$$

Table 5 shows the results. We ran the experiments five times and listed the average scores. Our observations can be summarized as follows: (1) UQA and GROSTOT perform the better when the number of recommended users are small, and TTEA and TEM begin to outperform UQA and GROSTOT when the number of recommended users is large; (2) TTEA-ACT shows the best performances compared with the baseline competitors; (3) both TTEA-ACT and TEM-ACT perform better than the other models. The activity modeling is a generic method that could improve the per-

Table 5: Question Routing experiments, Random denotes that we randomly recommend users for the test questions.

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc@20	msc@30
TTEA	0.024	0.019	0.015	0.013	0.045	0.072	0.111	0.142	0.112	0.178	0.269	0.339
TTEA-ACT	0.028	0.022	0.017	0.014	0.052	0.083	0.127	0.159	0.134	0.209	0.313	0.382
TEM	0.024	0.019	0.015	0.013	0.045	0.073	0.114	0.146	0.114	0.179	0.275	0.344
TEM-ACT	0.029	0.023	0.018	0.015	0.054	0.084	0.129	0.162	0.137	0.210	0.315	0.388
UQA	0.030	0.019	0.012	0.010	0.062	0.075	0.095	0.112	0.149	0.179	0.224	0.261
GROSTOT	0.027	0.017	0.011	0.009	0.055	0.067	0.085	0.099	0.134	0.164	0.204	0.236
RANDOM	0.001	0.001	0.001	0.001	0.001	0.002	0.005	0.007	0.003	0.007	0.013	0.019

formance not only of our model, but also of other models although here we only show the result for the activity model with TEM as an example; (4) even if TEM or TEM-ACT perform better than our model they remain again time consuming. Experiments show that the training process takes around 3~4 times longer compared to our model.

4.5 Recommendation of Expert Users

Given a question q and a set of users U , the task is now to recommend N users until one of the users gets the highest vote. The point is to rank recommended users by their expertise to answer question q . We score each user u by considering the similarity $SimExp(u, q)$ between user topic interest and user topic expertise to answer question q . The intuition behind equation 18 is that if the user is interested in the question, she will probably provide an answer to that question and if the user has expertise on the question, the answer will probably have the highest vote score.

$$SimExp(u, q) = (1 - JS(\theta_{uk}, \theta_{qk})) * Exp(u, q) \quad (18)$$

where θ_{uk}, θ_{qk} is the same than in 12 for user topic interest distribution. For our method, we compute $Exp(u, q)$ by equation 19

$$Exp(u, q) = \sum_{e=1}^E \theta_{kue} * e \quad (19)$$

As UQA and GROSTOT do not model expertise, like [17], we set $Exp(u, q)$ to 1 for these two methods. For TEM, we reuse equation 20 indicated in [17].

$$Exp(u, q) = \sum_{e=1}^E \phi_{z,u,e} * \mu_e \quad (20)$$

In order to evaluate different models, we consider the percentage of successful expert recommendation until position N . A successful expert recommendation until position N means that the N -th user, recommended by an algorithm, not only answers the question but also gets the highest votes.

Table 6: Expert recommendation experiments

Methods	N=30	N=60	N=100
TEM	0.128	0.228	0.392
TTEA	0.079	0.195	0.443
UQA	0.146	0.206	0.261
Grostit	0.127	0.172	0.220
Random	0.008	0.018	0.028

Table 6 shows the results. Random denotes that we randomly recommend users for the test questions. We ran the experiments five

times and listed the average scores. We summarize our observations as follows: (1) Our TTEA shows the best performances compared with the baseline models when the number of recommended users is large. This means that when we recommend 100 users for each testing questions, in around 44% cases we have one user not only answering the question, but also winning the highest vote. (2) When the number of recommended users is large, both TEM and TTEA perform better than other models which do not model expertise, so expertise modeling can improve expert recommendation. (3) TEM uses Gaussian Mixture Model to model expertise, while we directly model votes which is less precise. Therefore, we perform badly when the number of recommended users is small. (4) After ranking users by topic similarity scores, using expertise scores to re-rank those users actually lowers the probability of the top ranked user to answer the question. The intuition behind is that a user having high expertise on a question does not necessarily have high topic similarity score with the question.

4.6 Trends

With the temporal modeling of TTEA, we can explore topic dynamics at many different levels. We present illustrative case studies to show the advantage of temporal modeling.

We first set the time window at the month level. Figure 5-a shows the dynamics of *Android*, *Iphone* and *Flash* related topics at different months from Jan 2011 to Dec 2011. *Flash* related topics are more active in the early of 2011, but become less popular in the late of 2011. We then set the time window at the day level. Figure 5-b shows the dynamics of *Android*, *Iphone* and *Flash* related topics from July 1st 2011 to July 31st 2011. We can see that all topics are active from Monday to Friday, and not active during the weekend. Lastly, we set the time window at the hour level. Figure 5-c shows the dynamics of *Android*, *Iphone* and *Flash* related topics at different hours during a day. We can verify that both *Android* and *Iphone* related topics are more active during daytime, but *Flash* related topics are more active during the afternoon.

Previous figures show the topic dynamics on a global level. We now illustrate the topic dynamics at the user level. We choose top active users according to the output of θ_{ku} in *Android* related topic and *Iphone* related topic separately. Figure 6-a,b show the activity pattern of the two most active users in *Iphone* related topic. We can observe that the user in Figure 6-a is only active during work-time. The user seldom answers questions after 7PM. On the contrary, the user in Figure 6-b is active until very late but not midnight. Figure 6-c,d show the activity pattern of the two most active users in *Android* related topic. We can observe that the user in Figure 6-c is active in the morning, afternoon and evening. On the contrary, the user in Figure 6-d is even active at midnight. For all these users, we can observe that they are not actually active on the topics they are not interested in. We believe this information will benefit many

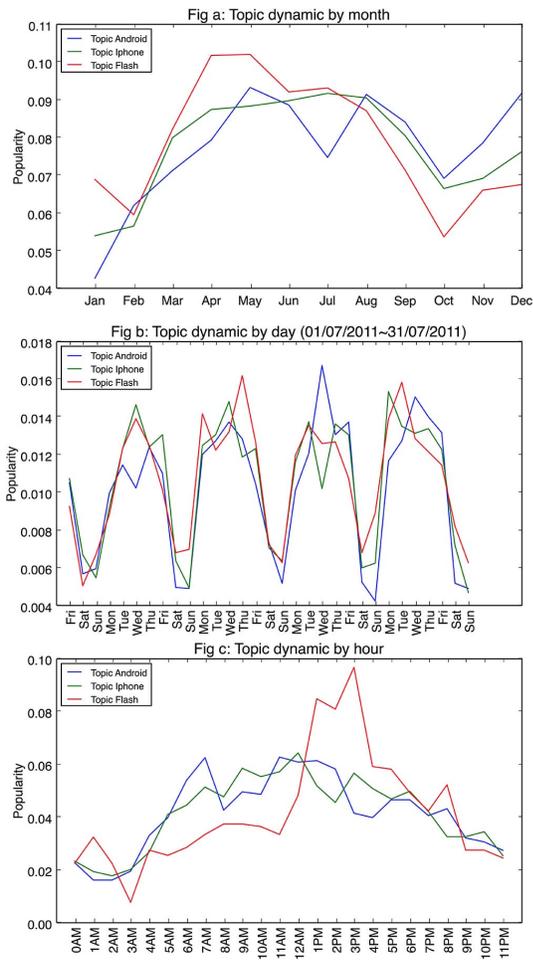


Figure 5: Topic dynamics

community management related tasks.

5. CONCLUSION

In this paper, we addressed the problem of topic detection, activity modeling, temporal modeling and expertise detection in CQA sites. We point out the *incomparability* problem of LDA based models. We proposed an post process to solve this problem. We then presented the TTEA (Temporal Topic Expertise Activity) model that simultaneously uncovers the topics, activities, expertise and temporal dynamics. This extracted information can enable us to improve tasks such as: question routing, expert recommending and community life-cycle management. We demonstrated that TTEA shows advantages in topic modeling. It also achieves good performances on question routing task and expert detection task compared with the state of the art models. There are still many future directions for this work, for instances, it is obviously that the model is not limit to CQA dataset and we intend to adapt our model to other kinds of social media.

Acknowledgment

The authors would like to thank StackOverflow for sharing their data. We thank the ANR-12-CORD-0026 Ocktopus project grant for the support of this research.

6. REFERENCES

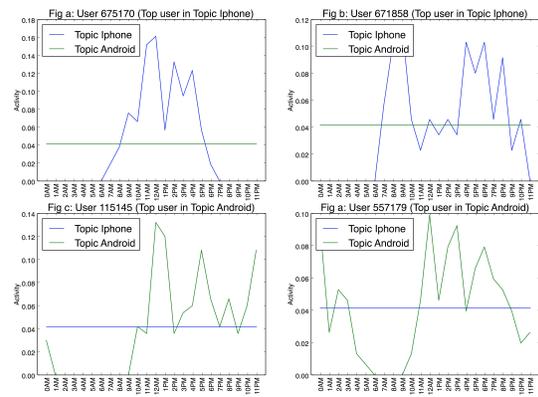


Figure 6: User Topic Activities

- [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 *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 850–858. ACM, 2012.
- [2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [3] M. Bouguessa, B. Dumoulin, and S. Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 866–874. ACM, 2008.
- [4] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009.
- [5] S. Chang and A. Pal. Routing questions for collaborative answering in community question answering. In *Proceedings of the 2013 IEEE/ACM ASONAM*, pages 494–501, New York, NY, USA, 2013. ACM.
- [6] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.
- [7] J. Guo, S. Xu, S. Bao, and Y. Yu. Tapping on the potential of q&a community by recommending answer providers. In *Proceedings of the 17th ACM CIKM*, pages 921–930. ACM, 2008.
- [8] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.
- [9] Z. Hu, J. Yao, and B. Cui. User group oriented temporal dynamics exploration. In *Twenty-Eighth AAAI14*, 2014.
- [10] Z. Ji and B. Wang. Learning to rank for question routing in community question answering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2363–2368. ACM, 2013.
- [11] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 919–922. ACM, 2007.
- [12] B. Li and I. King. Routing questions to appropriate answerers

- in community question answering services. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1585–1588. ACM, 2010.
- [13] D. Li, B. He, Y. Ding, J. Tang, C. Sugimoto, Z. Qin, E. Yan, J. Li, and T. Dong. Community-based topic modeling for social tagging. In *Proc. of the 19th ACM CIKM*, CIKM '10, pages 1565–1568, New York, NY, USA, 2010. ACM.
- [14] Z. Ma, A. Sun, Q. Yuan, and G. Cong. A tri-role topic model for domain-specific question answering. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [15] A. Pal, R. Farzan, J. A. Konstan, and R. E. Kraut. Early detection of potential experts in question answering communities. In *User Modeling, Adaption and Personalization*, pages 231–242. Springer, 2011.
- [16] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.
- [17] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen. Cqrank: jointly model topics and expertise in community question answering. In *Proceedings of the 22nd ACM CIKM13*, pages 99–108. ACM, 2013.
- [18] H. Yin, B. Cui, H. Lu, Y. Huang, and J. Yao. A unified model for stable and temporal topic detection from social media data. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 661–672. IEEE, 2013.
- [19] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, pages 221–230. ACM, 2007.

Table 11 shows the question routing results with users having more than 80 posts. We use the same period of dataset used in section 4.4. Due to the different filter limit, the training set contains 1275 users and 216940 q&a posts, the testing set contains 2589 questions, 8006 answers and 4196 involved users.

Appendix: Experiment Parameter Sensitivity Analysis

For the training dataset, we used all the posts in a three months period, from January 1th 2011 to March 31th 2011, from users having at least 50 q&a posts, rather than 80 posts like [17], in order to get enough users for recommendations. The training set contains 371181 posts by 3123 users. For the testing dataset, we used all the questions posted by the same set of users as in the training set, but this time from April 1th 2011 to June 31th 2011. Therefore the training and testing datasets have no overlaps. We removed questions with no or only one answer. The resulting test dataset contains 9048 questions, 27870 answers and 10147 users. Table 7 shows the question routing results. We can still find that TTEA-ACT outperforms all the baseline models. Besides, Both TTEA-ACT and TEM-ACT outperform all the other models.

Table 8 shows the question routing results with a number of topics set to 15. We use the same training and testing datasets as in section 4.4.

Table 9 shows the question routing results for the number of topics set to 50. We use the same training and testing datasets as in section 4.4.

Table 10 shows the question routing results with users having more than 40 posts. We use the same period of dataset used in section 4.4. Due to the different filter limit, the training set contains 3457 users and 338485 q&a posts, the testing set contains 8579 questions, 25500 answers and 10135 involved users.

Table 7: Question Routing Experiments on Another Dataset

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.026	0.020	0.015	0.013	0.047	0.073	0.110	0.136	0.123	0.186	0.273	0.332
TTEA-ACT	0.032	0.026	0.019	0.016	0.058	0.093	0.137	0.168	0.153	0.236	0.339	0.405
TEM	0.025	0.021	0.016	0.013	0.047	0.076	0.112	0.139	0.120	0.191	0.274	0.333
TEM-ACT	0.032	0.025	0.020	0.016	0.058	0.092	0.141	0.171	0.153	0.235	0.348	0.411
UQA	0.027	0.016	0.011	0.009	0.052	0.062	0.080	0.096	0.130	0.155	0.196	0.233
GROSTOT	0.023	0.014	0.009	0.007	0.044	0.055	0.069	0.081	0.112	0.137	0.172	0.200
RANDOM	0.001	0.001	0.001	0.001	0.001	0.002	0.004	0.005	0.003	0.005	0.010	0.015

Table 8: Question Routing experiments with 15 topics

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.016	0.013	0.012	0.010	0.030	0.050	0.086	0.112	0.076	0.127	0.213	0.269
TTEA-ACT	0.023	0.018	0.015	0.012	0.042	0.066	0.107	0.134	0.112	0.170	0.268	0.329
TEM	0.017	0.015	0.012	0.010	0.032	0.054	0.091	0.115	0.083	0.137	0.222	0.276
TEM-ACT	0.024	0.018	0.014	0.012	0.043	0.068	0.103	0.131	0.114	0.172	0.254	0.319
UQA	0.028	0.016	0.011	0.008	0.056	0.066	0.083	0.099	0.137	0.159	0.199	0.238
Grostot	0.023	0.015	0.010	0.008	0.045	0.058	0.075	0.089	0.112	0.143	0.183	0.216
Random	0.001	0.001	0.001	0.001	0.002	0.003	0.004	0.006	0.005	0.008	0.012	0.017

Table 9: Question Routing experiments with 50 topics

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.028	0.023	0.018	0.015	0.054	0.087	0.132	0.168	0.134	0.215	0.319	0.394
TTEA-ACT	0.033	0.025	0.019	0.016	0.063	0.095	0.142	0.178	0.158	0.235	0.343	0.418
TEM	0.029	0.024	0.018	0.015	0.056	0.088	0.136	0.171	0.141	0.220	0.325	0.400
TEM-ACT	0.033	0.026	0.020	0.017	0.062	0.096	0.145	0.182	0.157	0.240	0.347	0.427
UQA	0.032	0.019	0.012	0.010	0.065	0.077	0.097	0.116	0.158	0.185	0.227	0.270
Grostot	0.028	0.017	0.011	0.009	0.056	0.067	0.088	0.102	0.136	0.163	0.210	0.241
Random	0.001	0.001	0.001	0.001	0.002	0.002	0.005	0.007	0.004	0.006	0.013	0.018

Table 10: Question Routing experiments, with users having more than 40 posts

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.021	0.018	0.014	0.012	0.040	0.067	0.104	0.132	0.100	0.167	0.253	0.313
TTEA-ACT	0.026	0.021	0.016	0.014	0.049	0.076	0.118	0.149	0.126	0.193	0.292	0.360
TEM	0.023	0.018	0.014	0.012	0.043	0.069	0.106	0.137	0.109	0.170	0.255	0.323
TEM-ACT	0.027	0.021	0.016	0.014	0.050	0.078	0.121	0.152	0.128	0.194	0.295	0.362
UQA	0.029	0.018	0.011	0.009	0.059	0.071	0.087	0.101	0.142	0.169	0.205	0.235
Grostot	0.025	0.016	0.010	0.008	0.050	0.063	0.077	0.091	0.122	0.152	0.188	0.217
Random	0.000	0.000	0.000	0.000	0.001	0.002	0.003	0.005	0.002	0.004	0.008	0.013

Table 11: Question Routing experiments, with users having more than 80 posts

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.028	0.023	0.019	0.016	0.051	0.083	0.135	0.175	0.132	0.212	0.336	0.424
TTEA-ACT	0.031	0.026	0.020	0.018	0.058	0.094	0.146	0.188	0.150	0.238	0.364	0.457
TEM	0.031	0.026	0.020	0.017	0.056	0.095	0.147	0.188	0.143	0.238	0.356	0.445
TEM-ACT	0.035	0.027	0.021	0.018	0.063	0.100	0.151	0.193	0.165	0.253	0.375	0.468
UQA	0.040	0.025	0.016	0.013	0.077	0.096	0.124	0.150	0.194	0.237	0.299	0.357
Grostot	0.036	0.022	0.015	0.012	0.070	0.086	0.114	0.135	0.177	0.214	0.278	0.325
Random	0.001	0.001	0.001	0.001	0.002	0.003	0.006	0.011	0.005	0.008	0.019	0.030