Finding Expert Users in Community Question Answering

Fatemeh Riahi
Faculty of Computer Science
Dalhousie University
riahi@cs.dal.ca

Zainab Zolaktaf
Faculty of Computer Science
Dalhousie University
zolaktaf@cs.dal.ca

Mahdi Shafiei
Department of Mathematics
and Statistics
Dalhousie University
shafiei@cs.dal.ca

Evangelos Milios
Faculty of Computer Science
Dalhousie University
eem@cs.dal.ca

ABSTRACT

Community Question Answering (CQA) websites provide a rapidly growing source of information in many areas. This rapid growth, while offering new opportunities, puts forward new challenges. In most CQA implementations there is little effort in directing new questions to the right group of experts. This means that experts are not provided with questions matching their expertise, and therefore new matching questions may be missed and not receive a proper answer. We focus on finding experts for a newly posted question. We investigate the suitability of two statistical topic models for solving this issue and compare these methods against more traditional Information Retrieval approaches. We show that for a dataset constructed from the Stackoverflow website, these topic models outperform other methods in retrieving a candidate set of best experts for a question. We also show that the Segmented Topic Model gives consistently better performance compared to the Latent Dirichlet Allocation Model.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Text Mining; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Statistical Computing*

Keywords

Community Question Answering, Topic Modeling, Language Model, TF-IDF, Expert Recommendation

1. INTRODUCTION

Community Question-Answering (CQA) websites archive millions of questions and answers and provide a valuable resource that cannot be easily obtained using web search engines. Providing good quality answers to users' questions through collaboration of a community of experts is the main purpose of these services. Voting, badges and reputation are examples of mechanisms provided by some CQA services to assure the quality of questions and answers.

In current CQA services, a user who submits her question is required to either (i) wait for other users to post answers to the question which may take several days and sometimes results in incorrect, spam or offensive answers (ii) or use the archives of CQA sites. These archives often contain restricted answer sets and the user has to deal with the word-match constraint between her formulated question and archived questions [22].

The main problem of CQA services is the low participation rate of the users. It means that only a small portion of users are responsible for answering a notable number of questions. Two main reasons of low participation are: (i) Most users are not willing to answer questions or are not experts. (ii) Those users willing to answer questions are not aware of the new questions of interest to them [13].

Developing a system capable of finding experts for newly posted questions can contribute to the creation of high-quality answers for questions and mitigate the problem of low participation rates. The goal of expert finding is to return a ranked list of experts with expertise on a given topic. An essential part of an expert-finding task is the ability to model the expertise of a user based on her answering history.

Common methods for finding experts can be divided into two categories. The first category searches for relevant answers for a given question and then retrieves a ranked list of users based on their contribution to those answers. The second category builds a profile for each expert based on her activity and past answers and then uses these profiles to find experts. Our research falls into the second category by building a profile for each user and finding experts using these profiles.

Most of the current work in the latter category models user profiles by using classical information retrieval approaches. These approaches use lexical similarity measures and retrieve good results if sufficient word overlap exists. However, there is often little word overlap between new questions and user profiles, therefore these approaches may not lead to satisfactory results.

In this paper, we focus on the second cause of low participation rate in CQA mentioned above. Our objective is to route new questions to the best suited experts. We model the interests of users by tracking their answering history in the community. For each user, a profile is created by combining those questions answered by the user for which she has been selected the best answerer. Based on the user profiles, the relation between the answerer and a new question

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

is measured by using a number of different methods. These methods include language models with Dirichlet smoothing, TF-IDF, the Latent Dirichlet Allocation (LDA) and Segmented Topic Model (STM).

Questions posted on CQAs are usually short in length. A question may be semantically similar to a user profile but still lexically very different. Therefore, an expert recommender system that is capable of capturing the semantic similarities between the question and user profiles may achieve better results.

LDA and STM model the latent topics in user profiles to capture the semantic structure of user profiles. However, STM is more focused on taking advantage of the structure of CQAs and extracts more semantic information from the profiles. For using LDA, all questions answered by a user are concatenated together and build the user profile. However, a user is likely to have answered questions in different topics. STM, on the other hand, treats each question individually while considering all questions answered by the user as her profile. Our experimental results indicate that STM performs much better than LDA in retrieving a candidate set of best experts for a question.

Evaluating systems for expert finding is not a simple task. In our dataset, which is a snapshot of Stackoverflow, we have the actual best answerer for each test question and we use it to evaluate performance of a method. However, it is quite likely that other users returned by the system are also good answerers for the questions. In order to detect the relevancy of other returned users to the test question, a user study would be required.

2. RELATED WORK

In this section, we review some of the research on community question answering, expert recommendation and topic analysis using statistical models.

2.1 Community Question Answering

In the past few years, Community Question Answering websites such as Yahoo answers have been building very large archives of questions and their answers [1, 13, 26, 18].

Research on community question answering has seen a significant growth. One of the main goals of this research is to decrease the waiting time for a personal response. Relying on the available archive, one can approach this problem by either finding similar questions or relevant answers.

Relying only on the questions available in the archive, the objective is to find similar previously answered questions by the QA community. The problem of question recommendation is tackled in [7] by representing questions as graphs of topic terms, and then ranking recommendations on the basis of these graphs. An automatic method for finding questions that have the same meaning is proposed in [15] . This method finds semantically similar questions that have little word overlap.

Including the answers available in the archive, the main purpose is to find a right answer in QA archive for a given question. To build an answer finding system, four statistical techniques are used in [4] including TF-IDF, adaptive TF-IDF, query expansion and statistical translation. A semantic knowledge base (WordNet) is used in [6] to improve the ability of classical information retrieval approaches in matching questions and answers. Additionally, non textual features are used to improve the answer search quality in [16].

2.2 Expert Recommendation

Compared to the previous problem of retrieving relevant questions and answers for a new question, there are fewer works aiming to solve the problem of finding the best answerers for a new question. The task of expert recommendation is predicting the best users who can answer a newly posted question. A ranked list of best answerers can be returned based on the similarity between the query and users history. To locate users with desired expertise, quantitative measures of expertise are defined in [21]. They described how to obtain these measures from a software project's change management system. They also presented evidence to validate this quantification as a measure of expertise. Two general strategies for expert searching given a document collection are presented in [3] by using generative probabilistic models. Experts are found by mining expertise from email communications in [9]. Profile-based models for expert finding on general documents are proposed in [11].

There is also some research in question answerer recommendation. A new topic model which can simultaneously discover topic distribution for words, categories and users in a QA community is introduced to find a ranked list of answer providers [13]. Latent Dirichlet Allocation model is used in [17] and it has been combined with user activity and authority information to find the best answerers.

2.3 Topic Analysis Using Statistical Models

The use of topic models for information retrieval tasks is described in [23]. They found that the combination of Dirichlet smoothed language models and topic models lead to significant improvements in retrieval performance compared to using only the language models.

The most popular model for text retrieval is Vector Space Model [2]. However, this model suffers from high dimensionality when representing documents using the "bag of words“ assumption. Latent Semantic Indexing (LSI) [8] is one way to reduce the space dimension but it lacks semantic interpretation. To overcome this problem, pLSI [14] introduces latent topics to represent documents and model the data generation process as a Bayesian network. A novel model, Author-Persona-Topic (APT), is introduced in [20] to recommend the best reviewers for a given paper by dividing authors' papers into several "personas“. Each persona clusters papers with similar topical combinations. A new topic model, the author-topic model (ATM), is proposed in [27], for exploring the relationships between authors, documents, topics and words.

3. PROBLEM STATEMENT

Given a new question q, we need to return a ranked list of users u_1, u_2,...,u_n who are best suited to answer q. The probability of a user u, being the answerer for the question q is:

$$P(u|q) = \frac{P(u)P(q|u)}{P(q)} \tag{1}$$

where $P(q)$ is the probability of question q and we assume it is the same for all the test questions. $P(u)$ is the prior probability of user u which can be approximated by specific information such as user activity derived from the dataset.

In this study, our objective is to compute the probability $P(q|u)$ that captures the expertise of user u on question q. This probability model was first introduced by [3].

In the dataset used in this research, each question has three parts: question tag, question title, and question body. Question tag is the tag assigned by user who posted the question. Question title is a short description of the question. The detailed description is given in the question body. The main challenge is representing the questions. Additionally, the expertise and interest of a user should be modeled by taking advantage of the activity history of the user.

4. MODELING EXPERT SEARCH

In the Community Question Answering Services, answerers usually choose a category that they are more interested in and then pick a question from that category. Therefore, user interests can be inferred from answering history. In this section, we explore different methods for ranking users based on their interests. These methods can be divided into two main categories: word-based methods and topic models. In the first category, we model user interest by using TF-IDF and language model. In general, word-based methods use a smoothed distribution to estimate the likelihood of a query in a given collection of documents. Topic models have an additional representational level. Documents in these models are a mixture of topics and topics are mixtures of words.

4.1 TF-IDF

TF-IDF is a standard measure to compute importance and relevance of a word in a document based on the frequency of that word in the document and the inverse proportion of documents containing the word over the entire document corpus. Words that appear only in a small group of documents will have higher tf-idf scores than other words. Basically, TF-IDF is defined as follows: given a document collection Q, a word w, and a document $d \in Q$:

$$tfidf = f_{\mathrm{w,d}} * \log(\frac{|Q|}{f_{\mathrm{w,Q}}}) \tag{2}$$

where $f_{\mathrm{w,d}}$ is the number of times w appears in d, $|Q|$ is the size of the corpus and $f_{\mathrm{w,Q}}$ is the total number of documents that have the word w [28].

The end result is a term-by-document matrix for the entire corpus, where columns represent terms, rows represent documents and each value in the matrix represents the tf-idf weight for the corresponding term and document. Thus, the TF-IDF reduces documents of different lengths to vectors with a fixed length.

For the expert retrieval task, given a test question q composed of a set of words, we represent test question and each profile as vectors of their tf-idf weights and then calculate the Cosine Similarity between each user profile and question vector:

$$s(u,q) = \frac{\sum_w tfidf(u,w) tfidf(q,w)}{\sqrt{\sum_w tfidf(u,w)^2}\sqrt{\sum_w tfidf(q,w)^2}} \tag{3}$$

where $tfidf(q,w)$ is the tf-idf weight of word w in q, and $tfidf(u,w)$ is the tf-idf weight of w in the profile of user u.

4.2 Language Model

Language Model is related to traditional TF-IDF models. Similar to TF-IDF, rare terms in the corpus which occur in only a group of documents in the corpus, have a great influence on the ranking. Some research in information retrieval shows that the language model approach is more effective than TF-IDF [25].

In this model, a multinomial probability distribution over words in the vocabulary is used to represent a candidate user. A new question is represented by a set of words q= $\{w_1, w_2, ..., w_{\mathrm{N}}\}$ where w_{i} is a non-stop word and each question is assumed to be generated independently. Therefore, the probability of a question being generated by a candidate user can be computed by taking the product of each word's probability in the question given the user profile.

$$P(q|u) = \prod_w P(w|\theta_{\mathrm{u}})^{n(w,q)} \tag{4}$$

where θ_{u} denotes user profile for user u, $P(w|\theta_{\mathrm{u}})$ is the probability of generating word w from user profile θ_{u} and $n(w,q)$ is the number of times word w appears in question q. Since many words in the vocabulary will not appear in a given user profile, and $P(w|\theta_{\mathrm{u}})$ will be zero for such w, we need to use a smoothing method on the $P(w|\theta_{\mathrm{u}})$). By doing so, we can avoid zero probability for unseen words [29]. We apply Dirichlet smoothing method for $P(w|\theta_{\mathrm{u}})$:

$$P_{LM}(w|\theta_{\mathrm{u}}) = \lambda P(w|\theta_{\mathrm{u}}) + (1-\lambda)P(w) \tag{5}$$

where $P(w)$ denotes the background language model built on the entire collection Q and $\lambda \in$ [0,1] is a coefficient to control the influence of the background model and is defined as:

$$\lambda = \frac{\sum_{w \in \theta_{\mathrm{u}}} tf(w,\theta_{\mathrm{u}})}{\sum_{w \in \theta_{\mathrm{u}}} tf(w,\theta_{\mathrm{u}}) + \mu} \tag{6}$$

where $tf(w,\theta_{\mathrm{u}})$ is the frequency of w in the profile of u and parameter μ is set to 1000 in the experiments.

The background model $P(w)$ can be computed through a maximum likelihood estimation:

$$P(w) = \frac{n(w,Q)}{|Q|} \tag{7}$$

where $n(w,Q)$ denotes the frequency of words w being in the collection Q and $|Q|$ is the total number of words in the collection.

4.3 Latent Dirichlet Allocation

While TF-IDF and language model have some interesting features, they still provide a relatively small reduction of dimensionality and do not model much of the inter or intra document structure.

LDA is a three-level hierarchical Bayesian model and has been used extensively for modeling text corpora. Each document in a collection is modeled as a mixture over a set of topics where each topic is a distribution over words in a given vocabulary [5].

To apply LDA on the problem of expert finding, we have to model user profiles as a mixture of topics. A question is represented as a vector of N_{w} words where each w_{i} is chosen from a vocabulary of size V. A collection of user profiles is denoted as D =$\{(w_1, u_1), ..., (w_{\mathrm{m}}, u_{\mathrm{n}})\}$ where m is the total number of words in the vocabulary and n is the number of users in the corpus.

The LDA model assumes a certain generative process for data. To generate a user profile, LDA assumes that for each user profile a distribution over topics is sampled from a Dirichlet distribution. In the next step, for each word in the

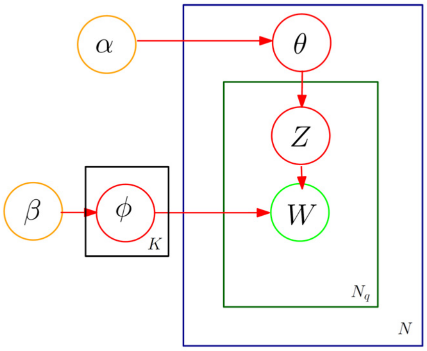

Figure 1: Generative model for documents in LDA

user profile, a single topic is chosen according to this topic distribution. Finally, each word is sampled from a multinomial distribution over words specific to the sampled topic. The generative process of the LDA model is shown in Fig. 1. The only observed random variable in this model is W and the rest are latent variables. A box around groups of random variables indicates replication.
In this model, θ is a matrix of user-profile probabilities for K topics drawn independently from a symmetric Dirichlet α prior. ϕ denotes a matrix of topic probabilities for all the words in the vocabulary drawn from a symmetric Dirichlet β prior. z is the topic assigned to word w from θ distribution, where w is drawn from the topic distribution corresponding to z. The process of generating the profile of user u is as follows:

1. Choose a topic $k \in \{1, ..., K\}$ from the θ distribution.
2. Pick a word w from the multinomial distribution ϕ_{k}.
3. Repeat the process for N_{w} times where N_{w} is the total number of words in user profile.

LDA assumes that this process is repeated for generating user profiles for every user in the dataset. We use Gibbs sampling [12] for estimating parameters of the model.

4.4 Segmented Topic Model

LDA is informative about the content of user profiles, but it does not take advantage of the structure of profiles. Each profile is composed of questions where each question contains sentences. The shared topics between questions can be extracted from the structure of profiles.
Segmented Topic Model (STM) introduced by Lan Du et al. [10] is a topic model that discovers the hierarchical structure of topics by using the two-parameter Poisson Dirichlet process [24]. A four-level probabilistic model, STM contains two levels of topic proportions. Instead of grouping all the questions of a user under a single topic distribution, it allows each question to have a different and separate distribution over the topics. This can lead to more realistic modeling of expertise.
In the STM model, words are the basic element of the data represented by 1,...,W. Each question q is considered as a segment that contains $N_{\mathrm{q,w}}$ words. A user profile is considered a document that contains questions (segments). A corpus is a collection of profiles. The complete list of notations is shown in Table 1.

Table 1: list of notations

Notation	Description
K	Number of topics
U	Number of profiles
N_{q}	Number of questions in profile u
$N_{\mathrm{q,w}}$	Number of words in question q, profile u
V	Size of vocabulary
α	Prior distribution for profile topic distribution
μ_{u}	Profile topic probabilities for profile u
$\nu_{\mathrm{u,q}}$	Question topic probabilities for user u, question q
ϕ	Words probability matrix
γ	Dirichlet prior for ϕ
$w_{u,q,v}$	Word in profile u, question q, at position v
$z_{u,q,v}$	Topic for word in profile u, question q, at position v

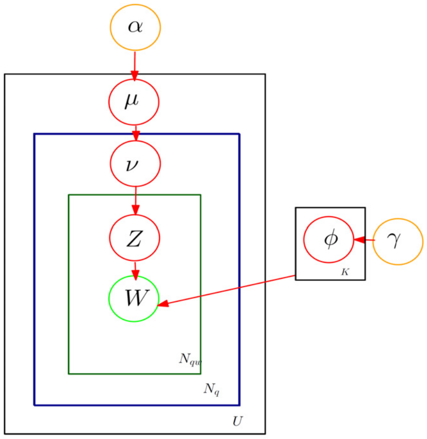

Figure 2: Generative model for documents in STM

Each profile u is a mixture of latent topics denoted by probability vector μ_{u}; each question is also a mixture on the same space of latent topics and is drawn from a probability vector $\nu_{\mathrm{u,q}}$ for question q of profile u. The expertise set of a user, the main topics of each question in the profile and the correlation between each profile and its questions are modeled by these distributions over topics μ and ν.
The generative process of STM for a profile u is as follows:

1. Pick $\mu_{\mathrm{u}} \sim$ Dirichlet(α).
2. For each question q draw $\nu_{\mathrm{u,q}}\sim$ PDP(a,b,μ_{u}).
3. For each word $w_{u,q,v}$ choose a topic from $z_{u,q,v} \sim$ discrete ($\nu_{\mathrm{u,q}}$).
4. Select a word from $w_{u,q,v}\sim$ discrete($\phi_{z_{u,q,v}}$).

The graphical representation of STM is shown in Fig.2. The number of topics is assumed to be given. The only observed random variable in this model is W.

5. EXPERIMENTAL STUDY

The experimental dataset is based on a snapshot of the community based question answering site Stackoverflow[1]. It features questions and answers on a wide range of topics in computer programming.

[1] http://stackoverflow.com/

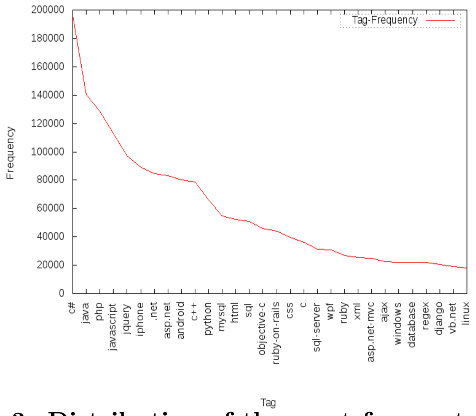

Figure 3: Distribution of the most frequent tags in Stackoverflow

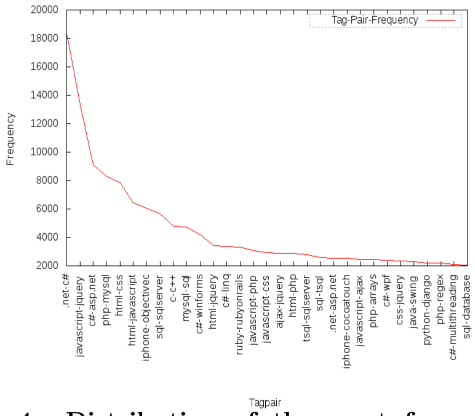

Figure 4: Distribution of the most frequent co-occurring tags in Stackoverflow

5.1 Stackoverflow

The purpose of this section is to provide readers with the necessary background to understand the main characteristics and the question answering mechanism of the Stackoverflow website.

Stackoverflow is a question answering service organized with a user-defined taxonomy of topics. Specifically, questions and answers are posted within categories. The writer of a question should specify the category of the question by assigning a keyword or tag for it. There are approximately three thousand different tags in Stackoverflow. The categories cover a range of different topics in computer programming and attract users from a wide variety of fields. Stackoverflow participants can thus save time in their quest for information because they can get an answer relatively quickly or find what they are looking for among the existing questions and answers.

Questions are the central elements of Stackoverflow. The life cycle of a question starts in an open state where it receives several answers. Then, at some point, the question is considered closed and cannot receive more answers. At this stage a best answer is selected either by the user who posted the question or by other users via a voting procedure. The question will be considered closed once a best answer is chosen.

Note that Stackoverflow participants do not limit their activity to asking and answering questions. They are also allowed to participate in regulating the system by voting and editing questions and answers. Users can mark interesting questions and evaluate the answers by voting for the best answers. Stackoverflow presents additional data, i.e. each user has reputation which shows how much the community trusts the user and badge, which shows how active the user is. This type of information can be used as ground truth for performance evaluation.

5.2 Dataset

To conduct experiments, we select a representative subset of the dataset. Tags are the only elements that categorize different topics. However, they belong to a very diverse topic set. Therefore, we need to create a subset of the dataset that exhibits the same properties as in the original one. Frequency of the top 2000 tags from posts with more than 2 answers is shown in Fig. 3. We use this diagram to extract the most frequent tag. The pair-wise frequency of the top 2000 tags from posts with more than 2 answers is presented in Fig. 4. By using these two diagrams, we examined tag frequency and tag co-occurrence statistics, and manually selected a total of 21 tags. This subset was chosen such that a similar tag distribution as the original data collection was maintained. Selected tags mostly belong to three categories: (i) tags that are highly frequent and mostly co-occur with other tags, (ii) tags that are frequent but never co-occur with other tags, and (iii) tags that sometimes co-occur with other tags. These tags are shown in Table 2.

Stackoverflow website is crawled and 118510 resolved questions and answers between Jan 2008 and Jan 2009, tagged with one or more of 21 selected tags are picked. This dataset is publicly available for research purposes[2].

Questions are resolved, thus each question has a best answer corresponding to one best answerer. Some statistics for this dataset is shown in Table 3. Numbers in parentheses are related to candidate best answerers. For the best answerer prediction, those users who have answered at least N best answers are considered (in this work N=20). As seen in Table 3, in the selected dataset 22027 users have given at least one best answer, while only 1845 of users wrote at least 20 best answers. Those 1845 users are very important to the question answering community. They constitute 0.5% of all users, but have answered 35% of all answered questions.

All the questions are stemmed and the stop-words have been removed using Mallet toolbox [3] [19]. Words that appear less than 5 times in the corpus are also ignored. All the questions answered by the best answerers between January and February 2009 are extracted to build the test dataset. As a result, there are 5128 test questions in our test dataset. Note that some best answerers give no best answer to questions within this period. Therefore, there is no test question assigned to them in our test dataset. However, they are still candidate users for best answerer prediction for a given test question.

5.3 Evaluation

Evaluation of the quality of the resulting ranked list of best users is a difficult task. Only users who have already answered a particular question are ranked in [26]. This will be useful in choosing the best answer for the question but

[2] http://web.cs.dal.ca/~riahi/

[3] http://mallet.cs.umass.edu/

Table 2: The first 10 most probable words for 10 different topics in STM

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
size	colour	servlet	server	email	game	iphone	os	table	thread
bit	draw	request	client	session	player	app	linux	query	lock
heap	graphic	response	connect	send	point	device	windows	select	run
space	png	message	socket	mail	rotate	application	mac	row	wait
cpu	img	web	port	attach	graph	mac	system	column	synchronize
machine	background	server	send	subject	video	apple	platform	join	start
allocation	jpg	redirect	remote	receive	matrix	sdk	virtual	order	call
usage	rectangle	session	network	queue	scale	opengl	pc	sql	process
limit	height	tomcat	host	address	board	os	machine	result	block
run	circle	http	ip	body	math	video	bit	mysql	sleep

Table 3: The first 10 most probable words for 10 different topics in LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
size	colour	servlet	server	email	game	iphone	os	table	thread
object	graphical	public	file	user	play	app	windows	id	run
memory	insert	string	code	post	develop	data	install	select	lock
class	collect	handle	string	view	program	animation	application	query	object
code	google	id	http	format	card	view	mac	key	wait
error	value	class	content	comment	sound	image	local	row	call
snapshot	jsp	persist	workspace	valid	video	application	framework	group	time
message	file	tag	line	page	audio	object	library	method	synchronize
profile	database	google	ibm	update	code	sdk	process	group	method
row	code	message	test	action	book	video	win	type	queue

Table 4: 21 selected tags for the training set

Frequently co-occur	partially co-occur	Rarely co-occur
C#	Bach	Django
SQL	Python	css
Linux	SQL-server	Ruby
Windows	Delphi	Ruby-on-rails
Java	Web-development	WPF
C	.net	iphone
Homework	Java script	Android

Table 5: Data Statistics. Numbers in parentheses show candidate best answerers for expert prediction

Questions	Askers	Best Answerers
369440(123933)	186027	22027(1845)

will not help in directing a new question to potential best experts. However, in [17], all the users in the corpus are ranked for the given question instead of ranking only those users who have answered the question. In our work, we used the second method and ranked the users according to the four methods mentioned previously. If a model could find the actual best answerer of the questions among the top N predicted users, then, prediction is successful. This method of measuring the quality of ranking is called success at N (S@N).

For S@N, if the best answerer for a test question is among the top N returned users, then the value of S@N is the reciprocal rank of that user, otherwise the S@N value is 0. The value of S@N of all the test questions is the average S@N value of the whole test set. Therefore, a large value for S@N means better performance. In the best case, when the best answerers for all the test questions are ranked number one by a method, S@N will be 1.

In topic models, hyper-parameters could play an important role. For the LDA model, a range of values between 0.01 and 0.05 for parameter α were explored. We also tried different settings of a and b for the STM model and eventually used $a = 0.2$ and $b = 10$ for all the experiments.

The results of S@1 for different number of topics are shown in Fig. 5. Y axis represents the S@1 values and X axis shows number of topics. Number of topics is a parameter of the topic models we have used. Performance for TF-IDF and language model is independent of this parameter and therefore, it stays the same as the number of topics changes for our topic models.

The results of predicting best answerers comparing four different methods are presented in Table 6. Topic models exhibit much better performance compared to the two traditional information retrieval approaches. As we expected, the STM model consistently performs better than the LDA model which indicates that taking advantage of the structure of profiles is important in retrieving the answerers. In general, semantic based methods seem to be more accurate in predicting the best answerer in our corpus.

Some examples of topics extracted from user profiles using the STM and LDA models are shown in Table 4 and Table 5 respectively. The purpose of these two tables is to intuitively demonstrate that the topics extracted using the STM model are superior compared to the ones extracted using the LDA model. Comprehensive user studies are required to verify our intuitive conclusion. The first 5 columns in Table 4 are some examples of topics extracted by the STM model, which the LDA model failed to properly detect. For example, the second column of Table 4 shows the computer graphics topic discovered by the STM model. The corresponding topic discovered by the LDA model is shown in the second column of

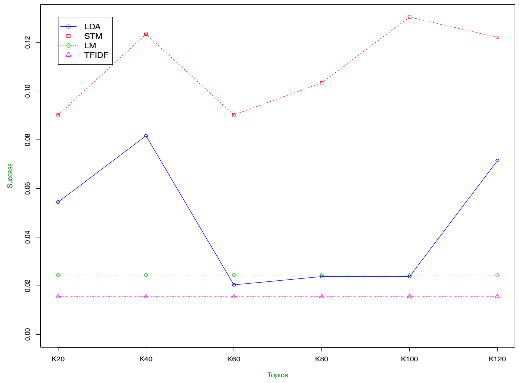

Figure 5: Results of best answerers prediction. Y axis shows S@1 values and X axis represents number of topics

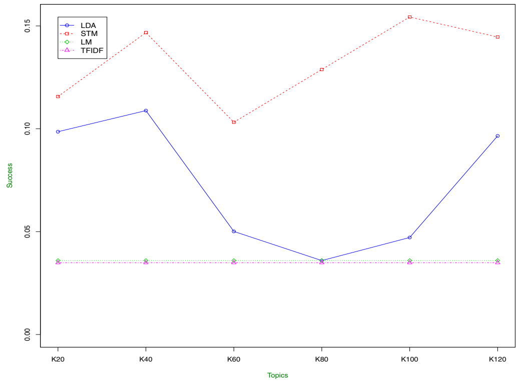

Figure 6: Results of best answerers prediction. Y axis shows S@5 values and X axis represents number of topics

Table 5. Comparison of these two columns suggests that the topic discovered by the STM model is more coherent compared to the corresponding topic discovered by the LDA model. It might seem unfair that we have compared the good quality topics discovered by the STM model against the corresponding topics found by the LDA model. Therefore, the last 5 columns in both tables show topics that were of high quality discovered by the LDA model and their corresponding topics found by the STM model. Comparing these sets of topics indicates that wherever the LDA model is performing well, the STM model can match its performance in terms of topic quality.

Table 6: Results of best answerers prediction for S@N

Method	S@1	S@2	S@3	S@4	S@5
LM	0.0243	0.0304	0.0304	0.0335	0.0359
TF-IDF	0.0155	0.0272	0.0298	0.0317	0.0348
LDA	0.0578	0.0765	0.0810	0.0836	0.0856
STM	0.1034	0.1051	0.1192	0.1200	0.1267

6. CONCLUSION AND FUTURE WORKS

Routing new questions to the right group of experts is an important problem in Community Question Answering systems. A solution to this problem provides users with high quality answers within a reasonable time. It also presents questions to the experts matching their expertise.

For experts in our dataset, we build profiles based on their answering history. These profiles are then used in comparison with a newly posted question. Two statistical topic models are used along with two more traditional approaches. Latent Dirichlet Allocation model, TF-IDF and language model assume that a user profile is a single text unit comprising all questions answered by the user. The Segmented Topic Model, on the other hand, recognizes individual questions as independent units of text. Our results indicate that the LDA model outperforms TF-IDF and language model in retrieving a candidate set of best experts for a question. The STM model performs considerably better than the LDA model, suggesting that the simple structural information used in the model helps produce better results. Our results suggest that statistical topic models can be considered as suitable replacements for more traditional methods in expert recommendation.

Community Question Answering websites produce other types of metadata for the posted question and answers such as score, favourite count and last edit date. Moreover, user information often contains metadata information such as badges and reputation. Using this additional information may help improve the performance of an expert recommendation system. Statistical topic models can be extended to model additional observed variables. Encouraged by the performance improvement for the STM model, we are planning to take advantage of this information in our future work by extending it.

7. REFERENCES

[1] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and Yahoo Answers: everyone knows something. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 665–674. ACM, 2008.

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 43–50. ACM, 2006.

[4] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 192–199. ACM, 2000.

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

[6] R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question

answering from frequently-asked question files: Experiences with the faq finder system. Technical report, AI Magazine, 1996.

[7] Y. Cao, H. Duan, C. yew Lin, Y. Yu, and H. Wuen Hon. Recommending questions using the MDL-based tree cut model. In *Proceeding of the 17th International Conference on World Wide Web, (WWW 08*, pages 81–90. ACM, 2008.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[9] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–48. ACM, 2003.

[10] L. Du, W. Buntine, and H. Jin. A segmented topic model based on the two-parameter poisson-dirichlet process. *Mach. Learn.*, 81:5–19, 2010.

[11] Y. Fu, R. Xiang, Y. Liu, M. Zhang, and S. Ma. A CDD-based formal model for expert finding. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, CIKM '07, pages 881–884. ACM, 2007.

[12] T. L. Griffiths and M. Steyvers. Finding scientific topics. 101:5228–5235, 2004.

[13] 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 Conference on Information and Knowledge Management*, CIKM '08, pages 921–930. ACM, 2008.

[14] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57. ACM, 1999.

[15] J. Jeon, W. B. Croft, and J. H. Lee. Finding semantically similar questions based on their answers. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 617–618. ACM, 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: Proceedings of the 29th Annual International ACM*, pages 228–235. ACM Press, 2006.

[17] M. Liu, Y. Liu, and Q. Yang. Predicting best answerers for new questions in community question answering. In *Proceedings of the 11th International Conference on Web-age Information Management*, pages 127–138. Springer-Verlag, 2010.

[18] X. Liu, W. B. Croft, and M. B. Koll. Finding experts in community-based question-answering services. In *Proceedings of the 14th ACM Conference on Information and Knowledge Management*, pages 315–316. ACM, 2005.

[19] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[20] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD*, pages 500–509, 2007.

[21] A. Mockus and J. D. Herbsleb. Expertise browser: A quantitative approach to identifying expertise. In *Proceedings of International Conference on Software Engineering (ICSE 2002)*, pages 503–512, 2002.

[22] M. S. Pera and Y.-K. Ng. A community question-answering refinement system. In *Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia*, pages 251–260. ACM, 2011.

[23] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. *Int. J. on AI Tools*, 2008.

[24] J. Pitman and M. Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25:855–900, 1997.

[25] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281. ACM, 1998.

[26] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th International Conference on World Wide web*, WWW '09, pages 1229–1230. ACM, 2009.

[27] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press, 2004.

[28] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.

[29] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22:179–214, 2004.