Routing Questions for Collaborative Answering in Community Question Answering

Shuo Chang
Dept. of Computer Science
University of Minnesota
Email: schang@cs.umn.edu

Aditya Pal
IBM Research
Email: apal@us.ibm.com

***Abstract*—Community Question Answering (CQA) service enables its users to exchange knowledge in the form of questions and answers. By allowing the users to contribute knowledge, CQA not only satisfies the question askers but also provides valuable references to other users with similar queries. Due to a large volume of questions, not all questions get fully answered. As a result, it can be useful to route a question to a potential answerer. In this paper, we present a question routing scheme which takes into account the answering, commenting and voting propensities of the users. Unlike prior work which focuses on routing a question to the most desirable expert, we focus on routing it to a group of users - who would be willing to collaborate and provide useful answers to that question. Through empirical evidence, we show that more answers and comments are desirable for improving the lasting value of a question-answer thread. As a result, our focus is on routing a question to a team of compatible users. We propose a recommendation model that takes into account the compatibility, topical expertise and availability of the users. Our experiments over a large real-world dataset shows the effectiveness of our approach over several baseline models.**

INTRODUCTION

Community Question Answering (CQA) services allow their users to ask and answer questions. These services are extremely popular and they attract tens of millions of users daily. Due to such large scale participation, it is a challenge to ensure that all the questions get answered in a timely manner. CQA services like Yahoo! Answer and Stack Overflow organize questions using tags and categories - making it easier for the users to find questions they prefer answering. Additionally, several algorithms have been proposed for automatically routing question to the users. In particular, [1] proposed a generative model to capture the interest of users and [2] proposed an auto-regression model to estimate the availability of users. These approaches then use IR algorithms to find the most appropriate user(s) that would answer a given question.

However, recent work on Stack Overflow and Quora [3] shows that these sites consist of a set of highly dedicated domain experts who aim at satisfying the question askers' query but more importantly at providing answers with high lasting value to a wider audience (similar to Wikipedia). We hypothesize, based on our experience, that valuable question-answer threads are those where several people collaborate together. As a result, a routing strategy that finds a set of compatible users - who would be willing to work together to answer a question - would ensure that the answers are of high lasting value. Unlike prior work, which focuses on finding the most appropriate answerers to a question, we focus on recommending a set of most compatible answerers and commenters to a question.

In this paper, we aim at enhancing question routing algorithms by targeting at improving lasting value of the answers in addition to reducing the response time. We first seek to understand *what are the key factors that lead to lasting value of a question-answer thread?* and then *what routing strategy should we employ to ensure that a question gets answers with lasting value?* We formally define our problem as collaborative question routing - finding a set of users who would collaborate together to provide content with lasting value on a QA thread. To tackle this problem, we propose a framework to capture compatibility, availability and expertise of the users. We then demonstrate how these attributes can be used while recommendation. Our results on Stack Overflow dataset shows that the *timely collaboration among the users leads to improving the lasting value of a QA thread*, thereby validating our hypothesis. We also observe that different types of users have different propensity to answer and comment. As a result, our strategy is to build separate lists of answerers and commenters. We consider comments as a first class citizen of a CQA system; as often times comments critically evaluate an answer leading to clarifications and refinements in the answers - in turn increasing the overall value. In particular, our contributions are following:

- We show how different kinds of participation, especially in the early stage, affect the lasting value of a QA thread and show that question-answering process is a collaborative effort that requires input from different types of users.
- We introduce the concept of user-user compatibility and propose a mechanism to model this concept in CQA services.
- We propose a routing framework that uses compatibility, availability and expertise of the users to recommend answerers and commenters to a question. We extensively evaluate several topic, expertise and availability models in order to build our framework.

RELATED WORK

QA communities have been studied from many perspectives in recent years. A popular perspective is to model the users in the community so as to understand who are contributing [4] and extract their patterns of participation [5].

The goal of this line of work is to identify the expert users in the community. [6], [7] used graph based method to find the important nodes as experts. [8]–[11] approached the problem by extracting expertise features such as question selection bias and applying machine learning algorithms to identify them.

Another perspective is to view the question answering as an information retrieval process with the purpose of satisfying the question asker. [12] used content, structure and community-focused features to predict the satisfaction of question asker and found information seeking patterns that correlate with satisfaction. With the goal to satisfy the question asker, much effort have been spent on the task of question routing. [1] developed a generative model for question answering and discovered the latent interest of users from previous questions and answers to do the answer provider recommendation. [13] combined the interaction graph model with latent topical expertise estimation to recommend incoming questions to top-k list of experts. [2] introduced simple autoregression model to estimate the availability of users in question routing in addition to prior work. [14] used the categorical information of the questions in Yahoo! Answers to improve the question routing. [15] modeled the question routing as a classification problem of whether a user will provide an answer to the question based on various features from questions, users and their relations. [16] implemented the idea of routing questions to the appropriate users in a QA system called "Aardvark". [17] uses collaborative filtering to recommend questions to each user based on the history of users.

Recently, [3] pointed out that the dynamics of the community activity shapes the outcome of answers. The author identified key properties of the community to predict the long lasting value of answers as well as whether a question requires better answers. Our work complements prior work by including the goal of improving lasting value of answers to question routing by routing questions to a collaborative group of users. We are the first to propose to improve the lasting value of question and answering thread as well as the common goal of satisfying question asker, because the QA systems are gradually becoming more important as knowledge reference. Moreover, we propose a framework to route question to compatible group of users.

Stackoverflow Dataset

Stack Overflow is one of the most popular CQA system targeted towards programming related topics. The system is famous for fast response and high archival value of question and answer threads. In this study, we use the public data dump of the site that contains all the public data since its launch in August 2008 to September 2011. Because Stack Overflow is a general site about programming, the activities of users may vary across different topics. We extract two data sets to carry the question routing experiments: one is all the questions, answers, comments and users associated with the tag of "Scala"(a programming language based on JAVA) and the other is a much larger data set with the "Java" tag, which is the second most popular tag on Stack Overflow. The statistics of the two data sets is summarized in Table I. We recognize that voting is also an important activity in Stack Overflow, but the detailed voting data is not available in the dataset.

	Scala	Java
# questions	6,810	181,560
# answers	15,864	445,468
# comments	29,681	769,362
# users	4,902	88,109

TABLE I: Two datasets with tag of "Scala" and "Java" in Stack Overflow

The "Java" data set contains a big portion of questions on Stack Overflow and includes many sub-topics that require different expertise, e.g. experts for "Android" may not be able to answer questions on "tomcat", while the "Scala" data set has more concentrated topics. The combination of the two is representative of the diversity of topics in CQA.

Participation in Question Answering

We begin by exploring how users collaborate to answer a question to not only satisfy the asker but also to provide a knowledge repository with lasting value to wider audience. We explore several different aspects of the question-answering (QA) process - the most important of which is answering. Another important aspect is commenting which often take a critical view of the answers leading to further clarifications and improving the answer quality. One direct proxy to measure the lasting value of QA thread is the number of unique users besides the contributors who viewed that thread. We use the number of views as this is most easily available in CQA datasets. We first take all the questions asked in Stackoverflow during September 1 to September 20, 2009, resulting in a collection of 21,740 questions. These questions were selected as they are all roughly two years old in the data set and hence the page views on these questions must have stabilized (unlike a new question which is yet to receive an audience). We subtract the number of answers and comments from the number of page views because each contribution attributes to one page view. The resulting number is a proxy measure of how valuable a question is to other community members.

Our main hypothesis is that a *timely collaboration between the users is pivotal for its increase in lasting value*, which is our main motivation to route a new question to a set of users who are more likely to collaboratively answer it. And we test our hypothesis on the data set described above with a linear regression model, in which log scaled number of page views is the response variable. We gathered the answers and comments (both on questions and answers) received on the question threads within one hour of posting the question. We constructed three explanatory variables: *number of answers*, *average score (#upvotes − #downvotes) of the answers* and *number of comments* received within the first hour. We use linear regression to learn the weights of all the explanatory variables[1]. The coefficients of the three variables are 0.04, 0.08 and 0.01. All three are positive and statistically significant using one sided t-test ($p \sim 0$). Figure 2 shows the trend of the log scaled number of views over each of the three variables. The result indicates that high quality answering and

[1] We eliminated instances where all feature values were 0 as that would compromise the weight values.

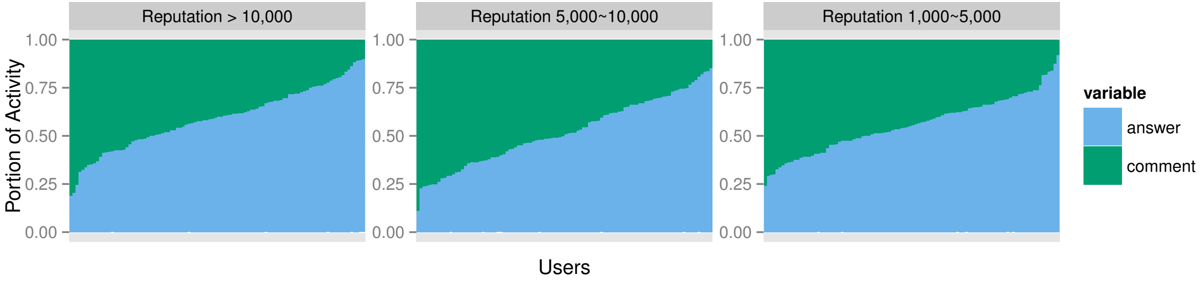

Fig. 1: The breakdown of activities into answering and commenting on posts of users randomly sampled from three reputation levels in Stack Overflow

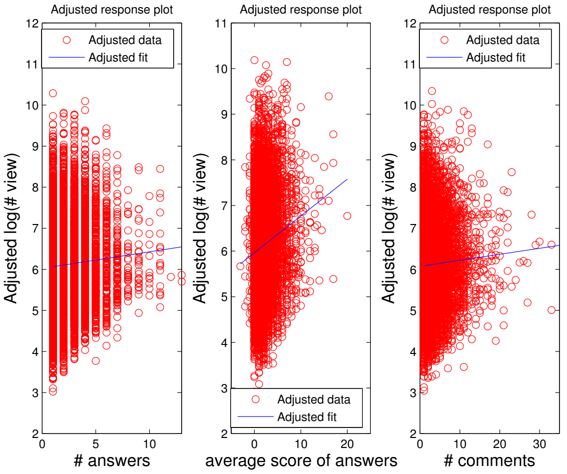

Fig. 2: Effect of number of answers, average score of answers and number of comments on log scaled number of views in linear regression model.

commenting in the early stage greatly improves the lasting value of the question threads in a long run. Therefore, we aim to stimulate this early activity by routing the new questions to a set of experts who would be willing to work collaborative answer and comment.

User Roles

Previous analysis shows that the question answering process requires collaborative effort from different users. Here we explore whether these users are specialized in certain kinds of activity i.e. answering or commenting. To perform this experiment, we first randomly sampled 300 users (100 users each from three reputation levels: $> 10,000$, between 5,000 and 10,000, and between 1,000 and 5,000). Reputation score is a Stack Overflow concept which assesses the value of users' contribution to the community. The three partitions allows us to focus on different kinds of users. For each user, we compute their number of answers. Note that it is common to comment on own answers as a way to defend the correctness of answers. Hence we do not consider such comments. Then we calculate the percentage of each activity that a user performs amongst that users overall activity. Figure 1 shows the activity chart of users. One thing that is common across all the sub-figures is that there is no fixed pattern between the number of answers and number of comments. It establishes that some users prefer to answer and some really like to comment. Overall we see that question-answering is a collaborative effort which results from an effective engagement of users with different specialities.

Collaborative Question Routing Problem

CQA services allow different kinds of users to participate. Some users, especially top contributors, prefer being the first answerer to a question, whereas some of them prefer answering or commenting when some answers have already been provided [10]. For a valuable knowledge exchange, it requires that a group of users collaborate together to answer a question. Based on this hypothesis, we define the collaborative question routing problem more formally as follows:

Problem 1: Given a set of users, $\mathcal{U} = \{u_1, \ldots, u_n\}$ and a question q, find a set of users $U_q \subseteq \mathcal{U}$, such that,

$$U_q = \arg\max_{U} \left\{ \frac{Value(U, q, \epsilon)}{\log |U|} \right\} \quad (1)$$

where $Value(U, q, \epsilon)$ represents the qualitative value of the answers produced by the group of users U on the question q within ϵ time of q's publish time. In Eq. 1, $\log$ scaling ensures that the problem is well-posed otherwise it would yield a degenerate solution $U_q = \mathcal{U}$, i.e., route to all the users.

Our Approach

Let $\mathcal{U} = \{u_1, \ldots, u_n\}$ be all the users, $\mathcal{Q} = \{q_1, \ldots, q_m\}$ be all the questions, and $\mathcal{T} = \{t_1, \ldots, t_l\}$ be all the topics in the QA community. We begin by modeling the basic characteristics of the users, namely,

- $Exp(topic, user)$: Topical expertise of a user.
- $Cmt(topic, user)$: Readiness of a user to comment on a specific topic.
- $Avail(time, user)$: Availability of a user during time interval $[time, time + \epsilon]$.
- $Compat(user_1, user_2)$: Compatibility of $user_1$ to answer (or comment) on a QA thread answered (or commented) by $user_2$.

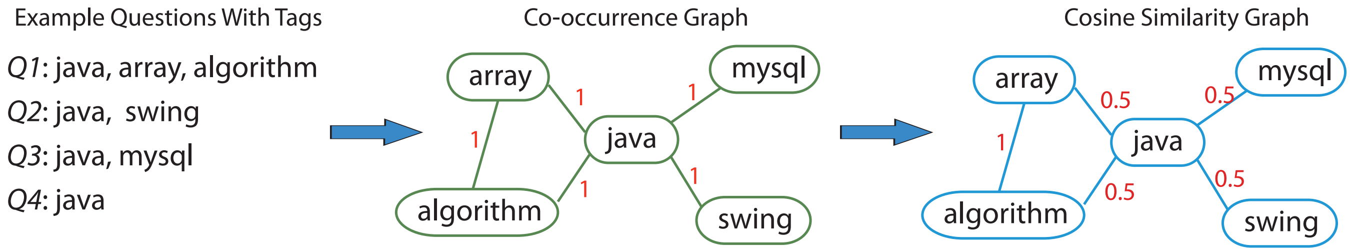

Fig. 3: Construction of tag graph for spectral clustering.

Similarly, for a question we compute its topic distribution $T(topic, question)$. The following subsections present our approach to model the above user characteristics.

TOPIC EXTRACTION

Typically, CQA services allow question askers to annotate their questions with tags. We intuitively hypothesize that tags are a better source for topic modeling over question title and description (text). This is because these tags are curated over time by the community members and a question is annotated with a small number of tags only. To validate this hypothesis, we consider three topic models: (1) Spectral clustering [18] over tag graph, (2) Latent Dirichlet Allocation (LDA) [19] over question tags, and (3) LDA over question tags and text. To construct the tag graph, we compute a tag similarity adjacency matrix. The similarity between two tags is computed based on their cosine similarity.

$$Cosine(tag_1, tag_2) = \frac{\#(tag_1, tag_2)}{\sqrt{\#(tag_1) \cdot \#(tag_2)}} \tag{2}$$

where $\#(tag_1, tag_2)$ is the number of questions containing both the tags and $\#(tag)$ is the number of questions containing tag. Fig. 3 illustrates the construction of tag graph for a toy example. We run spectral clustering [18] on the tag graph using normalized cut criteria. This leads to a partitioning of tags into several clusters, where each cluster is interpreted as a unique topic. LDA is currently the most popular topic modeling algorithm, which considers topics as latent variables and infers the mixing proportion of topics in the documents. Both topic models require the number of topics. We varied the number of topics and checked for two situations: (a) unnecessarily splitting one topic into multiple topics, and (b) mistakenly mixing multiple topics into one topic. Based on manual inspection, we found 30 topics to minimize occurrence of both situations for our dataset. Table II shows the top keywords for the topic labeled `homework`. We find the keywords produced by spectral clustering to be qualitatively more appropriate than LDA. LDA produced keywords that were either irrelevant (e.g. int, print, double) or not strongly related to the topic (e.g. audio, text).

Model	Keywords
SC_{tags}	homework string arrays algorithm arraylist library object datastructure math
LDA_{tags}	homework algorithm data search audio datastructures math text lucene
LDA_{text}	int array homework number arrays system print double loop

TABLE II: Top keywords for the topic `homework` as produced by different topic models. SC_{tags} is spectral clustering model on tags. LDA_{tags} is Latent Dirichlet allocation model on tags, whereas LDA_{text} is LDA on question's tags and text.

ANSWERING AND COMMENTING PROPENSITY

Let $\mathcal{Q}_u$ be the set of questions answered by user u and $\mathcal{Q}'_u$ be the set of questions that user u commented on[2]. We then compute the topical expertise of a user as follows:

$$Exp_n(t, u) = \sum_{q \in \mathcal{Q}_u} T(t, q) \tag{3}$$

where subscript n of Exp indicates that it is based on number of answers and $T(t, q)$ is the probability of q belonging to topic t. This definition of expertise ignores how the community perceives the value of the answers. Therefore we incorporate the score (subscript s) of the answer, i.e. #up votes minus # down votes to reflect the community perceived value as follows:

$$Exp_s(t, u) = \sum_{q \in \mathcal{Q}_u} T(t, q) \quad \sigma(\#\text{up}_q - \#\text{down}_q) \tag{4}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is used as a squashing function for numerical stability, otherwise an answer with very high #up votes would lead to misrepresentation of an user's expertise. Our dataset does not contain voting information for comments, so topical commenting propensity is simply:

$$Cmt(t, u) = \sum_{q \in \mathcal{Q}'_u} T(t, q) \tag{5}$$

ESTIMATING USER AVAILABILITY

To get a prompt reply from a user, one of the necessary requirement is that the user should be available online. We consider two aspects of this problem. The first is *computing the probability that a user would come online on a given day*. The second is *which hour will the user perform an activity*. We model the first problem as a classification problem and compare the performance of several machine learning methods. Given the previous n days of activity information we train a binary classifier to predict whether a user would come online

[2] To simplify our model, we consider a comment on an answer as a comment on the question.

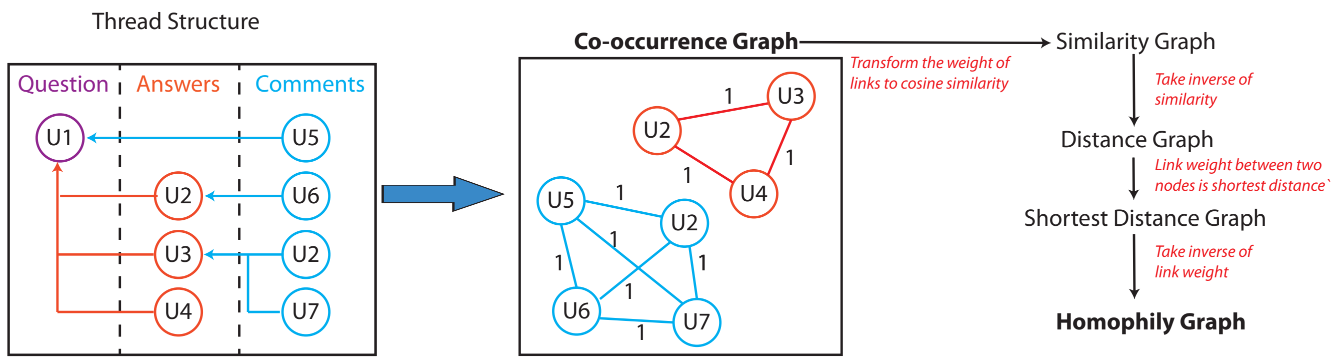

Fig. 4: Illustration of the steps take to compute compatibility between the users.

	Precision	Recall	F-measure
Random Forest	0.3425	0.3652	0.3108
Logit Boost	0.3372	0.3258	0.2757
SVM	0.3481	0.4563	0.3563
Previous Day	**0.3747**	0.4380	0.3749
Always Online	0.2912	**1**	**0.3991**

TABLE III: Performance of different models towards availability estimation over a sample of 5,178 *active* users.

on the next day or not. We tested Random Forest, Boosting, SVM along with two baseline approaches. One baseline is to simply output the last day's status and another is to output always online. We carried a cross validation experiment on a sample of 5,178 *active* users. Note we choose active users due to the importance of their availability prediction result, because recommendations of answerers/commenters are typically drawn from active users. Table III presents the accuracy of different models in predicting user availability. We observe that the machine learning models fail to beat the simple baselines. This is not surprising as the prediction of user activity is a hard problem. Therefore for the purpose of question routing, we assume users to be available everyday.

To estimate the time of the day when a user would be most active, we construct the hourly activity distribution of users. Fig. 5 is the activity distribution of one user, showing that this user is active during evening to night time. We construct daily availability vector for each user.

$$Avail(h,u) = \frac{\text{#answers given during hour } h \text{ by } u}{\text{#answers given by } u} \qquad (6)$$

COMPATIBILITY ESTIMATION

The motivation for routing an incoming question to a group of users is to ensure that the question gets high quality answers and comments. Prior work [3] also suggest that question answering is a collaborative effort. As a result the "chemistry" within the group is an important aspect. Specifically, we need to ensure that the users are compatible with one another and willing to work together. A simple estimation of the compatibility is based on the prior interactions between these users:

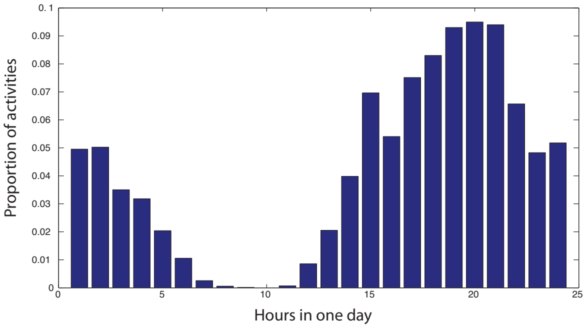

Fig. 5: Hourly activity distribution of a randomly selected user.

users interact frequently though commenting, and answering are likely to interact with each other on the new question.

We propose two graph models $G = (V, E)$ to represent the compatibility of users as depicted in Fig. 4. The first model is based on co-occurrence of users on QA threads: each node represents a user and each edge represents the number of times two users appear together. We construct these graphs separately for answerers and commenters. The second model is the homophily graph which first takes the co-occurrence graph and converts it to a similarity graph. The similarity graph assigns high similarity between users only when they are *equally active* and *collaborate frequently.* Then we construct a shortest distance graph by first converting similarity index to a distance measure and then running the shortest path algorithm on it. Finally, we construct homophily graph by converting distance metric in shortest distance graph back to similarity. The intuition behind homophily graph is that it gives a better estimate of compatibility of two users. Consider two people who have not collaborated in the past but they are similar to a particular user. Similarity graph would yield zero compatibility between these two users, while homophily graph would yield high homophily between the two. Fig. 4 summarizes all the steps taken to compute the graphs.

We obtain homophily graph for answerers and commenters separately. Then answering compatibility $Compat_a(u_1, u_2)$ is the edge weight between u_1 and u_2 in the answerer homophily graph and $Compat_c(u_1, u_2)$ is the edge weight between the two nodes in commenter homophily graph.

		Answerers								Commenters			
Metric	N	$\mathbf{SC_n}$	$\mathbf{SC_s}$	$\mathbf{LDA_n}$	$\mathbf{LDA_s}$	$\mathbf{LDA_{n,tag}}$	$\mathbf{LDA_{s,tag}}$	**B1**	**B2**	$\mathbf{SC_n}$	$\mathbf{LDA_n}$	$\mathbf{LDA_{n,tag}}$	**B**
Average Precision (**P@N**)	5	0.0235	**0.0241**	0.0201	0.0200	0.0217	0.0224	0.0222	0.0109	**0.0243**	0.0197	0.0209	0.0219
	10	**0.0167**	0.0166	0.0132	0.0133	0.0137	0.0139	0.0134	0.0059	**0.0187**	0.0134	0.0137	0.0151
	20	**0.0118**	0.0117	0.0094	0.0094	0.0094	0.0091	0.011	0.0033	**0.0124**	0.0104	0.0112	0.0118
	30	**0.0102**	0.0100	0.0087	0.0085	0.0091	0.009	0.0094	0.0028	**0.0095**	0.0084	0.0084	0.0099
Average Recall (**R@N**)	5	0.0732	**0.0749**	0.0614	0.0615	0.0643	0.067	0.0661	0.0359	**0.5433**	0.5345	0.5369	0.5384
	10	**0.1021**	0.1012	0.0780	0.0787	0.0821	0.0825	0.0804	0.0383	**0.5676**	0.5487	0.5494	0.5537
	20	**0.1440**	0.1425	0.1110	0.1103	0.1104	0.1059	0.1300	0.0426	**0.5914**	0.5755	0.5827	0.5855
	30	**0.1850**	0.1823	0.1553	0.1515	0.1614	0.1598	0.1686	0.0531	**0.6064**	0.5917	0.5924	0.6097
Matching Set Count (**MSC**)	5	0.114	**0.1169**	0.0977	0.0975	0.1043	0.1080	0.1071	0.0546	**0.5534**	0.543	0.5458	0.5475
	10	**0.1581**	0.1572	0.1260	0.1270	0.1310	0.1326	0.1291	0.0589	**0.5817**	0.5603	0.5608	0.5659
	20	**0.2175**	0.2153	0.1751	0.1740	0.1744	0.1678	0.2028	0.065	**0.6094**	0.5916	0.5991	0.6031
	30	**0.2744**	0.2705	0.2373	0.2325	0.2447	0.2423	0.2545	0.0807	**0.6266**	0.6098	0.6107	0.6301

TABLE IV: Comparison between different topic models. SC is Spectral clustering based topic model. LDA is Latent Dirichlet Allocation based state-of-art topic model. $B1$ is Z-score based current state-of-art expertise model. $B2$ is industry adopted reputation model (employed by Stackoverflow.com). Subscript n means expertise is computed using count of topical answers. Subscript s indicates that the expertise is computed by a weighted count of answers, weighted based on the votes on them.

		Answerers				Commenters			
	N	$\mathbf{A1}$	$\mathbf{A2}$	$\mathbf{A3}$	$\mathbf{A4}$	$\mathbf{A1}$	$\mathbf{A2}$	$\mathbf{A3}$	$\mathbf{A4}$
P@N	5	0.0242	**0.0253**	0.0101	0.0241	0.0247	**0.0274**	0.0132	0.0243
	10	0.0166	**0.0179**	0.0057	0.0166	0.019	**0.0195**	0.0088	0.0187
	20	0.0117	**0.0126**	0.0047	0.0117	0.0125	**0.0131**	0.0054	0.0125
	30	0.0101	**0.0105**	0.0036	0.01	0.0096	**0.0101**	0.0042	0.0095
R@N	5	0.0756	**0.0791**	0.0337	0.0749	0.5437	**0.5478**	0.5218	0.5433
	10	0.1017	**0.1091**	0.0373	0.1013	0.5687	**0.5701**	0.5302	0.5676
	20	0.1431	**0.1521**	0.0585	0.1426	0.5918	**0.5963**	0.538	0.5915
	30	0.1837	**0.1881**	0.0673	0.1826	0.6072	**0.6123**	0.5455	0.6066
MSC	5	0.1177	**0.1223**	0.0501	0.1169	0.5539	**0.5585**	0.5267	0.5534
	10	0.1577	**0.1688**	0.0563	0.1573	0.5831	**0.5861**	0.5374	0.5817
	20	0.2161	**0.2305**	0.0924	0.2154	0.6099	**0.6152**	0.5469	0.6095
	30	0.2724	**0.2808**	0.106	0.2708	0.6275	**0.6337**	0.5558	0.6267

TABLE V: Comparison between different availability models. $A1$ uses σ scaling, $A2$ is based on Eq. 10 (or Eq. 11). $A3$ and $A4$ are both linear weighted sum model ($exp\text{-}match(q,u) + w \cdot Avail(tm,u)$ with $w = 1, 0.01$, respectively.

EXPERIMENTS

In order to evaluate different models, we consider precision (Precision@N or simply P@N) and recall at position N (Recall@N or simply R@N), which are widely used measures in the IR community. Let R_q be the recommendations to a question q ($|R| = N$) and U_q be the actual set of users, then $P@N = \frac{1}{|Q|}\sum_{q\in Q}\frac{|R_q \cap U_q|}{|R_q|}$ and $R@N = \frac{1}{|Q|}\sum_{q\in Q}\frac{|R_q \cap U_q|}{|U_q|}$, where Q is the set of questions used for evaluation. We define another accuracy measure, Matching Set Count (MSC) as

$$MSC = \frac{1}{|Q|}\sum_{q\in Q}\mathbf{1}[R_q \cap U_q \neq \emptyset] \tag{7}$$

where $\mathbf{1}[cond]$ is an indicator random variable which equals 1 if $cond$ is true, otherwise 0.

To perform our experiments we create a hold out set from our dataset. Then the performance of the routing algorithm is computed by comparing the recommended set of answerers and commenters with the actual set of users who gave answers and comments. It is a commonly used method to test the routing models [2], [13], [14], [15]. We split the Java data set into training and testing parts based on time: the most recent 60 days data (from August 2011 to September 2011) is used for testing and the data before that period is used for training the routing algorithm. The training dataset consists of 161,996 questions asked by 52,886 askers, 406,378 answers given by 32,555 answerers and 685,025 comments provided by 31,958 commenters. The test dataset consists of 14,845 questions. Next, we test the performance of each building block of the question routing algorithm. The performance of the algorithm is consistent across different datasets, so for space considerations we show the performance over the largest dataset (Java).

Performance of Topical Models

In this section, we compare the performance of several different topic models for collaborative question routing. Using different topical models, we compute the topical expertise $Exp(t,u)$ and commenting propensity $Cmt(t,u)$. An incoming question is then routed to a set of answerers based on expertise match ($exp\text{-}match$) and to a set of commenters based on comment-match ($cmt\text{-}match$). The two matches are defined as follows:

$$exp\text{-}match(q,u) = \sum_{t\in\mathcal{T}} Exp(t,u)\cdot T(t,q) \tag{8}$$

$$cmt\text{-}match(q,u) = \sum_{t\in\mathcal{T}} Cmt(t,u)\cdot T(t,q) \tag{9}$$

We pick top N users with highest match values. Table IV list the performance of the different topic models. The result indicates that the spectral clustering model outperforms the LDA model for answerer as well as commenter recommendation. SC is significantly better than LDA using one-sided *t-test* ($p \sim 0$). The number of answers based model (Eq. (3)) is better than the score of answers based model (Eq. (4)) for all values of N. One reason for this observation is that the answer scores overly skew the expertise distribution of some users for their few answers that received very high votes. As a result, they appear in the recommendation result where they should not.

Table IV also shows the performance of two baseline models: (a) Z-score based prior state-of-art model ($B1$), (b) Stack Overflow's reputation model ($B2$). The Z-score based model is a special case of our topical expertise model in which the concept of sub-topics is not considered. Since our experiments are conducted over Java dataset, a user with lots of answers would intuitively appear to be an experts on Java. However our results indicate that considering sub-topics within the topic (SC_n) is significantly more effective over mechanisms that ignore the concept of sub-topics ($B1$). Stack Overflow's reputation based model is an industry standard which is used by Stack Overflow and its sister sites to gauge the activity levels of the users. Our results indicate that it is not at all desirable for question routing.

Overall, we see that the spectral clustering is more effective than the state-of-art LDA topic model. We also note that modeling expertise in sub-topics is more effective than ignoring the concept of sub-topics. Due to superior performance of spectral clustering, we use it for subsequent analysis.

Performance over User Availability

Here we explore the effect of incorporating user availability along with their expertise for routing. Since most questions on Stack Overflow get answered within a short time frame (a median 21 minutes[3]), we compute the availability $Avail(tm, u)$ of a user within 2 hours (setting $\epsilon = 2$) of the posting of question. To incorporate the availability scores, we directly multiply them by the user's expertise score (and/or commenting propensity). This has a nice probabilistic interpretation that the joint probability of a user's likelihood to answer/comment factors into that user's expertise/commenting probability and availability probability.

$$S_a(q, u, tm) = Avail(tm, u) \cdot \mathit{exp\text{-}match}(q, u) \qquad (10)$$

$$S_c(q, u, tm) = Avail(tm, u) \cdot \mathit{cmt\text{-}match}(q, u) \qquad (11)$$

Apart from this, we experiment with a weighted linear sum of the two scores . Weighted linear sum nicely gels with the regression framework which allows us to learn the optimal weight parameters. We also consider a sigmoid scaling mechanism to combine availability with user expertise, i.e., $S_a(q, u, tm) = \sigma\{Avail(tm, u)\} \cdot \mathit{exp\text{-}match}(q, u)$.

Table V shows the performance of the four availability models combined with the spectral clustering based topic model. The result shows that the $A2$ model (based on Eq. 10 and 11) significantly outperforms all the other models for all N. It is also significantly better than the models without availability (Table IV). This conforms our hypothesis that user availability can be critical for question routing.

Performance over Compatibility

```
**Result**: Produce recommendation $R_a$ and $R_c$
**initialization**: $R_a, R_c = \emptyset$;
**while** $|R_a| < N_a$ **do**
    **for** $u$ in $U \setminus R_a$ **do**
        Compute compatibility $Compat_a(u, R_a)$
        $V(u) = Compat_a(u, R_a) \cdot S_a(q, u, tm)$
    **end**
    Find best answerer $u^\star = \arg\max_{u \in U}\{V(u)\}$
    Add user to recommendation set $R_a = \{R_a \cup u^\star\}$
    Remove user $U = U \setminus u^\star$
**end**
**while** $|R_c| < N_c$ **do**
    **for** $u$ in $U \setminus R_c$ **do**
        Compute compatibility $Compat_c(u, R_c)$
        $V_c(u) = Compat_c(u, R_c) \cdot S_c(q, u, tm)$
    **end**
    Find best commenter $u^\star = \arg\max_{u \in U}\{V_c(u)\}$
    Add user to recommendation set $R_c = \{R_c \cup u^\star\}$
    Remove user $U = U \setminus u^\star$
**end**
```

Algorithm 1: Recommendation algorithm to generate a list of answerers and commenters to a question based on their compatibility, availability and expertise.

Here we combine the compatibility of the users along with their topical expertise and availability. The pairwise compatibility $Compat(u_1, u_2)$ is computed between the users based on their co-occurrence or homophily graph. In order to compute the compatibility of a user with a group of users $Compat(u, U)$, we consider the average compatibility between all possible pairs, as follows

$$Compat(u, U) = \begin{cases} 1 & U = \emptyset \\ \frac{1}{|U|} \sum_{v \in U} Compat(u, v) & otherwise \end{cases} \qquad (12)$$

To build a recommendation set, we employ a greedy algorithm (see Algorithm 1). In the first phase of the algorithm, it builds a set of answerers R_a that would be compatible with one other. In the next phase it builds the set of commenters R_c that are likely to comment over the answers provided by users in set R_a. To build these sets, the algorithm picks a new available user with high expertise and high compatibility with the existing set of users. This is done until the specified number of answerers and commenters is obtained.

We compare the two compatibility graph models based on the previous answering and commenting interactions. For the co-occurrence graph model, we apply sigmoid function to scale the compatibility $Compat(u_1, u_2)$ in the range of 0.5 to 1 so that the compatibility has probabilistic interpretation. The compatibility in homophily graph model is already normalized. The compatibility is combined with topical model and availability model as shown in Algorithm 1.

Table VI shows the performance of different models. It also shows the performance of the best model that runs without compatibility ($A2$ in Table V). We observe that compatibility model $C2$ (based on homophily graph) performs better than the best model that ignores compatibility for answerers. This improvement is statistically significant for $N >= 10$ using one-sided $t-test$ ($p \sim 0$). For commenters the compatibility model performs similar to the $A2$ model. One possible explanation of this observation is that the compatibility model narrows the recommendation to a closely connected set of users. If the first user (greedy choice) is not correct then the recommendation set is off the mark, whereas this is not the case for the model that

[3] http://meta.stackoverflow.com/questions/61301/statistics-what-is-the-average-response-time-on-stackoverflow

		Answerers			**Commenters**		
	N	$\mathbf{C1}$	$\mathbf{C2}$	$\mathbf{A2}$	$\mathbf{C1}$	$\mathbf{C2}$	$\mathbf{A2}$
P@N	5	0.0251	0.0241	**0.0253**	0.027	0.0235	**0.0274**
	10	0.0176	**0.0185**	0.0179	0.0193	0.0188	**0.0195**
	20	0.0121	**0.0128**	0.0126	0.0131	**0.0133**	0.0131
	30	0.0102	**0.0108**	0.0105	0.01	0.01	**0.0101**
R@N	5	0.0781	0.0741	**0.0791**	0.5471	0.5418	**0.5478**
	10	0.1064	**0.1132**	0.1091	0.5689	0.5672	**0.5701**
	20	0.1455	**0.1549**	0.1521	0.5956	**0.5976**	0.5963
	30	0.1836	**0.195**	0.1881	0.6111	0.6117	**0.6123**
MSC	5	0.1214	0.1158	**0.1223**	0.5579	0.5511	**0.5585**
	10	0.1656	**0.1735**	0.1688	0.5848	0.582	**0.5861**
	20	0.2223	**0.2329**	0.2305	0.6148	**0.6164**	0.6152
	30	0.2746	**0.2886**	0.2808	0.6329	0.6332	**0.6337**

TABLE VI: Performance of the compatibility based model for question routing. C1 uses co-occurrence graph based compatibility model and C2 user homophily graph based compatibility model.

ignores compatibility. This effect is mitigated to certain extent with the increase in value of N. Overall the result shows that the compatibility model is useful for recommending answerers.

Conclusion

In this paper, we propose a question routing framework that considers compatibility, availability and expertise of users. We hypothesize that question answering requires collaborative effort and we validate this hypothesis by performing experiments on several specialized and large datasets from the Stack Overflow domain. Our experiments show that indeed a model that takes user's compatibility into account performs better at recommending answerers over compatibility agnostic models. At the heart of compatibility computation is the concept of user homophily which provides a preference of collaboration of one user with another. We see that homophily graph as proposed in this paper works better than a naive computation of user's compatibility through their co-occurrence graph.

Another aspect of our model is computation of topics and topical expertise. Even though we selected a specific topic (e.g. Java, Scala) from the Stack Overflow dataset, yet our results highlights that it is crucial to model the concept of sub-topics otherwise the routing strategies won't be effective. This is because the topic Java contains several specialized sub-topics.If we do not consider these sub-topics, the recommendation model would only choose experts who are knowledgable in general but not necessarily capable of answer questions in certain sub-topics.

Overall our experiments over several different subtopics indicate the general applicability of our models in the technical CQA. We wish to experiment our models over other datasets such as `Yahoo! Answers` and `Quora`. We mark this as our future work. Additionally, we wish to explore different recommendation strategies apart from the greedy algorithm proposed in the paper.

Acknowledgment

We would like to thank Max Harper and other members from GroupLens Research for helpful discussions. We thank the ADC Graduate Fellowship and NSF-IIS-1210863 grant for the support of this research.

References

[1] J. Guo, S. Xu, S. Bao, and Y. Yu, "Tapping on the potential of Q&A community by recommending answer providers," in *CIKM*, 2008.

[2] B. Li and I. King, "Routing questions to appropriate answerers in community question answering services," in *CIKM*, 2010.

[3] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, and J. Leskovec, "Discovering value from community activity on focused question answering sites: a case study of stack overflow," in *KDD*, 2012.

[4] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: everyone knows something," in *WWW*, 2008.

[5] Q. Liu and E. Agichtein, "Modeling answerer behavior in collaborative question answering systems," in *ECIR*, Berlin, Heidelberg, 2011.

[6] P. Jurczyk, "Discovering authorities in question answer communities using link analysis," in *CIKM*, 2007.

[7] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities," in *WWW*, 2007.

[8] A. Pal and J. A. Konstan, "Expert identification in community question answering: exploring question selection bias," in *CIKM*, 2010.

[9] A. Pal, R. Farzan, J. A. Konstan, and R. E. Kraut, "Early detection of potential experts in question answering communities," in *UMAP*, 2011, pp. 231–242.

[10] A. Pal, F. M. Harper, and J. A. Konstan, "Exploring question selection bias to identify experts and potential experts in community question answering," *ACM Transactions on Information Systems*, vol. 30, no. 2, p. 10, 2012.

[11] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha, "Learning to recognize reliable users and content in social media with coupled mutual reinforcement," in *WWW*, 2009.

[12] Y. Liu, J. Bian, and E. Agichtein, "Predicting information seeker satisfaction in community question answering," in *SIGIR*, 2008.

[13] Y. Zhou, G. Cong, B. Cui, C. S. Jensen, and J. Yao, "Routing questions to the right users in online communities," in *ICDE*, 2009.

[14] B. Li, I. King, and M. R. Lyu, "Question routing in community question answering: putting category in its place," in *CIKM*, 2011.

[15] T. C. Zhou, M. R. Lyu, and I. King, "A classification-based approach to question routing in community question answering," in *WWW '12 Companion*, New York, NY, USA, 2012.

[16] D. Horowitz and S. D. Kamvar, "The anatomy of a large-scale social search engine," in *Proceedings of the 19th international conference on World wide web*, ser. WWW, 2010.

[17] G. Dror, Y. Koren, Y. Maarek, and I. Szpektor, "I want to answer; who has a question?: Yahoo! answers recommender system," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD, 2011.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*. MIT Press, 2001, pp. 849–856.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.