SmartQ: A Question and Answer System for Supplying High-Quality and Trustworthy Answers

Yuhua Lin and Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, South Carolina 29631
{yuhual, shenh}@clemson.edu

***Abstract*—Question and Answer (Q&A) systems aggregate the collected intelligence of all users to provide satisfying answers for questions. A well-developed Q&A system should incorporate features such as high question response rate, high answer quality, a spam-free environment for users. Previous works use reputation systems to achieve the goals. However, these reputation systems evaluate a user with an overall rating for all questions the user has answered regardless of the question categories, thus the reputation score does not accurately reflect the user's ability to answer a question in a specific category. We propose SmartQ: a reputation based Q&A System. SmartQ employs a category and theme based reputation management system to evaluate users' willingness and capability to answer various kinds of questions. The reputation system facilitates the forwarding of a question to favorable experts, which improves the question response rate and answer quality. Also, SmartQ incorporates a lightweight spammer detection method to identify potential spammers. Our trace-driven simulation on PeerSim demonstrates the effectiveness of SmartQ in providing a good user experience. We then develop a real application of SmartQ and deploy it for use in a student group in Clemson University. The user feedback shows that SmartQ can provide high-quality answers for users in a community.**

***Index Terms*—Question and answer system; spammer detection; reputation system; Real application; question category**

I. Introduction

Question and Answer (Q&A) systems aim to provide collaborative answering of questions by spreading messages to a group of people with registered interest in the question topic. These systems are becoming popular as they aggregate the collected contributions and assessments of all users. In Q&A systems, askers pose questions and other users answer them. Users' participation is typically motivated by various mechanisms (*e.g.,* earning points or monetary rewards). For example, in Yahoo! Answers (YA), an answerer will receive 2 points for answering a question and 10 points if his answer is selected as the best answer [1]. Social networking is also a motivation for answering in Q&A systems. The study in [2] shows that a knowledge-oriented online social network (OSN) with unidirectional links is formed in YA. If user A wants to frequently visit/track all questions and answers of user B, A adds B to its contact list by building a link to B. Then, A becomes B's fan. So every user has a contact list and fan list.

Q&A systems have significantly changed the way we seek information. When compared with traditional web search engines, Q&A systems tend to provide answers to a broader range of questions attributed to everyday life situations [3]. For example, users may use Q&A systems to ask for quick hotel suggestions, or advice on their college selection from users with relevant knowledge. There are four important issues affecting the performance of a Q&A system:

- **Response rate** The questions launched by askers need to be forwarded to the potential answerers who are willing to provide help. Otherwise, the askers will suffer a long delay before receiving satisfying answers.
- **Answer quality** The objective of a Q&A system is to return high quality answers to the questions, thus, identifying potential experts is crucial before forwarding the questions.
- **Spammer detection** The Q&A system should be able to identify potential spammers and prevent them from spreading trash information.

The first issue with Q&A systems is the answering rate. Answerers in the OSN are willing to and able to provide more tailored and personal answers to the questioners since they are familiar with the questioners [4]. However, there are users who do not bother to give any response to the questions they receive (lazy users). Thus, it is a common case that users will not receive answers for their questions, or suffer from long delay before they receive answers. This is normally due to lack of incentives for answering questions. Analysis on Mahalo [5], a fee-based Q&A site, shows that askers are ready to pay when requesting facts that they are interested in. However, monetary reward is not practical in most free Q&A systems, and a feasible way is to filter out lazy users when choosing answerers.

The second key issue in a Q&A system is answer quality. Users want to get satisfying answers to their questions, however, it is difficult to match a question to a user who has the expertise to answer it. Also, experts may not be willing to provide answers as it occupies their free time. Nam *et al.* [6] showed that altruism, business motives, learning, hobbies, and reputation score are important incentives in Q&A systems. And monetary rewards are effective incentives to improve answer quality to various user questions [7]. However, monetary rewards can only promote the quality of answers when the answerers have expertise in the related field. Forwarding questions to the right experts is the key solution to increase the answer quality.

The third issue with Q&A systems is the detection of spammers. As large Q&A systems are exposed freely to huge amounts of users, they provide an ideal environment for spammers to distribute their commercial advertisements, or malicious users to spread trash information. Existing spammer detection methods mainly focus on characterizing spam traffic [8] and network-level spammers' behavior [9, 10]. However, monitoring and analyzing the spamming features on network traffic and user behavior is expensive.

In order to answer the four key issues, we have incorporated two components in our system design: category and theme based reputation management, and a lightweight spammer detection method. 1) We employ a reputation management system to facilitate the forwarding of a question to favorable experts. For each question category and question theme, a user is assigned a reputation score, this reputation score is calculated in such a way that it reflects the user's trustworthiness and willingness to answer questions on a specific category or theme. So forwarding a question to experts with high reputation will increase the probability that the question is replied to with prompt and high quality answers. 2) Based on the rationale from [2] that a linear relationship exists between the number of best answers and the number of all answers for contributing users, we then propose a lightweight spammer detection method to identify potential spammers. This method examines the ratio of best answer count and total number of answers provided by each user (RBA), and users with low RBA will be regarded as spammers. We further improve the precision of spammer detection by studying the number of contacts a user attracts.

The remainder of this paper is arranged as follows. Section II presents an overview of the related work. Section III presents a detailed description of SmartQ. Section IV presents the experimental results on both PeerSim and real application. Section V concludes this paper with remarks on future work.

II. Related work

Recent years have witnessed a rapid rise in prevalence of online Q&A systems [1, 11, 12] in our daily lives. Facebook launched a Q&A application in July, 2010 which facilitates users posting and answering questions through the OSN in order to take advantage of the collective intelligence of their friends. Early works in Q&A system research community focus on analyzing some of the large-scale Q&A sites, such as Yahoo! Answers and Google Answers. Adamic *et al.* [13] showed that in Yahoo! Answers, users share knowledge across different topic categories (*i.e.*, experts in different domains help one another). Interaction among users is highly skewed depending on the question topics, and best answers can be predicted based on reply thread length. In their analysis of Naver KiN, Nam *et al.* [6] studied the user behavior of answerers and found that their level of participation in contributing knowledge is highly skewed, and answerers' participation tends to be intermittent.

Aside from the studies of the basic characteristics of Q&A systems, researchers also study different ways of improving the question answer rate and quality. Various approaches can be grouped into 3 main categories: 1) Using a centralized server to forward questions automatically; 2) Leveraging social networks for knowledge sharing; and 3) Adopting a reputation system to identify reliable answerers. Centralized Q&A systems, such as Aardvark [14] and IM-an-Expert [15], rely on a centralized server to forward questions to appropriate users in the community. However, the centralized server may suffer from a high service request rate and traffic congestion. Social networks are an effective tool for facilitating knowledge sharing [16, 17]. ReferralWeb [18] and Expertise Recommender [19] both exploit the social network within a community to identify a set of experts with regards to the information in need. Also, Shah *et al.* [20] ascribed the success of the Yahoo! Answers to its reward policy, such as the levels and ranks achieved through contributing useful answers to the community, they also concluded that one reason for the failure of Google Answers [11] was its lack of a social component. Some works apply a reputation system to locate credible answerers [21, 22], these systems maintain a general reputation score for every user as an indicator of whether the user reliably provides high quality answers. SmartQ is distinguished from these works in a way that it provides reputation scores for each user according to different question categories and themes, which help to navigate questions to the right experts and improves question response rate and answer quality.

Also, a Q&A system needs to be clean and user-friendly to earn user loyalty, and various methods have been proposed to assess whether a user is contributing relevant and useful information. Pelechrinis *et al.* [23] proposed a collaborative assessment method to identify spammers in a Q&A system, where each user monitors the activity of other users and observes their compliance with predefined cognitive models. Long *et al.* [24] proposed a collaborative filtering method to limit spammer hazards, which calculates the importance score of each user based on his/her relationships to other users. However, these schemes are not robust to the presence of malicious entities, as they do not consider the correctness of the subjective feedback from users. SmartQ incorporates a lightweight spammer detection method to keep the Q&A system clean and prevent the dissemination of useless information.

III. System Design

Q&A systems match askers to answers and facilitate the sharing of knowledge in both synchronous (answerers reply the question in real time) and asynchronous way (answerers do not need to reply the question instantly). Q&A systems have created abundant resources of millions of questions and hundreds of millions of answers, and continue to an effective source of information. This paper proposes a novel reputation system for computing user reputation score as a reflection of the answerer's willingness to reply, as well as the trustworthiness of the response information.

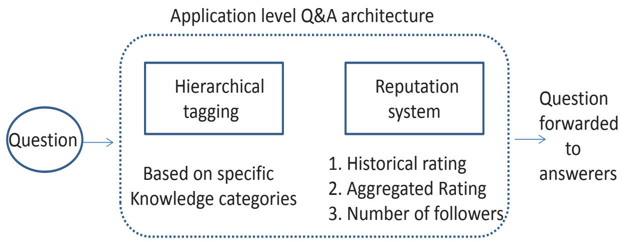


Fig. 1: An overview of distributed Q&A system.

A. An overview of SmartQ

When a user launches a new question, this question is labeled with tags describing the question's category and theme. Then the question is forwarded to its contacts who are registered with interests in the according category and theme. In order to improve the chance of getting satisfying answers, the question should be forwarded to users who are experts in the field of the question, and are willing to answer the question. We assign every user with reputation scores as indications of the user's ability and willingness to answer questions. The reputation scores are estimated by considering three factors. 1)Direct trust, which is calculated by examining the historical interaction records between two users. A user's reputation score is accumulated by serving incoming questions with valuable answers throughout a long time period. We assign different weights to answering activities based on their timestamps. 2)Aggregated trust, which is calculated by gathering the opinions from a user's fans. In large-scale online systems such as Q&A systems, a user only interacts with a subset of all users (*i.e.*, a user's fans) [25], and a user tends to trust its fans' opinions. So a user only aggregates opinions from its fans when calculating another user's reputation. 3)Trustworthiness, which is evaluated by the number of fans a user attracts. The first two factors consider the qualities of answers a user provides, by studying the interaction experience between users. While the third one relies on the fact that user A connects to B only when user A trusts B's knowledge, as A believes that B is capable of answering its questions. The reputation system helps to identify a list of users who are likely to provide high quality answers for each question, then the system forwards the question to a number of potential answerers.

An overview of SmartQ is shown in Figure 1. When user A launches a question at time t, it defines the question's main category and detailed theme in the hierarchical tagging stage. Then we compare the reputation rating of A' contacts, and select a number of contacts with high reputation scores. The question is then forwarded to the highly regarded contacts. The reputation system is responsible for updating users' reputation ratings at a specific frequency.

B. Question Category Selection

Popular Q&A systems can generate a large amount of questions everyday, grouping and organizing the questions by their specific knowledge area is crucial to help users find the questions they are interested in. In SmartQ, we assign two levels of tags to questions: category and theme. Category is a larger domain than theme and every category contains at least one theme. For example, sports, literature and movies are categories a question may belong to; under sports category, there are multiple themes such as soccer, football and basketball. We use q_k to denote a question; c_u be a category; t_{uv} be a theme belonging to category c_u. Thus, a question belonging to category c_u is denoted by $q \in c_u$, and a question belonging to theme t_{uv} is denoted by $q \in t_{uv}$.

C. Category and Theme based Reputation Management of Users

Rating and recommender systems are commonly used in Q&A system to help users evaluate one another's expertise and trustworthiness in answering questions. Given $q_k \in c_u$ and $q_k \in t_{uv}$, user A evaluates user B's reputation on question q_k based on direct trust, aggregated trust and trustworthiness. Direct trust is evaluated based on A's experience; aggregated trust is calculated by gathering opinions from all A's fans; and trustworthiness is measured by examining the number of B's fans.

1) Direct trust: In direct trust, user B's reputation is calculated based on past interactions (*i.e.*, the answers A receives from B). Direct trust is expressed by two factors, category reputation and theme reputation, which are represented by two vectors. $R^c_{ab} = (r^c_{ab1}, r^c_{ab2}, ...r^c_{abn})$ stores the rating of B in different categories, while $R^t_{ab} = (r^t_{ab1}, r^t_{ab2}, ...r^t_{abm})$ represents the rating of B in different themes. Every element in R^c_{ab} and R^t_{ab} is calculated by summarizing the questions belonging to a specific category or theme. In order to reflect a user's recent performance, his recent answering behaviors are assigned with higher weight in reputation calculation. We apply an exponential decay factor $\phi_k \in [0, 1]$ for question q_k, ϕ_k is initialized to 1 and decreases as time elapses. $\phi_k = e^{-\lambda t_k}$, and t_k is the time period that question q_k has been answered. Then, user B's reputation on category u is calculated by:

$$r^c_{abu} = \sum_{q_k \in \mathcal{Q}^u_b} \phi_k \times s_{abk}, \tag{1}$$

where s_{abk} is the rating on question q_k user A gives user B, and $\mathcal{Q}^u_b$ is a set of questions on category u that are answered by user B. The theme reputation r^t_{ab} is calculated in the same way as that in Equation (1), $r^t_{abv} = \sum_{q_k \in \mathcal{Q}^v_b} \phi_k \times s_{abk}$. After A calculates B's category reputation vector R^c_{ab}, and theme reputation vector R^t_{ab}, the two vectors are sent to all A's contacts. Suppose user L is one of A's contacts, R^c_{ab} and R^t_{ab} are needed by user L to calculate user B's aggregated trust value. Note that the information of direct trust does not need to be exchanged on a regular basis, when there are updates of A's direct trust towards another user, A needs to send this update information to all his/her fans. In order to make the exchange of reputation values safe and accurate, SmartQ should withstand some common types of network attacks, such as Man-in-the-Middle attack and data modification. Various approaches such as PKI [26] have been well-developed to prevent these attacks, so this issue is not the focus of our paper. Finally, the direct trust of A towards B regarding question q_k is calculated by:

$$r_{abk} = 1/|\mathcal{T}_u| \sum_{p \in (\mathcal{T}_u \setminus t_{uv})} \Lambda_p \times r^t_{abp} + \gamma \times r^c_{abu}. \quad (2)$$

In Equation (2), $\mathcal{T}_u$ is a set including all question themes under category c_u, and $|\mathcal{T}_u|$ is the number of themes in category c_u. $\gamma \in (0,1)$ is the weight of category reputation, $\Lambda_p \in (0,1)$ is the weight of theme reputation for theme t_{up}.

2) Aggregated trust: In aggregated trust, a user listens to his/her fans' opinions when evaluating another user's reputation. User A receives category reputation vectors and theme reputation vectors of user B from all its fans. The aggregated ratings of B in different categories and themes are stored in vector $r^{c'}_{ab}$ and $r^{t'}_{ab}$, respectively. In the following, for simplicity, we use r^x_{ab} to represent both r^c_{ab} and r^t_{ab}, and use $r^{x'}_{ab}$ to represent both $r^{c'}_{ab}$ and $r^{t'}_{ab}$. When A wants to compute the aggregated reputation of user B on category or theme y, A sums all B's category or theme reputations received from its fans weighted by the closeness between them in Equation (3).

$$r^{x'}_{aby} = \frac{\sum_{d \in F_a} \Theta_{ad} \times r^x_{dby}}{\sum_{d \in F_a} \Theta_{ad}}. \quad (3)$$

Where r^x_{dby} is the direct rating of user D towards user B on category or theme y. F_a is the set containing all user A's fans. Θ_{ad} is the weight of closeness between user A and D, $\Theta_{ad} \in (0,1)$. Similar to Equation (2), the aggregated trust of user A towards B is r'_{ab} calculated by:

$$r'_{abk} = 1/|\mathcal{T}_u| \times \sum_{p \in (\mathcal{T}_u \setminus t_{uv})} \Lambda_p \times r^{t'}_{abp} + \gamma \times r^{c'}_{abu}. \quad (4)$$

3) Overall reputation: Finally, user A calculates the overall reputation of B with respect to question q_k (denoted by Z_{abk}) in Equation (5).

$$Z_{abk} = \alpha \times r_{abk} + (1-\alpha) \times r'_{abk} + \beta(f_b/\Phi) \quad (5)$$

Where $\alpha \in [0,1]$ is the weight placed on direct trust, large value of α means that user want to evaluate another users reputation mainly based on its own experience. f_b is the number of fans B attracts, and Φ is the total number of users in the system. The first two elements in Equation (5) consider the quality of answers provided by B, while the third element considers the general reputation of B. $\beta \in (0,1)$ is the weight assigned on general reputation.

When the system needs to forward question q_k asked by user A to its contacts, the system examines the reputation of all its contacts with respect to question q_k. An example of calculating user A's reputation is shown in Figure 2. Suppose q_k is in category c_u and theme t_{uv}. User A first calculates its direct trust to B based on historical records in step 1, which includes category trust r^c_{abu} and theme trust r^t_{abv}. In step 2,

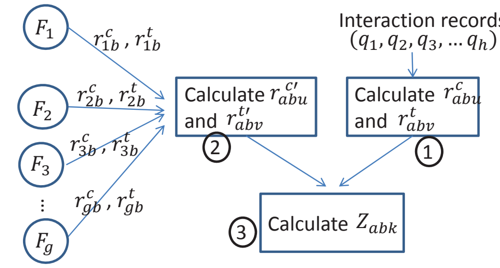


Fig. 2: An example of reputation calculation.

A gathers direct trust of all his fans towards user B, then calculates B's aggregated trust $r^{c'}_{abu}$, and $r^{t'}_{abv}$. Finally, user A calculates an overall reputation for B on question q_k in step 3.

D. Lightweight Spammer Detection

In online Q&A systems, every registered user can post questions and answers. Spammers can take advantage of this free environment and popularity of Q&A systems, and post commercial spam to gain attention for their products. Spammers are detrimental to the Q&A systems as they do not contribute useful information. Thus, identifying spammers quickly and precisely is crucial to maintaining healthy development in Q&A systems.

Study in [2] shows that a linear relationship exists between the number of best answers and the number of all answers of a user, and the correlation coefficient equals 0.712. A spammer tends to post many answers (which are in fact spam), and few of which would be selected as best answers. To determine if user A is a spammer or not, we can examine the ratio (R) of the number of best answers (N^b) and the number of all answers (N_a): $R_a = N^b_a/N_a$. Given a predetermined threshold ξ, if $R_a < \xi$, user A will be identified as a suspected spammer. Although spammers can collude to rate their own answers as best answers, thus increasing the ratio R. However, as the best answers are highlighted in the Q&A forum with high visibility to many other users, the false best answers can be easily identified using the abuse report policy.

Algorithm 1: Pseudo-code of Spammer detection algorithm.

//input: N^b_a, N_a, f_a
$R_a = N^b_a/N_a$
if $R_a < \xi$:
  if $f_a < \tau$:
    user A is a suspected spammer
  end if
end if

The value of ξ should be determined carefully to provide good performance of this spammer detection method. ξ needs to be set large enough to maintain a good detection precision. However, some users who are not knowledgable enough to contribute a minimum ratio of best answers will be falsely identified as spammers, thus increasing the detection false positive rate. To solve this problem, we further propose an incremental strategy to reduce the chance of falsely identifying a normal user as a spammer, which considers the user's social relationship. From [2], we see that a user with higher rank is likely to have a larger number of fans, and the number of fans of all users follows a power-law distribution. We first define a fan count threshold τ, which is a certain percentile of the number of fans of all users in the system. If a user is in the contact lists of a large number of users, the user is not likely to be a spammer. We then compare τ with the number of fans (f_a) that user A has, if $f_a < \tau$, user A is likely to be a spammer. The detailed spammer detection algorithm is shown in Algorithm 1.

IV. Performance Evaluation

We conducted trace-driven experiments on PeerSim [27]. The data set we used is crawled from *Yahoo! Answers* from Aug. 17 to Oct. 19, 2011, which includes: 1) personal information of 119,175 users such as best answer rate (which is the percentage of a user's answers that are chosen by the askers as best answers), number of followers (i.e., fans) and contacts for each users, 2) general information of 119,174 questions such as the categories they belong to and the answers they draw. According to *Yahoo! Answers*, the questions are grouped by 26 categories, including "Travel", "Environment", and 148 themes including "Air Travel" and "Australia" under category "Travel". In the simulation, we deployed 10,000 nodes as users on Q&A system; the users are selected from the trace data who have more than 6 contacts. Follower and contact relationships are set based on user information from the trace data. Each user has 1 to 4 randomly selected question categories (interests), and has 1 to 5 themes under each question category. The expertise level of each user in a category or a theme is chosen from 1 to 10. The expertise level indicates a user's ability to answer questions, higher level in a specific question theme represents higher proficiency in answering questions belonging to the theme. In order to have more capable answerers in the system, in addition to v number of actual answerers in the trace, we also randomly selected $10v$ users from the users who have interest in the question's theme as capable answerers. After receiving a question, if a user is a capable answerer of this question, (s)he will respond after a delay randomly chosen from [1,30] minutes. A user can answer up to 2 questions within every 30 minutes. An asker will rate each answer with scores based on the answerer's expertise level. If an answer is received from a user with level l expertise, then the asker will rate this answer with $l/10$ score. In order to generate answering activities and cumulate reputation scores for users, we executed a warm-up process by launching 20,000 questions. During the test, user reputation was updated every 30 minutes. Λ_p and γ in Equation (2) are set to 0.4 and 0.6, respectively; α and β in Equation (5) are to 0.7 and 0.5, respectively; other parameters are set as λ=1, τ=10 and η=100. The simulation contains a 12 hours process, within every 30 minutes, a number of random users post questions and these questions are forwarded to their contacts.

In our proposed SmartQ Q&A system, when a user posts a question, the contacts of this user are sorted by their reputation scores on the question's theme and category. The question is forwarded to 3 contacts with the highest reputation scores. If no answer returns after 30 minutes, this question is forwarded to the next 3 contacts, and then the question forwarding operation is terminated. We compared our proposed *SmartQ* Q&A system with three strategies. In the *Flooding* strategy, a user's question is broadcasted to all its contacts; In *Rank*, a user's question is forwarded to the user with the 3 highest best answer rate among its contacts; *SOS* [28] forwards a user's question to 3 contacts who have the highest similarity value to the asker, and the similarity is measured by examining the contact's interests and social closeness to the asker. Similar to *SmartQ*, if no answer returns after 30 minutes, *Rank* and *SOS* will forward a question to the next 3 contacts with the highest best answer rate and similarity value to the asker, respectively, and then the question forwarding operation is terminated. We are interested in the following metrics:

- **Response rate** The percentage of questions that can receive at least one answer [29].
- **Answer quality** The rating of answers given by the askers.
- **Response latency** The time spans from when a question is launched until it draws the first answer.
- **Overhead** The number of forwards executed by the system.

A. Simulation results

We examine the overall performance of *SmartQ* in terms of all interested metrics. Figure 3 shows the question response rate when there are different numbers of new questions posted in the system within 30 minutes. We see that *Flooding* achieves the best response rate at around 0.85 when the new question arrival rate is small. The response rate drops gradually when the question arrival rate increases, as users do not have enough capacity to answer all new questions. *Rank* yields the least response rate as new questions are always forwarded to users with high best answer rate, and these users are not capable of providing answers to all new questions. Also, users with high best answer rate may not have the expertise to answer a specific question. *SOS* outperforms *Flooding* and *Rank* due to the reason that *SOS* locates potential answerers by examining the closeness of user interests and a question's category. *SmartQ* is effective in providing high question response rate under different question arrival rates, due to the reason that the question is forwarded to a limited number of users with expertise in the question's specific area.

Figure 4 shows the average answer quality when new questions are posted in the system at different rates, which is evaluated by averaging all answer scores received from

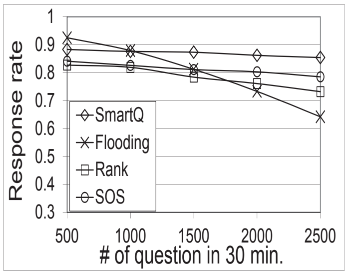


Fig. 3: The question response rate.

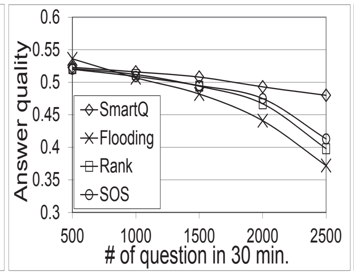


Fig. 4: The average quality of answers.

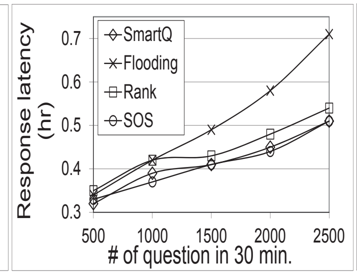


Fig. 5: Average latency of receiving answers.

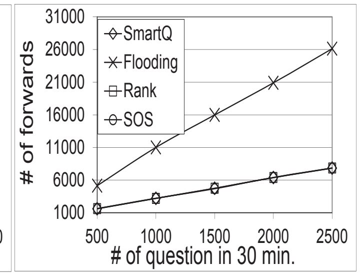


Fig. 6: # of forwards.

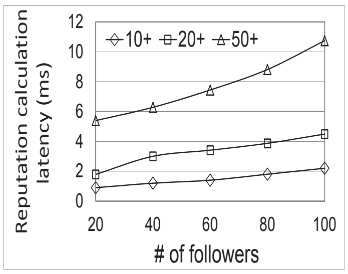


Fig. 7: Average reputation calculation latency.

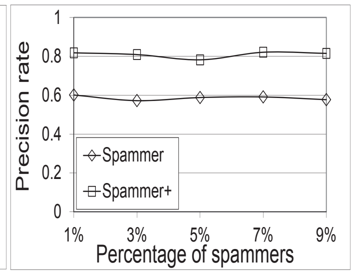


Fig. 8: Precision rate of spammer detection strategy.

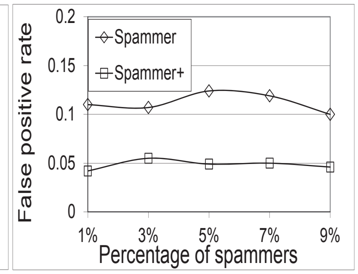


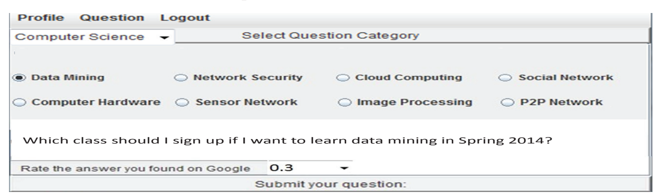
Fig. 9: False positive rate of spammer detection strategy.

askers. If no answer is provided by an asker, the score for this answer quality is 0. We see that *SmartQ* is advantageous in maintaining high answer quality at about 0.5, due to the reason that the question is forwarded to potential answerers with high reputation in the question's specific area. *SOS* achieves higher answer quality than *Flooding* and *Rank* as it studies the similarity between question category and user interests. However, a user's interest in a question category does not guarantee sufficient expertise in solving questions in this category, thus *SOS* gets lower answer quality than *SmartQ*. *Flooding* and *Rank* both do not consider the users' ability and willingness to answer a specific question, so they cannot provide high answer quality for askers.

Figure 5 shows the average latency of receiving answers, which is measured from the time a question is launched until the time the first answer arrives. We see that as questions are posted at a higher rate in the system, the average latency of receiving answers increases for all strategies, due to the reason that users can provide a limited number of answers within every time period. Also, both *SmartQ* and *SOS* outperform other two strategies in reducing answering latency, as they both explore users' expertise or interests while forwarding questions. *Flooding* can easily overwhelm users with an excessive number of questions, thus the average answering latency is increased.

Figure 6 shows the total number of forward actions executed with different new questions arrival rates. We see that *SmartQ*, *Rank* and *SOS* need less number of forward actions than *Flooding*, as they only forward a new question to at most 6 users.

(a) Snapshot of *SmartQ*'s main menu.



(b) Snapshot of *SmartQ*'s question page.

Fig. 10: Snapshot of *SmartQ* Q&A system.

Figure 7 shows the computation cost of reputation calculation when there are different numbers of followers for each user. Three lines represent calculation time for different numbers of users. We see that the latency of reputation calculation is generally short, and it takes about 11ms to finish reputation calculation of 50 users. Note that the calculation time can be further reduced by parallelism. Figure 7 indicates that *SmartQ* is able to execute fast question forwarding actions.

Figure 8 and Figure 9 show the performance of our proposed Lightweight Spammer Detection strategy (denoted by *Spammer+*). We compare it with *Spammer* strategy, in *Spammer*, each user has a 20% probability of reporting a spammer to the system, and 5% probability of falsely reporting a normal user as a spammer. Figure 8 shows the precision rate of different spammer detection strategy when there are

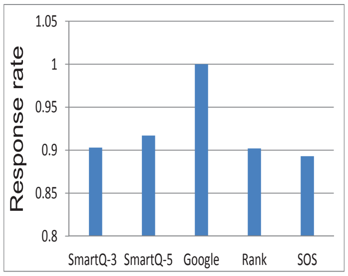


Fig. 11: The question response rate.

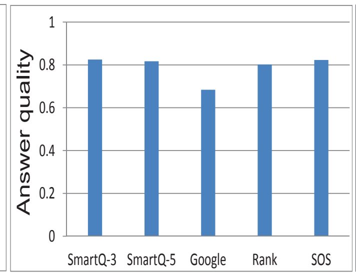


Fig. 12: The average quality of answers.

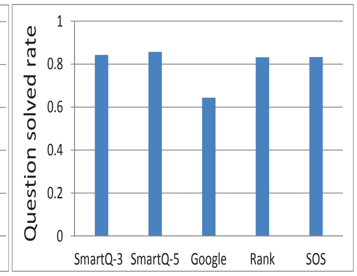


Fig. 13: Question solved rate.

different percentages of spammers in the system. We see that *Spammer+* outperforms *Spammer* in increasing the precision rate of spammer detection. Figure 9 shows the false positive rate of different spammer detection strategy when there are different percentages of spammers in the system. We see that *Spammer+* exhibits a low false positive rate.

B. Application implementation and testing

We developed *SmartQ* client based on Java Applet framework, and the server runs on Tomcat 7.0 using JDBC connector with MySQL. The client is running on any browser supporting Java runtime environment 1.7. 42 students from Clemson University installed *SmartQ* clients and participated in the test. Figure 10 shows the main menu and question page of *SmartQ*. As shown in Figure 10(a), users can ask and answer questions that meet their interests, and check question history. When a user wants to ask a question, he/she is required to select the question category and detailed themes for the question. As shown in Figure 10(b), ”Computer Science” is selected as the question category and ”Data Mining” as the question theme. Users are also required to search the question on Google and rate the Google results with a score ranged from 0 to 1.

After receiving an answer, an asker needs to rate each answer with a 0-1 score based on the answer quality. The test lasted one month and more than 300 questions were collected and analyzed. The questions were mainly focused on the ”Computer Science” category, and multiple themes under this category are presented to further identify the questions. We proposed two different question forwarding strategies: *SmartQ-3* and *SmartQ-5*, which forward a question to 3 and 5 contacts with the highest reputation scores, respectively. We then compared the performance of *SmartQ* with *Google*, *SOS* and *Rank*.

We first examine if users can receive answers for their questions. Figure 11 shows the comparison results of different methods in question response rate. We see that *SmartQ* outperforms *SOS* and *Rank* in drawing answers, due to the reason that the questions are forwarded to users with high reputation, which reflects the users’ willingness and ability to answer each specific category of questions. Also, *SmartQ-5* achieves higher question response rate than *SmartQ-3*, because forwarding a question to larger number of users will result in higher chance of reaching a potential answerer, thus questions are more likely to be solved. *SOS* yields a better response rate than *Rank*, as *SOS* aims to match the question’s category with potential askers’ interests. We assume that *Google* reaches a response rate of 100 percentage as the search engine is always available for information discovery.

Each asker in the test evaluates the answers for his/her questions by assigning quality scores. Figure 12 shows the average answer quality for different strategies. We see that *Google* provides lower quality answers for users than the other four strategies, due to the fact that most questions asked by participants in the testing group are non-factual questions. Questions such as ”What kind of personal information is safe to disclose on social network?”, ”What mathematical knowledge is important when studying data mining?”, cannot be easily found on Google, but can be solved by users with expertise in that areas. *SOS* considers user interests and expertise when forwarding questions, so it gets higher answer quality than *Rank*, which only considers users’ ranks they achieve in the system, but does not identify users’ expertise and willingness when forwarding new questions. *SmartQ* achieves the highest average answer quality due to the same reason in Figure 4.

When an asker receives a number of answers to his/her question, it is important to determine whether the question is solved or not. In the test, if an asker considers that an answerer solves the question, he/she will give higher than 0.8 points to this question. Thus, a question is solved if at least one of its answers receive more than 0.8 points. Figure 13 shows the percentage of questions solved by users in different strategies. We see that the question solved rate follows: *SmartQ-5*>*SmartQ-3*>*SOS*>*Rank*>*Google*. *SmartQ* selects potential answerers based on their expertise in the question’s area and willingness to answer questions, thus questions are more likely to be solved. *SOS* forwards questions to users who are interested in the questions’ area but may not have the ability to answer questions. Thus *SOS* outperforms *Rank*, which does not match potential answerers’ expertise to the questions’ area. For most non-factual and subjective question, *Google* is incapable of providing satisfying answers.

In the test, we chose 4 random students to be spammers, who were responsible for answering questions with advertisements or randomly generated words. After receiving spam to his/her question, the asker can report it to the system by clicking the "Report Spam" button, or choose to ignore the spam without reporting. We tested two different methods for spammer detection, one is the proposed lightweight detection strategy and the other is *Reported-based*. In *Reported-based*, if a user is reported as a potential spammer by at least 3 users, the system will finally regard he/she as a spammer. Both strategies are able to identify all 4 spammers, which indicates the effectiveness of our proposed lightweight spammer detection strategy.

V. Conclusions

The rapid growth of Q&A systems make them important ways of knowledge discovery. However, as Q&A systems are generally serving a large amount of users and tens of thousand of new questions are posted in the system everyday, forwarding questions to experts who are willing and able to provide satisfying answers is crucial in maintaining the performance of Q&A systems. Also, in order to improve user loyalty and experience, Q&A systems should be able to identify users who intentionally spread useless information or post advertisements. This paper proposes SmartQ, a reputation based Q&A System. SmartQ evaluates users' reputation towards every knowledge category and theme, and forwards questions to a number of reputable users in the question's knowledge category and theme. Also, SmartQ incorporates a lightweight spammer detection strategy, which examines a user's best answer rate and number of contacts. The advantage of SmartQ is verified by experiments on PeerSim and real application. In our future work, we will study using effective incentives to further improve answer quality and response rate, and detecting malicious users by analyzing their behaviors.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, Microsoft Research Faculty Fellowship 8300751.

References

[1] Yahoo!Answers. http://answers.yahoo.com/, [Accessed in March 2014].

[2] Z. Li, H. Shen, and J. Grant. Collective intelligence in the online social network of yahoo!answers and its implications. In *Proc. of CIKM*, 2012.

[3] U. Lee, H. Kang, E. Yi, M. Yi, and J. Kantola. Understanding Mobile Q&A Usage: An Exploratory Study. In *Proc. of CHI*, 2012.

[4] R. White, M. Richardson, and Y. Liu. Effects of community size and contact rate in synchronous social Q&A. In *Proc. of SIGCHI*, 2011.

[5] G. Hsieh, R. Kraut, and S. Hudson. Why Pay?: Exploring How Financial Incentives are Used for Question & Answer. In *Proc. of CHI*, 2010.

[6] K. Nam, M. Ackerman, and L. Adamic. Questions in, knowledge in?: a study of naver's question answering community. In *Proc. of CHI*, 2009.

[7] F. Harper, D. Raban, S. Rafaeli, and J. Konstan. Predictors of answer quality in online Q&A sites. In *Proc. of CHI*, 2008.

[8] L. Gomes, C. Cazita, J. Almeida, V. Almeida, and W. Meira. Characterizing a spam traffic. In *Proc. of IMC*, 2004.

[9] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proc. of SIGCOMM*, 2004.

[10] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao. Spam behavior analysis and detection in user generated content on social networks. In *Proc. of ICDCS*, 2012.

[11] Google Answers. http://answers.google.com/, [Accessed in March 2014].

[12] StackOverFlow. www.StackOverFlow.com/, [Accessed in March 2014].

[13] L. Adamicand J. Zhang, E. Bakshy, and M. Ackerman. Knowledge Sharing and Yahoo Answers: Everyone Knows Something. In *Proc. of WWW*, 2008.

[14] D. Horowitz and S. D. Kamvar. The anatomy of a large-scale social search engine. In *Proc. of WWW*, 2010.

[15] M. Richardson and R. W. White. Supporting synchronous social q&a throughout the question lifecycle. In *Proc. of WWW*, 2011.

[16] S. J. H. Yang and I. Y. L. Chen. A social network-based system for supporting interactive collaboration in knowledge sharing over peer-to-peer network. *IJHCS*, 2008.

[17] M. S. Ackerman. Augmenting organizational memory: a field study of answer garden. *ACM TOIS*, 1998.

[18] H. Kautz, B. Selman, and M. Shah. ReferralWeb: Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3), March 1997.

[19] D. McDonald and M. Ackerman. Expertise Recommender: A flexible recommendation system and architecture. In *Proc. of CSCW*, 2000.

[20] C. Shah, J. Oh, and S. Oh. Exploring Characteristics and Effects of User Participation in Online Social Q&A Sites. *First Monday*, 13(9), 2008.

[21] J. Guo, S. Xu, S. Bao, and Y. Yu. Tapping on the Potential of Q&A Community by Recommending Answer Providers. In *Proc. of CIKM*, 2008.

[22] Y. R. Tausczik and J. W. Pennebaker. Predicting the Perceived Quality of Online Mathematics Contributions from Users' Reputations. In *Proc. of SIGCHI*, 2011.

[23] K. Pelechrinis, V. Zadorozhny, and V. Oleshchuk. Collaborative assessment of information provider's reliability and expertise using subjective logic. In *Proc. of CollaborateCom*, 2011.

[24] P. Long, N. Anh, N. Vi, L. Quoc, and H. Thang. A meaningful model for computing users' importance scores in Q&A systems. In *Proc. of SoICT*, 2011.

[25] R. Delaviz, J. Pouwelse, and D. Epema. Targeted and scalable information dissemination in a distributed reputation mechanism. In *Proc. of STC*, 2012.

[26] M. Blaze, J. Feigenbaum, and A. Keromytis. Keynote: Trust management for public-key infrastructures (position paper). In *Proc. of Security Protocols Workshop*, 1998.

[27] Peersim: A peer-to-peer simulator. http://peersim.sourceforge.net/, [Accessed in March 2014].

[28] Z. Li, H. Shen, G. Liu, and J. Li. SOS: A Distributed Mobile Q&A System Based on Social Networks. In *Proc. of ICDCS*, 2012.

[29] J. Janes, C. Hill, and A. Rolfe. Ask-an-expert services analysis. *JASIST*, 52(13):1106–1121, 2001.