SocialQ&A: An Online Social Network Based Question and Answer System

Haiying Shen*, *Senior Member IEEE*, Guoxin Liu, Haoyu Wang, Nikhil Vithlani

Abstract—Question and Answer (Q&A) systems play a vital role in our daily life for information and knowledge sharing. Users post questions and pick questions to answer in the system. Due to the rapidly growing user population and the number of questions, it is unlikely for a user to stumble upon a question by chance that (s)he can answer. Also, altruism does not encourage all users to provide answers, not to mention high quality answers with a short answer wait time. The primary objective of this paper is to improve the performance of Q&A systems by actively forwarding questions to users who are capable and willing to answer the questions. To this end, we have designed and implemented SocialQ&A, an online social network based Q&A system. SocialQ&A leverages the social network properties of common-interest and mutual-trust friend relationship to identify an asker through friendship who are most likely to answer the question, and enhance the user security. We also improve SocialQ&A with security and efficiency enhancements by protecting user privacy and identifies, and retrieving answers automatically for recurrent questions. We describe the architecture and algorithms, and conducted comprehensive large-scale simulation to evaluate SocialQ&A in comparison with other methods. Our results suggest that social networks can be leveraged to improve the answer quality and asker's waiting time. We also implemented a real prototype of SocialQ&A, and analyze the Q&A behavior of real users and questions from a small-scale real-world SocialQ&A system.

Index Terms—Question and answer systems, Social networks, Information search

1 INTRODUCTION

The Internet is an important source of information, where the amount of data is vast and constantly growing. Users rely on search engines to find specific information in this knowledge base. Search engines such as *Google* and *Bing* use keywords provided by the users to perform searches. Recently, industrial research and development activities, such as Microsoft and Facebook's social-featured Bing search endeavor, try to combine search engines and online social networks for higher search performance. As previous research has indicated [1, 2], search engines perform well in indexing web pages and providing users with relevant content to their search but are not suited for non-factual questions such as "Which is the best local auto shop?". To address this particular class of non-factual questions, many Question and Answer (Q&A) systems such as *Yahoo! Answers*, *Baidu Zhidao*, *StackExchange*, *Quora* and *Ask* have been developed. Since their inception, Q&A systems have proved to be a valuable resource for sharing expertise and consequently are used by a large number of Internet users. For example, *Yahoo! Answers* was launched at the end of the year 2005 and attracted more than 10 million users in February of 2007 [3], and hit 200 million users in December of 2009 [4, 5]. Q&A systems also preserve all questions and answers, thus acting as a repository for information retrieval. They are not only important for sharing technical knowledge, but also as a source for receiving advice and satisfying one's curiosity about a wide variety of subjects [6].

With a vast population in a Q&A system, a large number of questions are posed online every day. For example, there are 823,966 questions and answers posed to *Yahoo! Answers* per day [4]. Then, when a user intends to answer a question, (s)he may be overwhelmed by the plethora of questions. Moreover, simply relying on altruistic users to provide answers cannot encourage all users to provide answers and to answer questions quickly. To locate appropriate answer providers, current Q&A systems allow users to choose tags (i.e., interest categories) for their questions. However, it may not be easy to determine the appropriate tag(s) for a question such as "how is the computer organization class at our university?".

As a result, current Q&A systems may not meet the requirement of providing high quality answer with a short answer wait time, though users wish to receive satisfactory answers quickly. This is confirmed by the study in [5]. It found that for *Yahoo! Answers*, only 17.6% of questions were answered satisfactorily; for the remaining 82.4%, one fifth of the questions remained unanswered. For *Baidu Zhidao*, 22.7% of questions were successfully answered, and 42.8% of the unresolved questions were not answered at all. Thus, there is an increasing need for an advanced Q&A system that can decrease the number of unanswered questions, enhance the answer quality and decrease the response time.

In addition, the privacy of the Q&A system is very important nowadays. Many users may ask or answer questions related to sensitive topics such health problem, political activism or even sexual orientation [7]. Although the user may want the response as soon as possible, he/she still needs the privacy protection to avoid potential disclosure of personal information [8]. Since Social Q&A is built upon social networks. The asker and answerer are social close to each other. Therefore, protecting the privacy is important and challenge.

To meet this need, we propose SocialQ&A, an online social network based Q&A system, that actively forwards questions to those users with the highest likelihood (capability and willingness) of answering them with expertise and interest in the questions' subjects. The design of SocialQ&A is based on two social network properties. First, social friends tend to share similar interests (e.g., lab members majoring in computer systems) [9]. Second, social friends tend to be trustworthy and altruistic due to the property

- * *Corresponding Author. Haiying Shen, Email: hs6ms@virginia.edu; Phone: (434) 924-8271; Fax: (434) 982-2214.*
- *Guoxin Liu,Haiying Shen, Haoyu Wang and Nikhil Vithlani are with the Department of Computer Science, University of Virginia, Charlottesville, VA, 22904. E-mail: {hs6ms,hw8c}@virginia.edu*

of "friendship fosters cooperation" [10]. Accordingly, SocialQ&A favors routing queries among friends and identifies a question's potential answerers by considering two metrics: the interest of the friend towards the question and the social closeness of the friend to the asker/forwarder. Thus, the answer receivers have high probability of providing high-quality answers in a short time [11, 12]. Different from the existing Q&A systems, due to the importance of users privacy, we future introduce security and efficiency enhancement to protect users privacy while users using social network answering questions. The contributions of this work are as follows:

- *The design of SocialQ&A.* SocialQ&A is composed of three components: *User Interest Analyzer*, *Question Categorizer*, and *Question-User Mapper*. *User Interest Analyzer* associates each user with a vector of interest categories. *Question Categorizer* associates a vector of interest categories to each question. Then, based on user interest and social closeness, *Question-User Mapper* identifies potential answerers for each question.
- *The design of security and efficiency enhancement methods.* SocialQ&A incorporates three methods to enhance its security and efficiency performance. The *bloom filter based personal information exchange* method protects users' privacy including friendship and interest information. The *onion routing based answer forwarding method* protects the identities of the asker and the answerer from being exposed. The *answer retrieval for recurrent questions* automatically finds the answers for recurrent questions.
- *Comparative trace-driven experiments.* We conducted comprehensive large-scale simulation to evaluate SocialQ&A in comparison with other methods. Our results suggest that SocialQ&A improves the quality of answers and reduces the wait time for answers.
- *The development of a real-world SocialQ&A.* We have prototyped the SocialQ&A system with user interfaces, and conducted a real-world small-scale test with real users from India, the United Kingdom, and the United States for a period of approximately one month.
- *The analysis of the data from real SocialQ&A.* We have analyzed the features of the questions posted, the questioning and answering activities of users, the quality of answers, and the wait time for answers. Analytical results show the benefits of SocialQ&A in enhancing answer quality and wait time.

Please note that is it difficult for us (as a small school lab) to implement a real-world SocialQ&A system that can attract thousands of users to conduct the performance evaluation. We only implemented a real-world SocialQ&A system that attracted a small number of users, which is better than no real-world implementation. We do not claim that such a system can represent a large-scale social network, but some our findings are interesting and they follow the previous observations and confirm the assumptions in the social network based Q&A systems. The rest of this paper is structured as follows. Section 2 presents a concise review of related work. Sections 3 and 4 present the details of the design of SocialQ&A and its enhancement methods. Section 5 measures the SocialQ&A's performance in comparison with other systems through trace-driven experiments. Section 6 describes the user interface of a real-world SocialQ&A prototype, and analyzes a real trace obtained from it. Section 7 concludes this paper with remarks on our future work.

2 Related Work

The growing importance of Q&A systems demands an effort to better understand these systems and to improve them [13]. The works in [14–19] studied the influence of different factors (e.g., users' profiles, messages prediction, system interactions and community size) in the social networks on Q&A performance. These study results lay the foundation of SocialQ&A to leverage social network properties [20] in the design. Note that the existing social network based on the asker-answerer relationship in current Q&A systems [17] is different from online social network based on the social relationship, which is used in SocialQ&A. The works in [21–24] concentrated on locating experts and authoritative users. Instead, SocialQ&A aims to find normal users that can answer questions including opinion-type questions. Some studies have been conducted to create reputation models in Q&A systems [25, 26] to increase the credibility of answers, and to determine the relationship between the reputation of the users and the quality of their provided answers [27]. SocialQ&A directly utilizes the social network property of mutual-trust friendship to motivate users to provide answers without relying on an additional reputation model. SocialQ&A shares similarity with other peer-assistant systems such as [28] in leveraging the collective power of peers for a certain goal.

Some research [29–31] categorizes questions into predefined categories, making it easier for users to locate previously asked questions and for experts to find questions they can answer. Quan *et al.* [30] proposed three new supervised term weighting schemes for question categorization, and evaluated each scheme using a trace from *Yahoo! Answers*. Song *et al.* [31] proposed a sequential process including topic-wise word identification and weighting, semantic mapping, and similarity calculation.

Text mining techniques also have been used to provide better answers [5, 32–36]. These categorization and text mining methods can be used in SocialQ&A to more accurately derive user interests and question interests. Li *et al.* [5] proposed a language model by combining expertise estimation and availability estimation, and later proposed category-sensitive language models [32] for expert identification, which helps route questions to available and capable experts. Zhou *et al.* [33] classified the questions using a variety of local and global features of questions and users' relationship in order to route a classified question to its potential answerers. Cao *et al.* [34] leveraged question category to enhance question retrieval in community-based Q&A systems. Guo *et al.* [35] proposed a topic-based model to identify appropriate answerers by calculating the similarities between questions' topics and users specialists. Nie *et al.* [36] proposed a scheme which can annotate social questions automatically to unravels the incomplete and biased problems of question tags.

Compared to previous Q&A system works, SocialQ&A also leverages both the common-interest and mutual-trust social network properties to improve the QoS performance. It incorporates different algorithms to determine user interest, question interest and the question-user mapping. Unlike previous Q&A system works, it does not assume that friendship is always trustable and incorporates algorithms that avoid revealing personal information to others as little as possible. Different from previous Q&A system works, our previously proposed SOS [39] is also a Q&A system based on a social network. However, SOS focuses on realizing a mobile Q&A system in a distributed manner and using knowledge engineering techniques. Also, it assumes that social closeness is already provided by users. Instead, SocialQ&A focuses on how to leverage social network properties in better identifying potential answerers with predefined interest categories and showing its benefits through the analysis on real users' Q&A activities.

3 SocialQ&A: An Online Social Network based Q&A System

3.1 The Rationale of SocialQ&A Design

A real-life social network is formed by regarding each person as a node and linking two nodes with a social relationship. This network is featured by social communities such as the football club and ECE department at a university. In real life, the people we rely on for answers to questions such as "how is the computer organization class at our university?" are usually those in our social communities. Persons in the same social community share common interests and trust each other on answering questions on their common interests, and are willing to answer the questions from community members.

An online social network connects friends with real-life relationship and online friendship, which shares similarity to the real-life social network. Friends in an online social network tend to share similar interests and trust each other [9, 40, 10]. Taking advantage of these properties, we design and develop SocialQ&A that incorporates an online social network to improve the quality of answers and decrease answer wait time. It forwards a user's questions to his/her social friends that have common interest and a close social relationship.

3.2 The Design of SocialQ&A

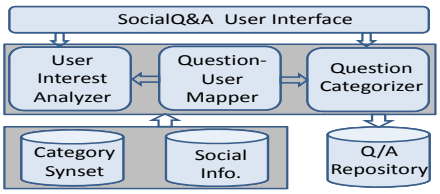


Fig. 1: The architecture of SocialQ&A.

Like all online social networks, the one in SocialQ&A has user profiles that record users' interests, education, hobbies and etc. Like *Yahoo! Answers*, SocialQ&A also predefines interest categories and subcategories. A total of 4 categories (music, movies, television, and books) and 32 subcategories (e.g., books: novel, drama) derived from *Yahoo! Answers* were used to implement SocialQ&A. We used these 4 categories as an example and will add more categories in our future work.

Figure 1 shows the high-level architecture of SocialQ&A and the interaction between the core components: *User Interest Analyzer*, *Question Categorizer*, and *Question-User Mapper*. *User Interest Analyzer* analyzes data associated with each user in the social network to derive user interests. *Question Categorizer* categorizes the user questions into interest categories based on the *Category Synsets*, which stores the synonyms of all categories' keywords from WordNet [41]. *Question-User Mapper* connects these two components by identifying potential answerers who are most likely to be willing to and be able to provide satisfactory answers. The data from user questions and answers is stored on *Q/A Repository* to serve subsequent similar questions. Below, we present each component and user interface.

3.2.1 User Interest Analyzer

User Interest Analyzer utilizes each user's profile information in the social network and user interactions (answers provided and questions asked) to determine the interests of the user in the predefined interest categories. This is because if a user asks or answers questions in an interest category, (s)he is likely to be interested in this particular category. As shown in Figure 2 (Including Rock music, classic music, action movie, thriller movie, news, shows and story), the interests of user U_j are represented by a user interest vector V_{U_j}=<I_i, W_{I_i}> ($i = 1, 2...$), where I_i represents an interest and W_{I_i} represents the weight (degree) of the user's interest in interest I_i. $W_{I_i} = 0$ indicates that the user does not have the corresponding interest. W_{I_i} is incremented by 1 for each appearance of the interest in the parsed information from a user's profile and interactions. The order of phrases does not necessarily represent the different preferences of a user. Thus, we count the frequency that an interest's synset appears in all phrases to indicate the user's perference on this interest, because the frequency represents a user's focus on an interest currently.

Algorithm 1 Pseudocode for the *User Interest Analyzer*.

Input: A user's profile, questions and answers
Output: The user's interest vector $V_{U_j} =< I_i, W_{I_i} >$

1: Parse the "interests" field to generate a token stream T_I
2: Parse the "activities" field to generate a token stream T_a
3: Use the inputs from the user's selection from the Music, Movie, Television and Book fields to generate token streams T_{mu}, T_{mo}, T_t and T_b
4: **for** each token stream T_x (T_x=T_I, T_a, T_{mu}, T_{mo}, T_t, T_b) **do**
5: Check each token in the Synset
6: **if** a matching interest category I_i exists **then**
7: Update interest weight: W_{I_i}++ (e.g., W_{music}++)
8: **end if**
9: **end for**
10: Keep updating W_{I_i} based on questions asked and answered and profile update
11: Periodically update W_{I_i} using $W_{I_i} = \alpha * W_{I_{i_{old}}}$

Algorithm 1 shows the pseudocode for the *User Interest Analyzer*. When a user registers for SocialQ&A, (s)he is given the option of entering his/her interests and activities and to mark predefined interest categories to add to his/her interest list. SocialQ&A uses WordNet to parse these text fields to token streams (Steps 1-3). For every token, its matching interest category is located in the Synset and corresponding weight is updated (Steps 4-9).

	Rock	Classic	Action	Thriller	News	Shows	Story
User i	2	0	3	0	1	0	4

Fig. 2: User interest vector.

For accurate user interest reflection, SocialQ&A keeps track of profile changes, the questions asked and answered by a user to update his/her interest vector. A user can indicate the interest tags for his/her questions. In the indicated tags and parsed interests, we use O_{I_i} to denote the number of occurrences of the interest I_i during the previous period. For an interest I_i, its weight is updated to $W_{I_{i_{new}}} = W_{I_i} + O_{I_i}$, where $W_{I_{i_{new}}}$ is the weight used in next period. In order to reflect users' current interests, the weight is periodically decayed by: $W_{I_i} = \alpha * W_{I_{i_{old}}}$, where $W_{I_{i_{old}}}$ is the weight used in last period

3.2.2 Question Categorizer

The primary task of *Question Categorizer* is to categorize a question into predefined interest categories based on the topic(s) of the question. We also allow users to input self-defined tags associate with questions, which are analyzed in question parsing. *Question Categorizer* generates a vector of question Q_i's interests, denoted by V_{Q_i}, using a similar algorithm as Algorithm 1. While processing a question, SocialQ&A uses WordNet to examine the tags and text of the question and generates a token string. The tokens are

compared to SocialQ&A's Synset to determine the categories where the question belongs. We have calculated the interest weight without normalization in order to predict the user intelligence to answer a question of Interest.

3.2.3 Question-User Mapper

Question-User Mapper identifies the appropriate answerers for a given question. The potential answer providers are chosen from the asker's friends in the online social network. Note that the changes in a user's friends in the online social network do not affect the performance of SocialQ&A as it always uses a user's current friends. To check the appropriateness of a friend (U_k) as an answer provider for a question, two parameters are considered: i) the interest similarity between the interest vectors of the friend and the question (denoted by Ψ_{I,U_k}); and ii) the social closeness between the friend and the asker (denoted by Ψ_{C,U_k}). The former represents the potential capability of a friend to answer the question, and the latter represents the willingness of a friend to answer the question.

We use $W_{I_j}^{U_k}$ to denote U_k's weight on interest I_j. For the asker's question with vector V_{Q_i},

$$\Psi_{I,U_k} = \sum_{I_j \in (V_{U_k} \cap V_{Q_i})} W_{I_j}^{U_k}. \quad (1)$$

In the online social network, a user's friends with more common interests, frequent interactions or common friends (i.e., higher social closeness) are more willing to respond to the user's question [39, 17, 2, 14]. Thus, to calculate Ψ_{C,U_k} between friend U_k and the asker, we consider three metrics: i) the similarity between their interest vectors (denoted by S, which is incremented by each matching entry); ii) their asking and answering interaction frequency (denoted by A); and iii) the number of their common friends, denoted by C. Given the asker's friend set $\mathcal{F}$, friend U_k's rates of S, A and C are calculated by:

$$P_{S_{U_k}} = \frac{S_{U_k}}{\sum_{i \in \mathcal{F}} S_i},\ P_{A_{U_k}} = \frac{A_{U_k}}{\sum_{i \in \mathcal{F}} A_i},\ P_{C_{U_k}} = \frac{C_{U_k}}{\sum_{i \in \mathcal{F}} C_i}. \quad (2)$$

Then, the social closeness of friend U_k is calculated as

$$\Psi_{C,U_k} = \gamma_S * P_{S_{U_k}} + \gamma_A * P_{A_{U_k}} + \gamma_P * P_{C_{U_k}}, \quad (3)$$

where γ_S, γ_A and γ_P are the weights of considering factors S, A and C, respectively. Since we make all metrics comparable by scaling them to $[0, 1]$, the weights represent the correlationship between each factor and the social closeness. We finally calculate the metric Ψ_{U_k} to measure the appropriateness of friend U_k as a potential answerer for U_j's question Q_i. That is:

$$\Psi_{U_k} = \beta * \Psi_{I,U_k} + (1 - \beta) * \Psi_{C,U_k}\ (0 < \beta < 1), \quad (4)$$

where β is the weight for each consideration factor. In different circumstances, we can give different β values. Higher β value helps identify friends with higher capability to answer the question, while a lower β value helps identify friends with higher willingness to answer the question.

SocialQ&A then orders an asker's friends in the descending order of their Ψ_{U_k} values, and routes the question to the top N friends. N is a tradeoff between system overhead and response efficiency. If N is larger, the system overhead is larger, but the answer response efficiency is improved; and vice versa. Algorithm 2 shows the pseudocode of the *Question-User Mapper*. Social distance between two nodes is the number of hops in the shortest path between them in the online social network. If no one responds during a specific time period, SocialQ&A can try the nodes in 2-hop social distance from the asker, and then in 3-hop social distance, until the nodes in Time-To-Live (TTL)-hop social distance have attempted. A question receiver can forward the question if (s)he cannot answer it. The question-user mapper algorithm is called while asking or forwarding questions. When forwarding a question, the asker's information is replaced by the forwarder's information. The *Question-User Mapper* can be executed in either a centralized manner or a decentralized manner [39]. In the centralized manner, the centralized server selects the potential answerers for each question and sends the question to them. In the decentralized execution, each node autonomously determines the potential answerers for the question initialized or received by itself to send the question. If there are not enough N selected friends through the *Question-User Mapper*, the remaining answerers are randomly selected from all users having such interests.

Algorithm 2 Pseudocode for the *Question-User Mapper*.

Input: Interest vectors of a user, his/her friends and question
Output: A list of potential answer providers

```
for each friend U_k in the friend set of U_j do
    Compute Ψ_{I,U_k} based on Eq. (1)
    Compute P_{S_{U_k}}, P_{A_{U_k}} and P_{C_{U_k}} based on Eq. (2)
    Compute Ψ_{C,U_k} based on Eq. (3)
    Compute Ψ_{U_k} based on Eq. (4)
end for
Order the friends in descending order of Ψ_{U_k}
Notify the top N friends
```

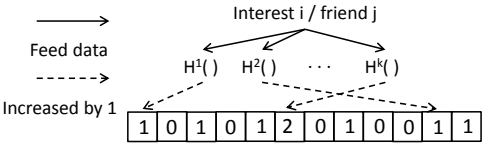


Fig. 3: An example of the counting bloom filter.

4 Security and Efficiency Enhancement

4.1 Secure Personal Information Exchange and Answer Forwarding

The friendship through online social networks may not be always trustable. It is important for users to reveal personal information to each other as little as possible. Besides, the askers and answerers for some questions, such as political sensitive questions, may want to be anonymous to the public. Therefore, a Q&A system should support secure question forwarding process through untrustable friendships. In the following, we propose bloom filter based personal information exchange method and onion routing based answer forwarding method to achieve a certain degree of security.

4.1.1 Bloom Filter based Personal Information Exchange

Section 3.2 introduces how the question-user mapper conducts potential answerer selection, which requires friends to exchange their personal information including their friend lists and interest vectors. To protect user privacy to a certain extent, friends should avoid exchanging such personal information directly. Instead, they should exchange the encrypted information of their friend lists and interest vectors. The challenge here is that the encrypted information should not only protect a user from revealing direct information to others but also serve counting the common friends and interests. The counting bloom filter technique [42] can meet this requirement. Therefore, to handle this challenge, SocialQ&A uses the counting bloom filter technique [42] to encrypt information that is exchanged between friends.

Figure 3 shows an example of a counting bloom filter. A counting bloom filter uses K hash functions to encrypt

personal information for protection. The bloom filter results are stored in an integer array of t entries. Each hash function encrypts the feed information into an integer m within $[0, t]$, and the m^{th} entry of the integer array is increased by 1. To search whether an information item is stored in a bloom filter, the information item is encrypted by each hash function of the bloom filter. If for each hashed result m, the value at m^{th} entry in the array is larger than 0, this information item has a higher probability of being stored in the bloom filter; otherwise, it is not stored in the bloom filter.

Then, to protect the friendship information of users, each user U_k feeds each of his/her friend IDs into a bloom filter, and its bloom filter result as shown in Figure 3 is denoted by $B^f_{U_k}$. Then, friends exchange the bloom filter results instead of friendship information directly. To identify the appropriate answerers among a user U_i's friends, the user needs to calculate the parameters in Equation (2). To calculate the social closeness with friend U_k, $P_{C_{U_k}}$, user U_i can find their common friends by searching the existence of each of his/her friends in U_k's friend bloom filter result, $B^f_{U_k}$. However, checking all friends' bloom filter results is time-consuming. Therefore, U_i directly calculates the cosine similarity between user U_i's friend bloom filter result and U_k's friend bloom filter result as $C_{U_k} = \frac{B^f_{U_i} * B^f_{U_k}}{|B^f_{U_i}| * |B^f_{U_k}|}$. Then, we can derive the social closeness, $P_{C_{U_k}}$, according to Equation (2).

We can derive the interest similarity ($P_{S_{U_k}}$) in a similar way. Different from the friend information, each of a user's interests (I_i) has a weight (W_{I_i}). To take into account the interest weight, user U_k feeds each interest I_i for $W^{U_k}_{I_i}$ times into its interest bloom filter, and we use $B^I_{U_k}$ to denote the interest bloom filter result of user U_k. Therefore, the interest bloom filter result represents not only the interest existence but also the interest weights. Then, we can calculate $S_{U_k} = \frac{B^I_{U_i} * B^I_{U_k}}{|B^I_{U_i}| * |B^I_{U_k}|}$. As a result, we can derive the interest similarity between user U_i and fiend U_k according to Equation (2). Finally, using the $P_{C_{U_k}}$ and $P_{S_{U_k}}$ calculated based on the bloom filters, user U_i can calculate the social closeness with friend U_k, Ψ_{C,U_k}, based on Equation (3).

Recall that the similarity between the interest vectors of friend U_k and the question, Ψ_{I,U_k}, is needed in identifying the appropriate answerers based on Equation (4). Then, a problem is how to derive Ψ_{I,U_k} based on the interest bloom filters. Recall that there are K hash functions. Therefore, an interest I_i increases W_{I_i} for each of K different entries among all t entries in the bloom filter result. Then, in the sum of all t entries in the bloom filter result, an interest I_i contributes $W_{I_i} * K$ value, which is K times of the value for a common interest in Equation (1). Thus, we can calculate Ψ_{I,U_k} by $\Psi_{I,U_k} = \sum_{j \in [1...t], E^j_{U_k} > 0 \wedge E^j_{Q_i} > 0} E^j_{U_k}/K$, where $E^j_{U_k}$ is the value at the j^{th} entry of the bloom filter result of U_k's interests, and $E^j_{Q_i}$ is the value at the j^{th} entry of the bloom filter result of question Q_i's interest. Similar to Equation (1), we calculate Ψ_{I,U_k} by summing the weights of the common interests between the friend and the question. The common interests are identified by checking whether each common entries of the two bloom filter results of the friend's interests and the question's interests have a value larger than 0.

In all possible user IDs or interests, a malicious user U_i can check the existence of each user ID or interest in the bloom filter result of his/her friend U_k in order to derive U_k's friends and interests. Note that a bloom filter is generated with a predefined false positive rate and an expected maximum number of feed inputs (i.e., interests and user IDs). The generated bloom filter has an actual false positive rate no larger than the predefined false positive rate if the number of actual feed inputs is no larger than the expected maximum number. Therefore, for the same number of feed inputs, in order to increase the false positive rate to protect the users' privacy, we can also reduce the expected maximum number of inputs. However, a larger false positive rate generates a higher probability of choosing some friends falsely regarded with many common interests and friends. Therefore, the answer quality is sacrificed to a certain extent but the personal information is better protected. In reality, the false positive rate needs to set according to the requirement of security and answer quality performance to break the tie.

4.1.2 Onion Routing based Answer Forwarding

Some questions, such as religious and political questions, may be sensitive to censorship, so that some askers and answerers may want to protect their identities from being exposed. SocialQ&A can leverage the onion routing technique [43] to provide user anonymity. At the initial stage, each user U_k generates a pair of public and private keys (denoted by Pri_{U_k} and Pub_{U_k}), and the public keys are exchanged between friends. To generate an onion routing path, a user randomly selects several users and form an encrypted routing path such as $Pub_{U_i}(U_j, Pub_{U_j}(U_k))$ for path $U_i \rightarrow U_j \rightarrow U_k$. For each relay user at the path, the remaining path is encrypted by its public key. The encrypted routing path is sent along each relay node. When U_i receives the encrypted routing path, it decrypts the path using its private key (Pri_{U_k}), and then sends $Pub_{U_j}(U_k)$ to U_j. Each receiver does the same operation to learn its successor. Therefore, each relay can only know its predecessor and successor.

However, we cannot directly adopt the onion routing technique in SocialQ&A. In the direct adoption, in order to protect its identity, an asker randomly selects several relay nodes among all users to form a path. A communication is established between any two consecutive relay nodes in the path to forward the question, and the final relay node searches the appropriate answerers as explained previously. The answerer will forward the answer along this established path from the final relay to the asker. However, such an established random path cannot ensure that the final relay user is in the same social community as the asker. For example, user U_i in the ECE department may select user U_k at the BIOE department as the last relay user to ask question "How to prepare qualify exams in ECE?". Due to the different communities, the potential answerers identified by U_k may not provide U_i a correct answer. Besides, it is easy for U_i to collude with its friends by selecting a particular malicious U_k to be the last relay. Then U_i and its malicious friend U_k can know the identity of the answerer when the answerer sends the answer back.

Therefore, instead of directly applying the onion routing for question forwarding, we apply the onion routing for the answer forwarding to protect the identities of the answerer and asker. In our onion routing based answer forwarding, the asker searches the answers using the previously introduced method. The asker builds another onion path leading to itself that consists of randomly selected relay nodes and first relay U_i (which is different from the question forwarding onion path). It forwards its question along with this established onion path and does not transmit its identity along the question forwarding path. The answerer builds an onion path starting from itself that consists of randomly

selected relay nodes and final relay U_j. Then, along this path, the answerer forwards its answer with the asker's onion path. The final relay U_j forwards the answer and the asker's onion path to U_i, and then the answer is forwarded backwards to the asker along the asker's onion path from U_i. Since any relay user is only aware of its predecessor and successor without knowing the message initiator or the entire path, the identities of the asker and answerer are protected.

4.2 Answer Retrieval for Recurrent Questions

A large amount of daily questions in a Q&A system usually are recurrent. For example, among 15% of English questions crawled from *Yahoo! Answers*, 25% questions are recurrent [44]. Therefore, we can save users' efforts and system resources to answer recurrent questions by providing satisfying answers of the former same questions in repository. In order to release the workload of the centralized server to search recurrent questions, each asker stores the former questions and their associated answers, and users depend on nearby users in the social network for searching the recurrent questions. A straightforward way to search a recurrent question of a newly asked question is to broadcast the question to all friends of the asker or question forwarder if TTL>0. However, it generates high network traffic among social friends and friends-of-friends, and high workload for similar question searching in inquired users. Therefore, we introduce our bloom filter based similar question searching method.

In this method, each user feeds his/her questions with satisfying answers into a bloom filter, denoted by B^q. Since the recurrent questions may not be exactly the same, the success rate to find the former similar question may not be high if we directly feed the whole new question into the bloom filter. To solve this problem, we feed all n-grams [45] of the new question into the bloom filter. The n-gram is a contiguous n words of a question. For example, "at Clemson" and "Clemson University" are two 2-grams of the question "Where is the football stadium at Clemson University?". In n-gram, n is an integer larger than zero. We use 2-gram in our implementation as an example due to its high accuracy to find the recurrent questions as shown in Section 5.3. That is, each asker feeds all 2-grams of all of his/her questions with satisfying answers into a bloom filter result. To find recurrent questions in a bloom filter, K hash values of all 2-grams of the new question are calculated and the corresponding entries in the bloom filter are checked.

Each user periodically broadcasts his/her bloom filter results B^q for his/her questions with satisfying answers. In the broadcasting, each bloom filter is propagated through the social links for TTL hops. Whenever a user U_k asks a question, before U_k launches question Q_i's forwarding process, U_k first looks over all bloom filter results received. For each bloom filter result, all n-grams of the newly asked question Q_i are checked in this bloom filter, and the owner of each bloom filter result is scored by the number of successfully found n-grams of Q_i. The asker then selects the top N users with the highest scores, and sends them the recurrent question searching request for Q_i. When a user, say U_i, receives the request, it then finds the question with the largest number of common n-grams with Q_i, and forwards the question associated with the satisfying answers back to the asker. If the answerer is satisfied by the asker, the question is solved; otherwise, the question-user mapper processes this newly asked question as shown in Algorithm 2.

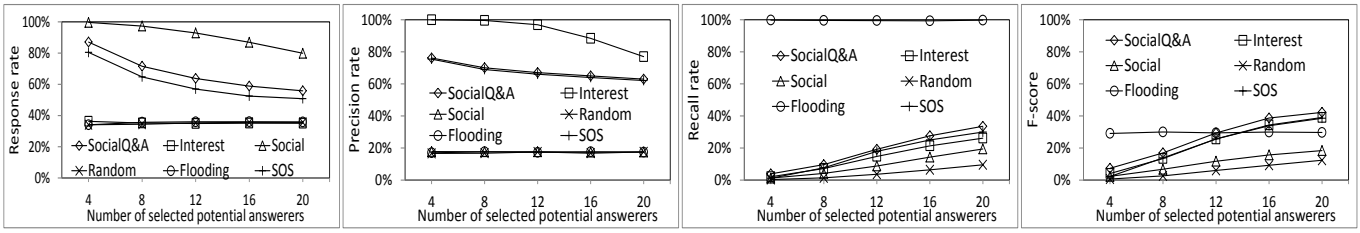
The onion routing based answer forwarding can also be applied in this answer retrieval process for recurrent questions. When a user U_j broadcasts his/her bloom filter results of questions with satisfying answers, U_j also associates it with an onion routing path p_a leading to itself. We use $U_{j'}$ to denote the last relay user of path p_a connecting to U_j. Asker U_i also uses a similar way in Section 4.1.2 to form an onion routing path p_q to protect its identity. We use $U_{i'}$ to denote the last relay user of path p_q. Asker U_i sends its newly asked question through p_q to the last relay $U_{i'}$, and $U_{i'}$ further forwards it through p_a to the last relay user $U_{j'}$. Relay user $U_{j'}$ forwards the question to user U_j, which then returns the answers of the matched question to $U_{j'}$. Then, $U_{j'}$ forwards the answer to $U_{i'}$, which then forwards the answer through p_q to the asker U_i.

5 Simulation Performance Evaluation

We conducted trace-driven experiments on the PlanetLab [46] to evaluate the performance of SocialQ&A in both searching effectiveness and efficiency for potential answerers. PlanetLab is a real worldwide testbed, consisting of 1295 nodes, on which you can deploy and test your network applications. We used the *Yahoo! Answers* question/answer trace data and Facebook user profile (interests and activities) trace from [39]. The Yahoo! Answers trace includes 9419 questions posted in the Entertainment & Music Movies section, and the Facebook trace includes 1000 users with profiles and friendship information. We tested 1000 users using randomly selected 200 PlanetLab nodes worldwide, each simulating 5 users. We randomly assigned users' profile and associated relationship in the trace to simulated users. Therefore, a social network is formed among the users based on the friendship relationship in the Facebook trace. From the question trace, we randomly selected 100 questions with keywords which can be mapped to the category of movies. We also used this set of questions to simulate those in each of other three interest categories. All questions were randomly assigned to users having the same interest. The TTL was set to 3, since according to [47], people are influenced by other people who are at most at 3-hop social distance. The rating range in *Yahoo! Answers* is [1, 5].

A successful response to a question includes answering or forwarding the question, in which if a question receiver has an answer to the received question, (s)he replies to it; otherwise, (s)he forwards the question. Intuitively, each potential answerer willing to answer the question should have at least one very high score for S, A or P in Equation 3. Thus, we give equal weights to all factors as $\gamma_S = \gamma_A = \gamma_P = 1$. The social closeness between friends ranges in [0,1.2]; if an asker's friend has social closeness larger than 0.6, (s)he is willing to respond to the asker's question. If we set this willingness threshold to be larger, there will be fewer successful responses in both our method and comparison methods, and vice versa. The probability that other friends respond to the question was randomly chosen from {10%, 20%, 30%}. The question query rate was set to one question per minute. These parameters are adjustable parameters and their changes will not affect the relative performance differences between the systems in comparison. The distribution of response time to a question follows the trace [39]. We use BA to denote the Best Answer set of a question existing in the system, and use RA to represent the Retrieved Answer set in the system. We define the *precision rate* as $|RA \bigcap BA|/|RA|$ to represent the received answers' quality, and define the *recall rate* as $|BA \bigcap RA|/|BA|$ to denote the received answers' completeness. We measured the overall effectiveness of our method using the F-score, which is calculated as $F_1 = \frac{2*precision*recall}{precision+recall}$.

Recall that SocialQ&A considers both interest similarity and social closeness. We compare SocialQ&A with i) So-



(a) Response rate (b) Precession rate (c) Recall rate (d) F-score

Fig. 4: Effectiveness of Q&A systems.

cialQ&A only considering interest similarity (denoted by Interest), ii) SocialQ&A only considering social closeness (denoted by Social), iii) random friend selection (denoted by Random), iv) flooding method selecting all friends [39] (denoted by Flooding), and v) SOS [39]. These systems were implemented in a distributed manner; that is, each node selects its friends to send/forward questions autonomously. Unless otherwise specified, β in Eq. (4) was set to 0.6. We first measured SocialQ&A's performance without the security and efficiency enhancement methods. We then compared the performance of SocialQ&A with and without the enhancement to measure its improvement.

5.1 Performance with Varying Number of Selected Answerers

We calculated the response rate as the number of all successful responses divided by the total number of question receivers. Figure 4(a) shows that the response rate of all systems versus the number of selected potential answerers, which are the top N friends to forward the question in Algorithm 2. It shows that the response rate follows Social>SocialQ&A>SOS>Interest≈Random≈Flooding. In SocialQ&A and Social, users choose friends with higher social closeness who are most willing to answer questions, so they have a higher response rate than others. SOS does not consider the potential willingness of friends with many common interests when calculating social closeness. Thus, its response rate is lower than SocialQ&A and Social, but higher than the other three methods without social closeness consideration. In SocialQ&A, users may choose friends with high interest similarity but lower social closeness. Thus, it generates a lower response rate than Social. We also see that the response rate of SocialQ&A, Social and SOS decreases as the number of selected answerers increases, since friends with lower social closeness are more likely to drop questions. This result implies that SocialQ&A's incentive works well when the set of answerers selected is small.

Figure 4(b) shows the average precision rate of each system, which follows Interest>SocialQ&A≈SOS>Social ≈Random≈Flooding. This is because Interest, SocialQ&A and SOS choose answerers with interest consideration, while Social, Random and Flood do not. By considering interest and social closeness simultaneously, SocialQ&A and SOS have lower precision rate than Interest, and their precision rates decrease as the number of selected answerers increases due to the same reason as in Figure 4(a). SocialQ&A and SOS both consider the interest similarity, so they produce similar precision rates. This implies that they have higher answer quality when the number of selected neighbors is small. Combining the results in Figures 4(a) and 4(b), we see that SocialQ&A performs the best regarding both response rate and answer quality.

Figure 4(c) shows the average recall rate of each system. We see that the recall rates of all systems except Flooding increase when the number of selected potential answerers increases. As more potential answerers are selected, more answers are provided, which increases the probability of receiving the best answers. We also see that the recall rate follows Flooding>SocialQ&A>SOS> Interest>Social>Random. Flooding sends a question to all friends, thus produces the highest recall rate. Since SocialQ&A and SOS consider both interest and willingness to respond, they produce much more high-quality answers than in other systems except Flooding. SocialQ&A has a higher recall rate than SOS due to its higher response rate than SOS as shown in Figure 4(a). We see that Interest has a larger recall rate than Social, especially when the number of selected neighbors is large. This is because Social ensures high willingness to respond but cannot guarantee the answer quality, while Interest provides high answer quality. As Random does not consider interest and social closeness, it generates the lowest recall rate. This figure indicates that SocialQ&A can recall more answers than other systems without flooding questions.

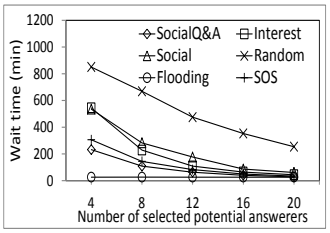


Fig. 5: Wait time.

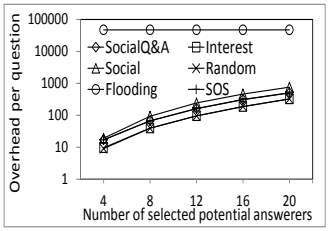
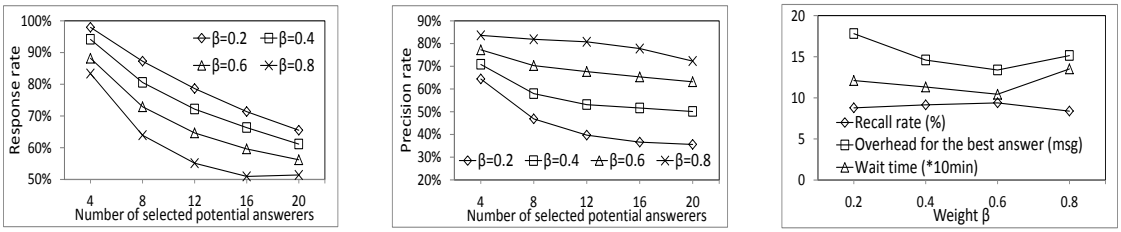


Fig. 6: System overhead.

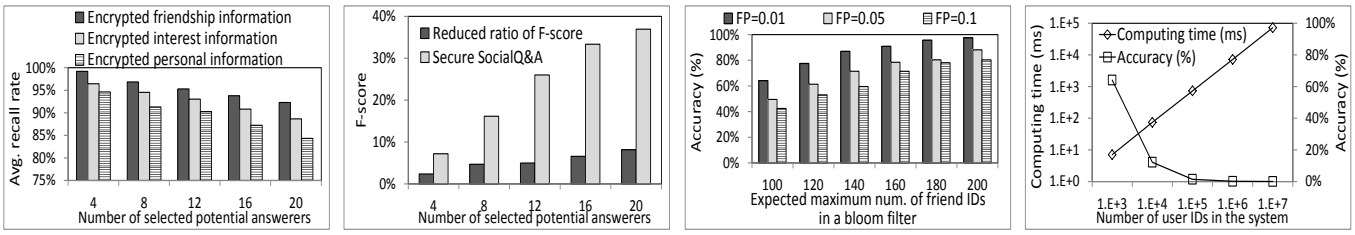
Figure 4(d) shows the F-score of each system, which represents the overall accuracy of each system's potential answerer searching method. From the figure, we can see that the F-score follows SocialQ&A>SOS≈Interest>Social >Ran- dom. That is because SocialQ&A, SOS and Interest have larger precision and recall rates than Social and than Random as shown in Figures 4(b) and 4(c). Due to the larger deviation between precision and recall rates of Interest than of SocialQ&A, SocialQ&A has a larger F-score than Interest. SocialQ&A has a larger F-score than SOS, due to its larger recall rate. Thus, SocialQ&A has the highest F-score. Also, because SOS has a recall rate larger than Interest, but not larger enough as SocialQ&A does, SOS and Interest have a similar F-score. In all, the figure indicates that SocialQ&A achieves a better overall searching accuracy than all other systems by considering both precision and recall rates.

We define the *wait time* of a question as the time interval between the time when a question is asked and the time when the first best answer of this question is received. Figure 5 shows the average wait time for all questions. It follows Flooding<SocialQ&A<SOS<Interest<Social <Random and all methods have shorter response time when there are more answerers selected due to the same reason as in Figure 4(c). This figure indicates that SocialQ&A leads to shorter wait time for answers than other methods except Flooding. However, Flooding generates low precision rate and also high overhead which will be shown in Figure 6.



(a) Response rate (b) Precision rate (c) Effectiveness

Fig. 7: Performance vs. varying β values.



(a) Recall rate (b) F-score (a) Varying the expected maximum num. of friend IDs in a bloom filter (b) Varying number of users

Fig. 8: Accuracy of the bloom filter based personal information exchange method.

Fig. 9: Performance of privacy protection.

We define the *overhead* of a question as the number of forwarding messages generated for the question. Figure 6 shows the overhead per question. It follows Social>SocialQ&A>SOS>Interest≈Random due to the same reason as in Figure 4(a). A higher response rate leads to more forwarding messages. SOS, Interest and Random produce lower overhead because many message receivers do not respond. Flooding generates the highest overhead which is nearly constant because of its broadcasting feature. We also see that the overhead increases as the number of selected answerers increases in other methods as more queries are sent or forwarded.This figure indicates that by more accurately routing a question to users that are capable and willing to answer the question, SocialQ&A generates relatively low overhead and high response rate.

5.2 Performance with Varying Weight Values

Figure 7(a) shows the response rate of different β values versus different numbers of selected answerers. It shows that the response rate decreases when β increases. Because a larger β gives less weight on the social closeness, which reflects the users' willingness to respond to a question. Figure 7(b) shows the precision rate of different β values versus different numbers of selected answerers. It shows that the precision rate increases as β increases because a larger β gives more weight on the interest similarity, which reflects users' capability to answer questions. The results show that β controls the tradeoff between the response rate and precision rate.

We define *overhead for the best answer* as the the number of forwarding message generated for a question before the first best answer is received. Figure 7(c) shows the recall rate, wait time and overhead for the best answer when 8 potential answerers were selected in SocialQ&A. As β increases, the wait time and overhead first decrease then increase, and the recall rate first increases and then decreases at the point of $\beta = 0.6$. This is caused by the increasing weight of interest closeness and decreasing weight of social closeness. This figure shows that when β equals 0.6, SocialQ&A has the shortest wait time, largest recall rate and lowest overhead, thus reaching the optimal tradeoff between efficiency, effectiveness and cost.

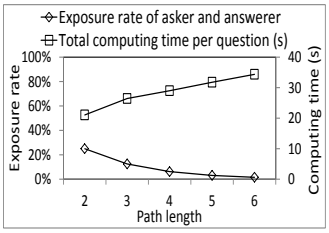


Fig. 10: Security and overhead of the onion routing based answer forwarding.

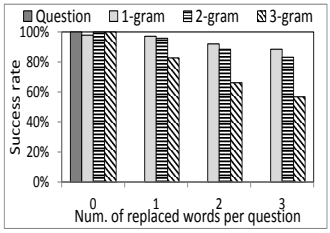


Fig. 11: Success rate of answer retrieval for recurrent questions.

5.3 Performance of Security and Efficiency Enhancement

In this section, we measure the performance of our proposed security and efficiency enhancement methods. For the counting bloom filter [42] adopted in the evaluation, we set the false positive rate to 1% and the expected maximum number of inserted information items to 100. Unless otherwise specified, all settings are set as the default. Using the bloom filter based personal information exchange method, the interest similarity and social closeness are calculated differently from the original method in Section 4.1.1. Then, this enhancement method may select different potential answerers. In order to show the accuracy of identifying appropriate answerers, we measure the recall rate of a question as $\frac{|X' \cap X|}{|X|}$, where X and X' denote the set of answerers selected for a question with the enhancement method and with the original method introduced in Section 4.1.1, respectively.

Figure 8(a) shows the average recall rate of all questions using the bloom filter based personal information exchange method that encrypts only the friend information, only the interest information, and both friend and interest information (i.e., personal information). If the fridendship/interest is encrypted by a bloom filter, we use formulas in Section 4.1.1 to calculate social closeness/interest similarity; otherwise we use the formulas in Section 3.2.3. The figure shows that this method cannot recall all potential answerers generated by the original method. This is because the false positive rate of the bloom filter and the cosine similarity calculation increase some users' social closeness/interest similarity scores by falsely

regarding some uncommon friends/interests as common friends/interests. But this enhancement method with encrypted friendship/interest/personal information still can recall at least 92.3%/88.7%/84.3% of the potential answerers found by the original method when the number of selected potential answerers no larger than 20. The figure also shows that the recall rate decreases when the number of selected potential answerers increases. Recall that all friends are ranked in a descending order of Ψ_{U_k} to be selected as a potential answerer. By selecting more potential answers, a larger percentage of falsely ranked friends will be selected, which leads to lower recall rate. We then test the overall accuracy of answerer searching, measured by F-score, of SocialQ&A with this enhancement method (denoted by *Secure SocialQ&A*) compared to SocialQ&A without this enhancement method (denoted by SocialQ&A). Figure 8(b) shows the F-score of *Secure SocialQ&A* and its reduced ratio compared to F-score of SocialQ&A. The reduced ratio of F-score is calculated by the $\frac{F-F'}{F}$, where F and F' represent the F-scores of SocialQ&A and *Secure SocialQ&A*, respectively. Due to the same reasons as Figure 8(a), it generates a slightly smaller F-score than SocialQ&A, and the reduced ratio is at most 8%. Figures 8(a) and 8(b) indicate that the bloom filter based personal information exchange method can provide user privacy protection with a little sacrifice on answer quality.

In the bloom filter based personal information exchange method, users exchange bloom filter results of friendship and interest information in order to protect their privacy. However, a malicious user can detect the friendship or interest information by looping all user IDs and interests. In this experiment, we assume there is a user with 200 friends, and then loop all user IDs to calculate the accuracy of detected friends of the user, which is calculated by the ratio of the actual number of friends (i.e., 200) over the number of found users contained in the bloom filter result. We evaluate the accuracy of the personal information exchange method with different predefined false positive rates and expected maximum number of inserted elements. Below, we show the performance of user friendship protection, and the performance of user interest protection is similar as the friendship protection. Figure 9(a) shows the accuracy of found friends of this user versus the expected maximum number of inserted elements (i.e., friends) in the bloom filters with different false positive rates, denoted by FP. It shows that the accuracy increases when the expected maximum number of inserted elements in the bloom filter increases and vice versa. It also shows for the same expected maximum number of inserted elements, a larger false positive rate leads to a lower accuracy. This is because a larger expected maximum number leads to a smaller false positive rate. With a smaller false positive rate, fewer strangers are found as friends contained in the bloom filter, which leads to a higher accuracy. The figure indicates that to protect user privacy, we can decrease the accuracy by decreasing the expected maximum number of input elements and increasing the pre-defined false positive rate in the bloom filter. Figure 9(b) shows the accuracy versus the total of user IDs in the system, which is increased from 1,000 to 10,000,000 enlarged by 10 times at each step. It shows that the accuracy decreases as the number of user IDs increases. This is because with the same false positive rate, looping a larger number of user IDs leads to more found users contained in the bloom filter, so that the accuracy decreases. The figure also shows the computing time (ms) to detect all users contained in the bloom filter. It shows that due to the larger number of user IDs in the system, it takes a longer time to find all users contained in the bloom filter result. The figure indicates that a larger user ID space and sparse user ID distribution can also help protect the user's friendship privacy by decreasing the accuracy. Figures 9(a) and 9(b) confirm that a relatively small expected maximum number of inserted information, a lower false positive rate in a bloom filter or a large user ID range in the system can improve the performance of user privacy protection; and both Figures 8 and 9 indicate that the encrypted friendship information can protect users' friendship privacy at a little cost of sacrificing answer quality.

We then measure the performance of the onion routing based answer forwarding method in protecting the identities of answerers and askers and its system overhead. In this experiment, we assume that 50% of users are malicious users, and the identities of answerers or askers are exposed if all relay users in the whole forwarding path are malicious users. This is because the malicious users form all relay users of this onion routing path and the path length of the onion routing is constant, so that they can know the whole path and the first (last) relay user knows that its predecessor (successor) is the answerer (asker). We define the exposure rate as the number of total identified askers and answerers by malicious users over the total number of asker and answerer. Figure 10 shows the exposure rate versus the path length of the onion routing. It shows that the expose rate decreases as the path length of the onion routing increases. This is because for a longer path length, the probability that all relay users are malicious users is lower. It indicates that the onion routing based answer forwarding can better protect the asker/answerer identities when the path length is longer. Figure 10 also shows the total computing time of all users in an onion path for answer forwarding per question. It shows that the computing time increases as the path length increases. This is because each secure communication between two relay users in the path involves data encryption and decryption. Thus, the figure is a showcase to help determine an appropriate path length in reality, that is, we should consider both the identity protection requirement and the system overhead for each user.

Next, we measure the performance of the answer retrieval method for recurrent questions. In this experiment, each user's bloom filter result of the successfully answered questions is broadcasted through social links within three hops, and top 2 users with the highest scores of the bloom filter result matching are selected to send the recurrent question searching request. In reality, the newly asked question may not be the same as the former one. Thus, to generate a newly asked question, instead of using the exact recurrent question, we replaced m number of words in the question with m randomly selected other words among all words of all questions, where m is increased from 0 to 3 with a step size of 1. We measure the success rate as the percentage of questions, which are resolved by satisfying answers fetched by the answer retrieval method. We used *Question* to denote the method using the whole newly posted question to match former questions for answer retrieval, and use n-gram to denote the n-gram based answer retrieval. Figure 11 shows the success rate of answer retrieval methods with different n-grams versus the number of changed words in each question. It shows that n-grams can successfully retrieve the answers for most of the recurrent questions. Among them, the 1-gram has a better performance than the others when the asked question is not exactly same as its former one. This is because when matching a new question to a former question, 1-gram produces more the same grams than other n-grams, which makes the former question have a higher

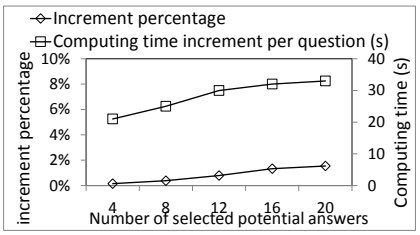


Fig. 12: The Waiting time with Security Enhancement.

probability to be selected as the recurrent question. However, when the newly asked question is the exactly same as the former asked question, 1-gram may not accurately find the recurrent question. For example, the question "where is Clemson University?" and "where is Clemson post office?" containing all words in the newly asked question "where is Clemson?" and will be returned in 1-gram searching.

As shown in the figure, *Question*, 2-gram and 3-gram fetch all recurrent questions when the newly asked question is exactly the same as its former asked question. 2-gram has a comparable performance with 1-gram when the newly asked question is different from the former one, while *Question* has a success rate equal to zero, and 3-gram generates a much lower success rate than 1-gram and 2-gram do. The figure indicates that the n-gram based answer retrieval method can successfully retrieve the answers for recurrent questions, and the 2-gram has an overall better performance than other n-grams. In reality, the optimal n-gram can be selected by using sample recurrent questions from the system.

Figure 12 shows the average waiting time for all questions. It follows Social Q&A with Security protection (SocialQ&A Enc)≈Social Q&A due to the same reason as in Figure 4(c). This figure indicates that, SocialQ&A with security enhancements takes additional several seconds computing time on each question average. But, the Social Q&A Enc can still get similar average waiting time per question as Social Q&A. That is because the computing time is negligible compared to the waiting time. The Social Q&A with security protection has similar performance of efficiency and better performance of user privacy protection.

6 Simulation Experimental Results

Section 5 shows the outperformance of SocialQ&A compared with other systems by measuring its potential answerer searching effectiveness and efficiency. In this section, we show the performance of SocialQ&A under real environment. This section presents analysis on the usage of our deployed real-world SocialQ&A system over a period of approximately one month starting at the beginning of March, 2012. Since it is not a mature and commercial software, and it is hard to attract users to use it, we call for volunteers to use it within this month. A total of 124 users registered and used SocialQ&A, 163 questions were posted and 282 answers were posted in response. In the experiment conduction, we ask users to follow their answers and ask questions behaviors (the willingness and interest and frequency) in their daily usage of Yahoo! Answer so that we can compare the system performance with Yahoo! Answer to certain extent.

The distribution of the users. Approximately 35 users were from the United States, 70 users were from India, and 1 user was from the United Kingdom. This small-scale testing shows the potential benefits in both the searching effectiveness and efficiency and interesting usages of SocialQ&A to a certain degree. Evaluation on a large-scale user base remains as our future work.

6.1 Prototype Implementation of SocialQ&A

SocialQ&A allows users to register and modify user information, add/remove friends, ask/answer/forward questions and check question notifications. Consider a hypothetical user named Mike. When Mike registers, he is required to provide essential information about himself, such as his personal information, area of study/expertise, his current interests, and his involvement in other activities. Users are also encouraged to describe their interests in terms of a few predefined categories, such as movies, books, television, music. *User Interest Analyzer* uses the registration information to determine Mike's interests.

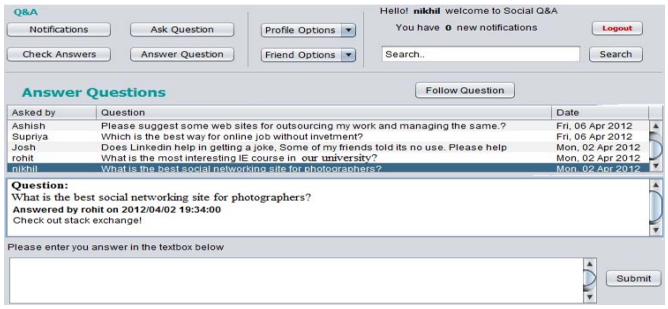


Fig. 13: Interface of a question and answer thread.

As shown in Figure 13, the main user interface includes a social platform (including Profile and Friend Options), a Q&A domain (including Ask Question, Answer Questions and Check Answers), and a notification module (including events from both social platform and Q&A domain). Whenever there is a user profile or friendship change of Mike, the interest similarity S and friend closeness C are updated accordingly by *Question User Mapper*.

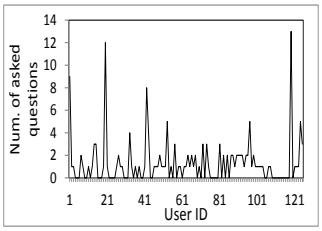
When Mike submits questions, the user interface contacts *Question Categorizer* first, and then passes the question interests to *Question User Mapper*, which finally decides the top N answerers and sends out the questions. Another feature provided by SocialQ&A is the option to forward and follow questions so that the forwarder will also receive answers from the answerers. Further, SocialQ&A provides search function to enable users to search in the repository of previous posted questions and answers.

Unlike previous Q&A approaches, SocialQ&A exploits the users' profile information and interests, in addition to the user's social network and Q&A activities to determine potential answer providers. Additionally, the interest information of all users in the system is continuously updated based on their actions. SocialQ&A is also distinguished by routing questions only to potential answer providers rather than flooding to all friends, thereby reducing overhead and frustration to users.

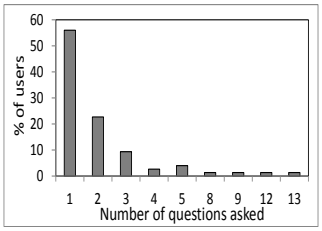
6.2 User Questioning and Answering Activity

We used the number of questions and answers posted to characterize user activity. Out of 124 users, 75 unique users posted at least one question; 81 unique users provided at least one answer; 26 users (approximately 20%) did not post or answer any questions. The remaining 80% contributed actively to SocialQ&A.

Figure 14(a) shows the number of questions asked by each user, ranging from 0 to a maximum of 10. Figure 14(b) displays the percentage of users who asked a given number of questions. As seen from the figures, approximately 56% of the users asked just 1 question, approximately 23% of the users asked 2 questions, approximately 10% of the users asked 3 questions, and the remaining 11% asked more than 3 questions. Thus, most of the users were fairly active.

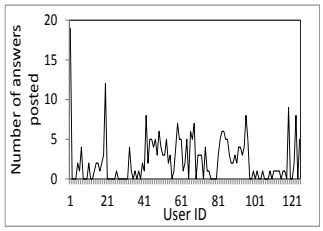


(a) # of questions of each user

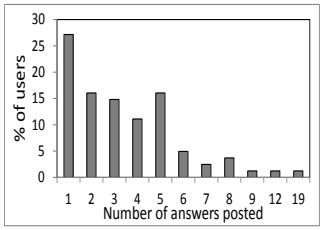


(b) % of users vs. # of questions asked

Fig. 14: Analysis on asked questions.



(a) # of answers from each user



(b) % of users vs. # of answers posted

Fig. 15: Analysis on posted answers.

Figure 15(a) shows the number of answers posted by each user, indicating the answering activity of the users. On average, users posted 2 to 3 answers. There are some users that were extremely active and posted 5 or more answers, and one of the users posted a total of 19 answers. Figure 15(b) shows the number of answers posted versus the percentage of users. Approximately, 25% of the users provided just a single response, 15% of the users provided 2 answers, 15% of the users provided 3 answers, 10% of the users provided 4 answers, and 40% of the users provided 4 or more answers. Comparing Figure 15(b) with Figure 14(b), we see that the users tend to answer questions more actively than they asked questions.

In the test, a total of 24 out of 163 questions (around 15%) remain unanswered, while all other questions have at least one response. As SocialQ&A identifies potential answer providers who have more common interests, close social relationships with the asker, and have interest in the question's category, those question receivers are more likely to answer the question. Thus, SocialQ&A is able to achieve an improvement even with a very limited number of users. We expect that the number of unanswered questions tends to reduce with user growth, because with more users, the range of expertise also becomes broader, a user has more friends to ask questions and more users are willing to answer questions. Practically, we were not able to test SocialQ&A with millions of users. However, current results indicate the promises of SocialQ&A in improving current Q&A systems.

Potential benefit: The questions in SocialQ&A are more likely to be answered since the potential answer providers have a close social relationship with the asker and have an interest in the question category, as indicated in [14–16].

6.3 Analysis of Questions

In this section, we analyze the questions asked in SocialQ&A from the aspects below: (1) question paraphrasing, (2) question categories, (3) question types, and (4) the number of answers received for each question. To determine the question types, categories and subcategories to which the question belongs, we manually examined every question.

Why do people like bass low frequencies in music?	I want to start photography can anyone suggest me a good camera?
What are the effects of music on plants?	What is a good car renting service in the washington DC area?
Please can anybody suggest me the book for multimedia Systems?	What is better programming language PHP or python?
What is RTP?	skype or yahoo which is better....
Best movies of 2012...	How do I make my playlist private on youtube?
Best character in The Lord Of the rings...???	What is a computer virus?
What is the best comedy show on TV right now?	What is our purpose of living life?
Who is the best actor in big bang	Why am I so happy today?

Fig. 16: A sample of questions from SocialQ&A.

After analyzing the 163 total questions, we found that the average number of characters per question is 45.5 (10.65 words). The majority of questions (91%) are comprised of a single sentence. Approximately, 75% of the questions were properly paraphrased with a question mark, although some questions contained multiple question marks.

Recall that SocialQ&A uses four major categories: music, books, movies, and television. The left column in Figure 16 shows two example questions for each category. Figure 17 shows the distribution of questions among the four major categories. Approximately, 38% of the questions are in music, 29% are in books, 41% are in movies, and 13% are in television.

Figure 18 shows the distribution of questions among the various 32 subcategories. We show top 10 subcategories in the figure, which indicate the interests of the current users.

We further classified the questions based on the following four question types:
(1) *Recommendation*: Questions like "Please recommend some local places for food."
(2) *Opinion*: Questions like "What is a better programming language, PHP or Python?"
(3) *Factual*: Questions like "What is the capital of Oregon?"
(4) *Rhetorical*: Questions like "What is the aim of life?"

The right column in Figure 16 shows two example questions for each question type. Figure 19 shows the distribution of questions based on their types. We see that the users asked a large number of opinion-type questions. Approximately, 20% of the questions were recommendation type questions, 36% were opinion-type questions, 25% were factual-type questions, and 19% were rhetorical-type questions.

Figure 20(a) shows the number of answers posted for each of the questions with at least one response. Figure 20(b) shows the distribution of the number of responses for these questions. Approximately, 47% of questions have just 1 answer, and 13% of questions have more than 4 answers. Most of the questions receiving only one answer are factual questions, since one answer is sufficed for such questions. Opinion-type questions tend to have more responses, as no answer is the final answer. For example, in the test, question "Should I buy a Windows laptop or MacBook?" received more answers than question "What is the capital of Oregon?".

Potential benefit: SocialQ&A provides a platform for both factual and non-factual questioning, and there tend to be more answers for opinion based questions from social friends, which can be a better reference for the askers on non-factual questions. This feature of social network based Q&A systems is also indicated in [1, 2].

6.4 Quality of Answers

For every question asked, the asker was able to rate the answer on a scale of 1 to 10. Out of 282 answers posted, the users of SocialQ&A rated 233 answers. A single question may have multiple answers; hence, we calculated the average rating for each question and present the results in Figure 21(a). The average rating of all answers is 8.675, ignoring

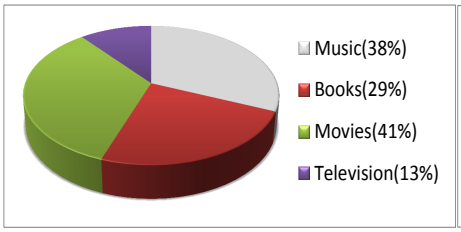


Fig. 17: Distribution of questions among the major categories.

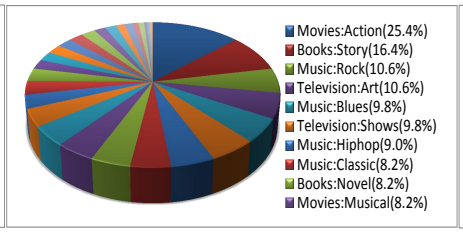


Fig. 18: Distribution of questions among various subcategories.

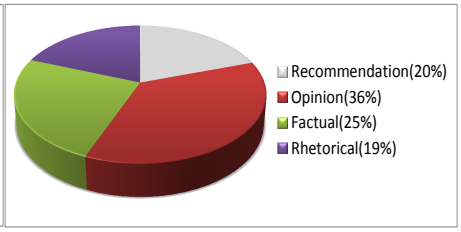
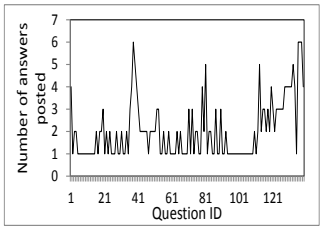
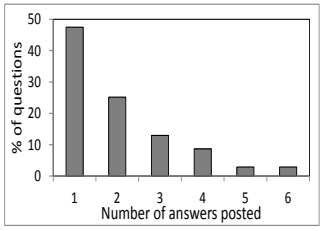


Fig. 19: Distribution of questions based on question types.

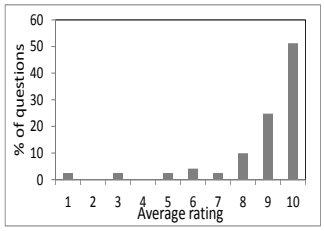


(a) # of answers received by each question

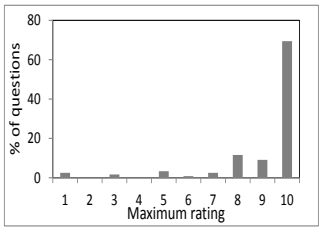


(b) % of questions vs. # of answers received

Fig. 20: Analysis on the # of received answers.



(a) % of questions vs. the average rating



(b) % of questions vs. the maximum rating

Fig. 21: Analysis on the answer rating.

those that were not rated. The median is 9.29, the minimum is 1, and the maximum is 10. The result means that most answers provided in this test received high ratings.

We also analyzed the correlation between the question length and the question rating. Intuitively, long questions tend to be easier to understand. Thus, long questions help the answer provider determine what the asker is looking for, enabling him/her to provide a more accurate answer. Any question that was explained using more than one sentence is considered a long question. Our results show that longer questions have an average rating of 9.33, which is higher than the overall average rating.

Another way to examine the answer quality is to find the maximum rating that an answer received for a particular question. The analysis of the maximum rating is meaningful because the highest rated answer provides the asker with the desired information and the other answers could be neglected. Figure 21(b) plots the percent of questions versus the maximum rating of each question. The average maximum rating over all questions is 9.05, the median is 10, the minimum is 1, and the maximum is 10. The results indicate that SocialQ&A provides satisfactory answers in most cases in this test.

The high answer ratings in SocialQ&A may be attributed to two factors: (1) since the answerer belongs to the asker's immediate social network, (s)he is highly motivated to provide better quality answers, and (2) the question is mapped to the potential answer provider whose interests most closely match the topics of the question. The result of this analysis verifies the advantages of SocialQ&A by leveraging the previous studies [14–17] on the influence of social networks on Q&A performance to effectively identify potential answer providers that can provide high-quality answers. We expect that the answer quality would be further improved as more users join SocialQ&A, because more users will be willing to respond and the probability that an expert exists among users also increases.

We further analyzed the answer quality based on the aforementioned types of questions and found that:

(1) The average rating per factual question is 9.14.
(2) The average rating per opinion-type question is 8.67.
(3) The average rating per suggestion-type question is 8.18.
(4) The average rating per rhetorical-type question is 8.95.

The observations indicate that factual questions have a higher average rating per question, most likely because such questions can only have one correct answer. The answer quality for rhetorical questions is determined solely by the asker's perception. Also, the opinion-type questions have a higher average rating than the suggestion-type questions. This is because when asking an opinion-type question, the user typically asks for a choice between 2-4 items that (s)he has shortlisted, whereas suggestion-type questions typically have a wider range of options.

Potential benefit: SocialQ&A can enhance the degree of satisfaction of askers on the answers especially for non-factual questions since answerers share interests with askers and are motivated to answer their questions, as indicated in [2, 14, 16, 17].

6.5 Wait Time for Answers

Figure 22 plots the distribution of wait time for an asker to receive a response to his/her question. We see that a large percentage of questions (around 50%) are answered within 8 minutes. These results are promising and show signs of future improvement on current Q&A systems. By considering the social closeness, SocialQ&A can more accurately identify potential answer providers that are willing to answer the questions within a short time period. We also see that 15% of the questions in SocialQ&A are answered after a time period of one day for two reasons. First, due to the limited number of users in the system, sometimes the answer providers to whom the question was forwarded were not online, leaving that question unanswered until those users log in again. Second, because the number of users in the system was very small, it may not be easy to find enough potential answers who are capable to answer the question. We expect that more users will help reduce the wait time because the number of users willing to answer questions quickly increases and the number of users having expertise on the question's topics also increases.

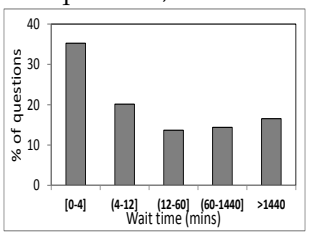


Fig. 22: % of resolved questions with different wait times.

We also analyzed the wait time of answers based on the four types of questions and found that:

(1) Most of the factual questions (around 80%) were answered within an average of 16.1 minutes.

(2) Most of the opinion-type questions (around 70%) were answered within an average of 59.87 minutes.
(3) Most of the suggestion-type questions (around 70%) were answered within an average of 71.62 minutes.
(4) Most of the rhetorical-type questions (around 70%) were answered within an average of 123.83 minutes.

From these results, we conclude that the reason for late responses regarding the rhetorical questions is the nature of the questions; conversely, factual questions receive responses faster because the answers are well established. Also, as mentioned previously, the asker generally narrows down the choices for opinion-type questions; hence, they are answered faster than the closely related suggestion-type questions.

Potential benefit: SocialQ&A reduces the wait time of answers because as the questions are mapped to the asker's close friends, they tend to respond quickly due to the close social relationship and their expertise/interest on the questions, as indicated in [16, 14].

6.6 Q&A Activity Examples

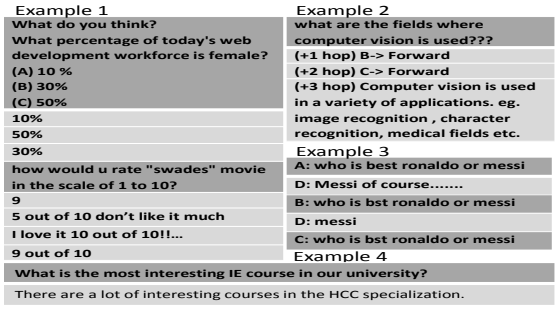


Fig. 23: Sample questions and answers from SocialQ&A.

In this section, we take a close view on users' behavior in SocialQ&A. Figure 23 shows Q&A activity of SocialQ&A users on four question examples. In Example 1, it is interesting to see users took SocialQ&A as a polling tool; they wanted statistical results for references. In the second question, a user asked the opinion about an India movie, and quickly received answers from his/her India friends, who have the same background as him/her. This supports that users with similar interests and high social closeness tend to answer questions. In our data trace, there are many technical questions and discussion in the computer science area, which makes SocialQ&A as a forum. This also indicates that communities exist in the SocialQ&A system and it is necessary to consider both interest and social closeness in potential answerer selection.

In Example 2, a user wanted to know where computer vision is used. The question cannot be answered by his/her friend B, who forwarded the question to C, who further forwarded the question. Finally, a user in 3-hop distance returned the asker an answer. This example shows that the multi-hop question forwarding in SocialQ&A is indeed useful. Sometimes, users ask questions with topics beyond his/her communities, which may be quickly resolved by routing to another community through social links.

In Example 3, users A, B and C asked the same question. User D answered twice, but did not answer the third time. Since (s)he already answered the question, (s)he wanted to see others' answers by acting as a follower (shown in our system). So, a form of "follow" functionality in QuestionQ&A is necessary, which allows users to receive answers for others' questions they are interested in.

One of the most interesting features of SocialQ&A is that it allows askers to receive answers hypercustomized to their information need. In Example 4, a user answered a question particularly related to his university. It is difficult to find answers through traditional search engines for such non-factual questions for a particular user in a particular place or environment. SocialQ&A aims to meet the need of these questions.

In a nutshell, SocialQ&A allows users not only to ask factual and non-factual questions but also to conduct Q&A through different formats (e.g., polling, forum, chatting). Also, users can leverage the follow and forward functionality to gain knowledge beyond their communities or propagate their thoughts to a larger group of users.

7 Conclusion

Q&A systems are used by many people for purposes such as information retrieval, academic assistance, and discussion. To increase the quality of answers received and decrease the wait time for answers, we have developed and prototyped an online social network based Q&A system, called SocialQ&A. It utilizes the properties of a social network to forward a question to potential answer providers, ensuring that a given question receives a high-quality answer in a short period of time. It removes the burden from answer providers by directly delivering them the questions they might be interested in, as opposed to requiring answer providers to search through a large collection of questions as in *Yahoo! Answers* or flooding a question to all of an asker's friends in an online social network. The bloom filter based enhancement methods encrypt the interest and friendship information exchanged between users to protect user privacy, and record all n-grams of answered questions to automatically retrieve answers for recurrent question. The onion routing based answer forwarding protects the identities of askers and answers. Our comprehensive trace-driven experiments and analysis results on the real-world Q&A activities from the SocialQ&A prototype show the promises of SocialQ&A to enhance answer quality and reduce answer wait time in current Q&A systems, and demonstrate the secure and efficiency improvement achieved by the enhancements. Since same questions may be presented very differently and the same question may be answered differently in different situation. In the future, we will cooperate with other techniques (e.g. topic modeling [48] and word embedding [49]) into SocialQ&A to find the redundant question with a large scale user set. Due to the dynamic of user behavior, SocialQ&A can cooperate a machine learning method to adjust three parameters appropriately, which needs a large user base and much more usage. We will conduct tests on a large user base in the real-world experiment.

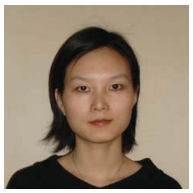
Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006 and Microsoft Research Faculty Fellowship 8300751. An early version of this work was presented in the Proc. of ICCCN'15 [50].

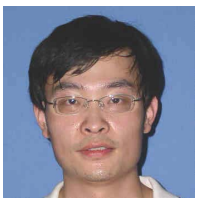
References

[1] M. R. Morris, J. Teevan, and K. Panovich. A Comparison of Information Seeking Using Search Engines and Social Networks. In *In Proc. of ICWSM*, 2010.
[2] M. R. Morris, J. Teevan, and K. Panovich. What do People Ask Their Social Networks, and Why?: A Survey Study of Status Message Q&A Behavior. In *Proc. of CHI*, 2010.
[3] Z. Gyongyi, G. Koutrika, J. Pedersen, and H. Garcia-Molina. Questioning Yahoo! Answers. In *Proc. of QAWeb*, 2008.
[4] Yahoo!Answers Team. Yahoo! Answers BLOG. http://yahooanswers.tumblr.com/, [Accessed on 10/20/2014].
[5] B. Li and I. King. Routing Questions to Appropriate Answerers in Community Question Answering Services. In *Proc. of CIKM*, 2010.

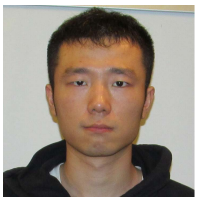
[6] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge Sharing and Yahoo Answers: Everyone Knows Something. In *Proc. of WWW*, 2008.
[7] G. Drosatos, P. Efraimidis, A. Arampatzis, G. Stamatelatos, and I. Athanasiadis. Pythia: A privacy-enhanced personalized contextual suggestion system for tourism. In *COMPSAC*, 2015.
[8] S. Li, Q. Jin, X. Jiang, and J. Park. *Frontier and Future Development of Information Technology in Medicine and Education: ITME 2013*. Springer Science & Business Media, 2013.
[9] A. Mtibaa, M. May, C. Diot, and M. Ammar. Peoplerank: Social Opportunistic Forwarding. In *Proc. of Infocom*, 2010.
[10] E. Pennisi. How Did Cooperative Behavior Evolve? *Science*, 2005.
[11] H. Shen, Z. Li, G. Liu, and J. Li. Sos: A distributed mobile q&a systembased on social networks. *TPDS*, 2014.
[12] A. Spagnolli and L. Gamberini. Interacting via sms: Practices of social closeness and reciprocation. *British Journal of Social Psychology*, 2007.
[13] M. L. Radford, C. Shah, L. Mon, and R. Gazan. Stepping Stones to Synergy: Social Q&A and Virtual Reference. *Proceedings of the American Society for Information Science and Technology*, 2011.
[14] M. Richardson and R. White. Supporting Synchronous Social Q&A Throughout the Question Lifecycle. In *Proc. of WWW*, 2011.
[15] R. W. White, M. Richardson, and Y Liu. Effects of Community Size and Contact Rate in Synchronous Social Q&A. In *Proc. of SIGCHI*, 2011.
[16] J. Teevan, M.R. Morris, and K Panovich. Factors Affecting Response Quantity, Quality, and Speed for Questions Asked via Social Network Status Messages. In *Proc. of ICWSM*, 2011.
[17] Z. Li and H. Shen. Collective Intelligence in the Online Social Network of Yahoo!Answers and Its Implications. In *Proc. of CIKM*, 2012.
[18] J. Bian, Y. Yang, and T. Chua. Predicting trending messages and diffusion participants in microblogging network. In *Proc. of SIGIR*, 2014.
[19] X. Geng, H. Zhang, Z. Song, Y. Yang, H. Luan, and T. Chua. One of a kind: User profiling by social curation. In *Proc. of Multimedia*, 2014.
[20] Z. Yang, J. Xue, C. Wilson, B. Y. Zhao, and Y. Dai. Uncovering user interaction dynamics in online social networks. In *Proc. of ICWSM*, 2015.
[21] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise Networks in Online Communities: Structure and Algorithms. In *Proc. of WWW*, 2007.
[22] 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 *Proc. of WWW*, 2009.
[23] P. Jurczyk and E. Agichtein. Discovering Authorities in Question Answer Communities by Using Link Analysis. In *Proc. of CIKM*, 2007.
[24] M. Bouguessa, B. Dumoulin, and S. Wang. Identifying Authoritative Actors in Question-Answering Forums: the Case of Yahoo!Answers. In *Proc. of KDD*, 2008.
[25] L. Hong, Z. Yang, and B. D. Davison. Incorporating Participant Reputation in Community-Driven Question Answering Systems. In *Proc. of CSE*, 2009.
[26] W. Chen, Q. Zeng, W. Liu, and T. Hao. A User Reputation Model for a User-Interactive Question Answering System: Research Articles. *Concurrency and Computation: Practice and Experience*, 2007.
[27] Y. R. Tausczik and J. W. Pennebaker. Predicting the Perceived Quality of Online Mathematics Contributions from Users' Reputations. In *Proc. of SIGCHI*, 2011.
[28] F. Liu, Y. Sun, B. Li, B. Li, and X. Zhang. FS2You: Peer-Assisted Semi-Persistent Online Hosting at a Large Scale. *TPDS*, 2010.
[29] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor. Learning From the Past: Answering New Questions With Past Answers. In *Proc. of WWW*, 2012.
[30] X. Quan, W. Liu, and B. Qiu. Term Weighting Schemes for Question Categorization. *TPAMI*, 2011.
[31] W. Song, W. Liu, N. Gu, X. Quan, and T. Hao. Automatic Categorization of Questions for User-Interactive Question Answering. *Information Processing and Management*, 2011.
[32] B. Li, I. King, and M. R. Lyu. Question Routing in Community Question Answering: Putting Category in its Place. In *Proc. of CIKM*, 2011.
[33] T. C. Zhou, M. R. Lyu, and I. King. A Classification-Based Approach to Question Routing in Community Question Answering. In *Proc. of WWW (Companion Volume)*, 2012.
[34] X. Cao, G. Cong, B. Cui, C. S. Jensen, and C. Zhang. The Use of Categorization Information in Language Models for Question Retrieval. In *Proc. of CIKM*, 2009.
[35] 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.
[36] L. Nie, Y. Zhao, X. Wang, J. Shen, and T. Chua. Learning to recommend descriptive tags for questions in social forums. *TOIS*, 2014.
[37] B. M. Evans and E. H. Chi. Towards a Model of Understanding Social Search. In *Proc. of CSCW*, 2008.
[38] D. Horowitz and S. D. Kamvar. The Anatomy of a Large-Scale Social Search Engine. In *Proc. of WWW*, 2010.
[39] Z. Li, H. Shen, G. Liu, and J. Li. SOS: A Distributed Context-Aware Question Answering System Based on Social Networks. In *Proc. of ICDCS*, 2012.
[40] M. Mcpherson. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 2001.
[41] G. A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 1995.
[42] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood. Fast hash table lookup using extended bloom filter: an aid to network processing. In *Proc. of SIGCOMM*, 2005.
[43] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proc. of USENIX Security*, 2004.
[44] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor. Learning from the past: Answering new questions with past answers. In *Proc. of WWW*, 2012.
[45] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Proc. of WWW*, 1997.
[46] Planetlab. http://www.planet-lab.org/.
[47] N. A. Christakis and J. H. Fowler. *Connected: The surprising power of our social networks and how they shape our lives*. Hachette Digital, 2009.
[48] H. M Wallach. Topic modeling: beyond bag-of-words. In *Proc. of ICML*, 2006.
[49] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, 2014.
[50] H. Shen, G. Liu, and N. Vithlani. Socialq&a: An online social network based question and answer system. In *Proc. of ICCCN*, 2015.



Haiying Shen received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively. She is currently an Associate Professor in the CS Department at University of Virginia. Her research interests include distributed computer systems and computer networks with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and grid and cloud computing. She was the Program Co-Chair for a number of international conferences and member of the Program Committees of many leading conferences. She is a Microsoft Faculty Fellow of 2010, a senior member of the IEEE and a member of the ACM.



Guoxin Liu received the BS degree in BeiHang University 2006, and the MS degree in Institute of Software, Chinese Academy of Sciences 2009. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering of Clemson University. His research interests include distributed networks, with an emphasis on Peer-to-Peer, data center and online social networks. He is a student member of IEEE.



Haoyu Wang received the BS degree in University of Science & Technology of China, and the MS degree in Columbia University in the city of New York. He is currently a Ph.D student in the Department of Computer Science of University of Virginia. His research interests include data center, cloud and distributed networks.

Nikhil Vithlani received his BS degree in Computer Engineering from Mumbai University in 2010 and a MS degree in Computer Engineering from Clemson University in 2012. He is currently employed by Amazon.com. His research interests include digital video delivery and DRM technologies, Social Network, Machine Learning, and Data Mining.