IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 31, 2018, accepted September 21, 2018, date of publication September 28, 2018, date of current version October 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872568

SOQAS: Distributively Finding High-Quality Answerers in Dynamic Social Networks

IMAD ALI[1,2], RONALD Y. CHANG[3], (Member, IEEE), AND CHENG-HSIN HSU[4], (Senior Member, IEEE)

[1]Taiwan International Graduate Program in Social Networks and Human-Centered Computing, Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan
[2]Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu 300, Taiwan
[3]Research Center for Information Technology Innovation, Academia Sinica, Taipei 11529, Taiwan
[4]Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan

Corresponding author: Ronald Y. Chang (rchang@citi.sinica.edu.tw)

ABSTRACT Compared with community-based question answering systems and modern search engines, social network-based question answering systems are more efficient in addressing non-factual questions. In such systems, askers search answerers among their 1-hop neighbors; however, high-quality answerers may exist in the k-hop neighbors of the social networks who are not known to askers directly. To address this problem, we propose a dynamic SOcial network-based Question Answering System (SOQAS) that finds high-quality answerers to each asker's question with high response rate and low-response time. The SOQAS finds high-quality answerers in the k-hop dynamic social network and selects optimal relays at each hop to forward the question to, via social referral chains. In particular, the profile information is exchanged among k-hop neighbors, and leveraged for finding high-quality answerers and optimal relays at each hop, so as to increase the response rate and reduce the response time. We conduct trace-driven simulations, which show that, compared with the state-of-the-art schemes, SOQAS achieves: 1) higher average expertise levels by more than 42%, 2) higher average response rate by more than 26%, and 3) lower response time with as high as 27% reduction. Furthermore, under diverse system parameters, such as question arrival rate, keywords per question, answerers per question, number of hops, and predictability, the SOQAS consistently outperforms the state-of-the-art schemes.

INDEX TERMS Dynamic social networks, distributed question answering system, high-quality answerers, protocols, social referral chains.

I. INTRODUCTION

Modern search engines such as Google, Bing, and Yahoo! may not provide satisfactory answers to *non-factual* questions [1], due to the lack of relevant contents to retrieve from their databases [2]. Non-factual questions require people's opinions, suggestions, recommendations, etc., and therefore are better answered by humans through question answering systems. Community-based question answering systems such as Yahoo!Answers [3], Quora [4], Answer.com [5], and Stack Overflow [6] suffer from low-quality answers and long response time [7]. Alternatively, social network-based question answering systems such as Aardvark [8] and SOS [9] are promising in coping with the issue, because friends in social networks know each other's expertise (levels) [7] and trust each other [10]. Finding high-quality answerers in dynamic social networks is however challenging, because social network users have diverse expertise levels and are not online (active) all the time.

In this article, we study the problem of distributively finding high-quality answerers in a k-hop dynamic social network to each question. The core challenge of our work is *whom to choose among the neighbors to reach the high-quality answerers via social referral chains in a k-hop dynamic social network*. We argue that high-quality answerers are those who have high expertise levels in their respective fields. To the best of our knowledge, the problem considered in this article is new. The closest studies in the literature are probably Shen *et al.* [9], Lin and Shen [11], Zhang *et al.* [12], and Ali *et al.* [13]. The former two studies [9], [11] retrieve pre-assigned answers to given questions, while the third study [12] searches for randomly selected experts. These three studies, however, assume *static* social networks where

users do not dynamically go online/offline. Our earlier work [13] finds answerers of particular expertise levels in a *centralized* infrastructure, which is more suitable in smaller-scaled social networks.

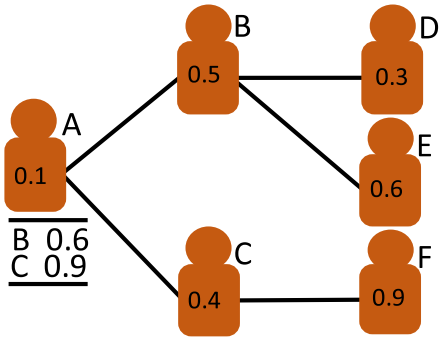

FIGURE 1. An illustrative social network.

Fig. 1 shows a simplified example of a social network. The number represents the expertise level (on the scale of 0 to 1) of each user with respect to user A's question, where a higher value means better expertise level and potentially higher-quality answerers. In natural settings, A only knows his/her 1-hop neighbors B and C's expertise levels. This may lead to suboptimal choices of answerers. Therefore, we propose to maintain a table of *highest expertise level* for each user from next-hop neighbors' social networks. For example, as shown in Fig. 1, the table maintained at user A shows that the highest expertise levels that users B and C can promise A to reach from their networks are 0.6 (user E) and 0.9 (user F), respectively. Accordingly, user A forwards the question to neighbor C, and then C helps A to find answerer with expertise level of 0.9 by forwarding the question to neighbor F.

To distributively find high-quality answerers and the optimal relays at each hop to forward the questions to via social referral chains, we propose a dynamic SOcial network-based Question Answering System (SOQAS). SOQAS consists of two protocols responsible for: (i) exchanging information of users' expertise among k-hop neighbors, and (ii) identifying users of highest expertise levels and forwarding each question along the social referral chains. Furthermore, we also propose to take users' online times into consideration in dynamic social networks. Lastly, we exclusively consider *distributed* solutions for several reasons: (i) a scalable solution is crucial as the number of users in the social networks is rapidly increasing, (ii) a robust solution is vital to handle high question rates and avoid single point of failure, and (iii) an inexpensive solution in terms of cost and bandwidth is desired. The main contributions of the article are summarized as follows:

- We propose SOQAS which is run by individual users to identify high-quality answerers by exchanging information of k-hop neighbors' expertise in dynamic social networks.
- SOQAS leverages the exchanged information to identify optimal relays at each hop to build social referral chains that lead to high-quality answerers. As a result, SOQAS increases the response rate and reduces the response time of each question.
- We conduct large-scaled simulations to evaluate SOQAS's performance in comparison with state-of-the-art schemes. Our simulation results show that compared to the state-of-the-art schemes, SOQAS achieves: (i) higher average expertise levels by more than 42%, (ii) higher average response rate by more than 26%, and (iii) lower response time with as high as 27% reduction. Additionally, over networks of different sizes, SOQAS consistently performs better under diverse parameters.

The rest of the article is organized as follows. Sec. II reviews the related work. Sec. III presents the system model and problem statement. Sec. IV describes the proposed solution method. Sec. V presents and discusses our trace-driven simulation results. Sec. VI concludes the article.

II. RELATED WORK

We survey two categories of the question answering systems in this section.

A. CENTRALIZED QUESTION ANSWERING SYSTEMS

In centralized question answering systems, each question is sent by an asker to a central server that identifies the answerers and forwards the question to them. *Community-based question answering systems* [14]–[16] are classic examples of centralized question answering systems. Zhao *et al.* [14] identify potential answerers from their profiles and answers' history, and the question is then forwarded to the best answerer. Nie *et al.* [15] present a scheme consisting of offline learning component and online search component that is used to rank the potential answerers via pairwise comparisons. To find potential answerers, semantic relevance between pairs of question-answer and users' authority on the question is presented [16]. Srba and Bielikova [17] conduct a comprehensive survey on community-based question answering systems such as Yahoo!Answers, Quora, and Stack Overflow. These community-based systems may not always provide high-quality and trusted answers to the askers because users are anonymous to each other and their expertise levels are unknown.

On the other hand, Paul *et al.* [18] analyze Twitter for different types and topics of questions and find that social networks can be leveraged for asking questions. Some representative *social network-based question answering systems* in this category are Aardvark [8], IM-an-Expert [19], CRAQ [20], and SearchBuddies [21] for finding answerers to asked questions. Aardvark [8] is a social search engine, where questions are forwarded to identified answerers in asker's extended social network. IM-an-Expert [19] is a synchronous question answering system that identifies available potential answerers and then forwards the questions to them, for real-time dialogue via the instant messenger. CRAQ [20] finds a group of potential answerers from their

tweets to answer the question in a collaborative mechanism. SearchBuddies [21] is a system that responds to questions asked by Facebook users on their Facebook-walls as their status message where SearchBuddies provides the best link to users' messages from the search results. Lappas *et al.* [22] identify experts with required skills for a given project such that all experts can work as a team. Bouadjenek *et al.* [1] conduct a detailed survey on different social network-based systems. Ali *et al.* [13] propose a social network-based question answering system for identifying and forwarding questions to potential answerers in dynamic social networks. All these systems achieve their desired objectives; however, centralized systems may suffer from high service request rate, single point of failure, and privacy concerns.

B. DISTRIBUTED QUESTION ANSWERING SYSTEMS

In distributed question answering systems, a user searches for potential answerers among neighbors who can most likely answer the question. Flooding a question to all neighbors (and neighbors of neighbors, etc.) results in large overheads and an overwhelming number of received answers, and therefore is inefficient [23]. Targeting specific neighbors for different applications is more efficient. Kukla *et al.* [24] find that users are more likely to answer the questions if the requests come via a chain of acquaintance in a social network. Some of the representative studies in this category are reported in [25]–[29]. In [25], a social network-based question answering system with a spammer detection mechanism is proposed which focuses on the trustworthiness of required answerers along with their willingness and capability. Similarly, another work [26] proposes a framework that leverages multi-hop friendship relations to identify and select trustworthy participants among neighbors or neighbors of neighbors to participate in a sensing campaign. Guo *et al.* [27] present a privacy-preserving based friend recommendation scheme for social network users who are interested in finding similar users, and want to establish social links with unknown similar users by leveraging multi-hop chains. Likewise, Shen *et al.* [28] propose a social network-based question answering system for improving the security of social users by protecting their privacy while forwarding questions in the social referral chains. In Lin *et al.* [29], a system is introduced that helps the users to manage their social networks, and reach out to their extended network (the neighbors of their neighbors) to find their expertise and information.

Generally, in a social network-based question answering system, if no answerer is found among neighbors for a given question, the question is then forwarded either to the most relevant neighbor, or neighbor with the highest degree until the desired experts are found [12]. Similarly, two state-of-the-art question answering systems, i.e., SOS [9] and iASK [11], select relevant neighbors for forwarding questions based on combination of different metrics such as neighbors' willingness to answer or forward, profiles' similarity to question, and response rate to search for answerers, if no answerer is found among the neighbors. These strategies, however, cannot guarantee that the answerers found are high-quality answerers in a k-hop social network. There are also studies on social network-based peer-to-peer systems [30]–[32] which aim to search for a particular content in the network using social network properties; however, they are quite different from social network-based question answering systems.

III. SYSTEM MODEL

In this section, we present the system model and the problem statement. Table 1 summarizes the notations used throughout this article.

TABLE 1. Summary of notations.

Notation	Description
T, $\mathcal{T}$	Time and one-hour time slots' set
$d^T(u_i, u_j)$	Response time of user u_i to user u_j
$\mathcal{O}_i$	Set of online times of user u_i
$\mathcal{K}$	Set of answerers
G^T	Time-dependent graph
σ_q	Question q buffered time
p_i	Profile keywords vector of user u_i
$\mathcal{N}_i$, $\mathcal{N}_i^*$	Neighbors and online neighbors of user u_i
Γ_i	Neighbors Information Table of user u_i
Υ_i	Keywords Score Table of user u_i
$\Psi_{k,q}$	Expertise level of user u_k on question q
Ω_i	Hop distance of user u_i

A. DISTRIBUTED DYNAMIC SOCIAL NETWORK MODEL

The underlying social network in SOQAS is dynamic in nature where users have varying online times. It has two main components: (i) *bootstrap server* and (ii) *clients*, as shown in Fig. 2. In SOQAS, clients are the users who construct the dynamic social network based on bidirectional neighbor links (e.g., Facebook). Similar to online social networks, each SOQAS user is associated with a unique social ID and a profile containing personal information, personal attributes, timeline posts, and question answering record. We assume

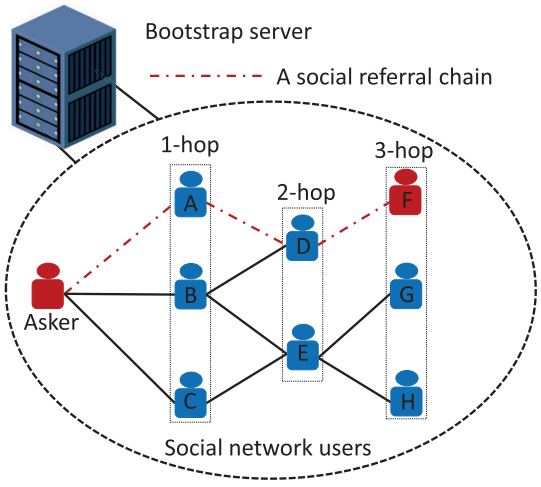

FIGURE 2. The considered social network-based question answering system.

that social network users *truthfully* post information on their timelines and provide answers to questions of their interests and expertise. However, the spammer detection techniques reported in [25], [33] can be utilized to detect and prevent the malicious users from taking part in the question answering system if the assumption of truthfulness is violated. The profiles comprising of timeline posts and previously answered questions are used to identify experts, similar to [12], [34]. Each user may have k-hop neighbors but can directly communicate with 1-hop neighbors only. An example is shown in Fig. 2, where the asker has 3-hop neighbors. The bootstrap server maintains a list of users who have recently joined/left the social network. With the help of the bootstrap server, a user u_i knows the personal attributes of his/her neighbors $\mathcal{N}_i$ in the dynamic social network. Multiple bootstrap servers can be utilized, if necessary, to avoid the single point of failure problem.

In dynamic and distributed systems, modeling of network users' online time plays an important role in forwarding questions. Let $[O^s_{i,n}, O^e_{i,n}) = \left\{t \in \mathcal{T} \mid O^s_{i,n} \leq t < O^e_{i,n}\right\}$ represent the nth online time of user u_i, where, without loss of generality, we consider 1-hour time slots, i.e., $\mathcal{T} = \{0, 1, 2, \ldots, 23\}$, and $O^s_{i,n} \in \mathcal{T}$ and $O^e_{i,n} \in \mathcal{T}$ are the starting and ending times of online time intervals. All the disjoint online times of user u_i is represented by $\mathcal{O}_i = \bigcup_n [O^s_{i,n}, O^e_{i,n})$. For example, $\mathcal{O}_i = [14, 17)$ shows user u_i's online time is from 2 p.m. to 5 p.m., and $\mathcal{O}_j = [14, 17) \cup [20, 23)$ represents user u_j's online times are from 2 p.m. to 5 p.m., and from 8 p.m. to 11 p.m.

We model the dynamic social network by a time-dependent undirected social graph, $G^T = (\mathcal{V}^T, \mathcal{E}^T)$, where the vertex set $\mathcal{V}^T$ represents all users and the edge set $\mathcal{E}^T$ contains all neighbor links at time $T \in \mathcal{T}$. Let $d^T(u_i, u_j)$ represent the response time user u_i needs to process (i.e., answer or forward) user u_j's question after receiving the question. In the graph G^T, we consider user u_i *online* at time t if $t \in \mathcal{O}_i$. Every neighbor $u_j \in \mathcal{N}_i$ of user u_i is either online or offline with respect to user u_i's online time. We say that user u_j is online in u_i's nth online time if $[O^s_{i,n}, O^e_{i,n}) \cap \mathcal{O}_j \neq \emptyset$, and otherwise offline. A message is forwarded from user u_i to user u_j if $\mathcal{O}_i \cap \mathcal{O}_j \neq \emptyset$. If $[O^s_{i,n}, O^e_{i,n}) \cap \mathcal{O}_j \neq \emptyset$, user u_i answers/forwards the question to user u_j during their common online time after $d^T(u_i, u_j)$; otherwise, the question is buffered at user u_i until both users go online in their forthcoming overlapping online times. Let $[t^r_{i,q}, t^f_{j,q}]$ be the *question buffered time* at u_i for u_j for question q, where $t^r_{i,q} \in \mathcal{O}_i$ and $t^f_{j,q} \in \mathcal{O}_i \cap \mathcal{O}_j$ represent the times u_i receives and forwards question q to u_j, respectively. The question buffered time is given by $\sigma_q(u_i, u_j) = t^f_{j,q} - t^r_{i,q}$.

Consider an asker $u_a \in \mathcal{V}^T$ has a question q, and wants to find a high-quality answerer $u_k \in \mathcal{V}^T$ for the question. Since a positive correlation between human-ranked answerers and their expertise levels exists [35], we use the expertise levels as a measure to find high-quality answerers. The expertise level is a function of the asked question and answerer's profile keywords. We denote the expertise level of user u_k on question q by $\Psi_{k,q}$. Let p_k denote the vector of keywords, extracted from user u_k profile posts and previously answered questions. We say that user u_k has high expertise level if user u_k's profile keywords vector p_k matches question q well. Naturally, in a social network, an asker has no knowledge of the entire network users; however, high-quality answerers may exist in the asker's k-hop social network. Thus, an asker u_a requests his/her neighbors to help search and forward the question to high-quality answerers. The neighbors then search for the answerers and forward the question to their neighbors for help, if no suitable answerers are found. Forwarding question from neighbors to neighbors defines a social referral chain, as shown by the dash-dotted line in Fig. 2 from an asker to an answerer. However, there is no guarantee that the found answerers are high-quality answerers, as this is only possible if the question is flooded to asker's all k-hop neighbors, which increases tremendous overhead and is not feasible.

B. PROBLEM STATEMENT

The objective of our work is to find high-quality answerers to each question in a k-hop dynamic social network. Considering that high-quality answerers are positively correlated with the answerers' expertise levels, we explicitly aim to search and forward the question q to answerers who have the highest expertise levels $\Psi_{k,q}$ to that question in the k-hop dynamic social network via social referral chains. To this end, SOQAS utilizes two protocols, as presented in the next section.

IV. PROPOSED SOLUTION

The objective of SOQAS is to: (i) identify high-quality answerers to each question, and (ii) select optimal relays for building social referral chains to forward the question to the answerers. We propose two protocols, `BuildNIT` and `SearchNIT` in SOQAS. We describe the details in the following.

A. BUILDNIT

Social closeness is a strong attribute for the willingness of users to forward or answer the questions. As neighbors having social links are considered socially close, they are more willing to share some information and help each other by forwarding or answering questions [9], [11], [28]. In SOQAS, each user u_i shares his/her social identity (ID), hop-distance (Ω), and profile keywords vector (p) with (socially close) neighbors $\mathcal{N}_i$. User u_i's profile keywords vector p_i consists of l most frequently used keywords $p_i = \{w_{i,n}\}_{n=1,2\ldots,l}$ where w represents a keyword, and l varies on each profile bases. State-of-the-art privacy-preserving techniques (e.g., [36]–[38]) can be utilized to protect users' privacy while sharing their profiles information with neighbors.

Every two neighbors u_i and u_j can only exchange the information with each other if $\mathcal{O}_i \cap \mathcal{O}_j \neq \emptyset$. Each user u_i maintains all k-hop neighbors in a Neighbors Information

Table (NIT) denoted by Γ_i. User u_i's NIT Γ_i contains neighbor u_j's information $\forall u_j \in \mathcal{N}_i$, and neighbor u_m's information $\forall u_m \in \Gamma_j,\ \forall u_j \in \mathcal{N}_i$. The information of a neighbor $u_m \in \Gamma_j,\ u_j \in \mathcal{N}_i$ in user u_i's NIT Γ_i is represented as $\Gamma_i(j) = (\mathrm{ID}_m, \Omega_m, p_m)$. Only the hop-distance Ω_m of a particular user u_m is updated in user u_j's NIT Γ_j if user u_m's information is received again with smaller hop-distance, which is then forwarded to user u_i for update. The information of particular user u_m is disseminated until $\Omega_m = k$ in the dynamic social network via gossip protocol [39], where k is a system parameter. While disseminating neighbors information, the hop-distance is incremented by one each time.

1-hop IDs	k-hop IDs	Hop-dist.	Profile Keywords
B	-	1	Networks (2), Java (3), Database (4)
B	D	2	Algorithms (2), Java (2), Music (3)
B	E	2	Java (4), Database (3), Music (3)
C	-	1	Networks (3), Algorithms (3), Database (2)
C	F	2	Networks (5), Algorithms (5), Java (4)

FIGURE 3. Sample Neighbors Information Table (NIT) for user *A* in Fig. 1.

For illustrations, let the sample keywords of Fig. 1 users' profiles be Networks, Algorithms, Java, Database, and Music, as shown in Fig. 3 for user A's NIT Γ_A. The number in parenthesis represents corresponding term frequency (tf), which is the number of times a particular keyword occurs in a user profile. If a keyword does not exist in a user's profile (zero tf), it is not shown. The second column represents the k-hop neighbors IDs received via corresponding 1-hop neighbors in the first column, while the third column shows the hop-distance of each neighbor from user A. The profile keywords in the forth column correspond to users in the second column if exist, otherwise, to users in the first column.

The `BuildNIT` protocol mainly uses *exchange* and *update* operations for building and maintaining users' NITs as summarized in **Algorithm** 1. The exchange operation is used when information exchange takes place between two neighbors for the first time, while update operation is used when already exchanged information needs some modification. Initially, each user u_i maintains the following sets: (i) $\bar{S}_i$ contains a list of 1-hop neighbors $\mathcal{N}_i$, (ii) S_i contains those 1-hop neighbors who already agreed, and exchanged their information at least one time with user u_i, and (iii) R_i contains u_i's 1-hop black-listed neighbors who are not interested in information exchange, as shown in line 1.

Whenever user u_i goes online, u_i finds online neighbors $\mathcal{N}_i^* \subseteq \{\bar{S}_i \cup S_i\}$ with the help of the bootstrap server in lines 2 and 3. User u_i initiates the exchange operation with neighbor u_j if u_j is online, i.e., $u_j \in \mathcal{N}_i^*$, and is not yet contacted for information exchange, i.e., $u_j \in \bar{S}_i$. User u_i sends his/her information to u_j, and waits for user u_j's response in line 5. User u_j is shifted from set $\bar{S}_i$ to set S_i, if u_j accepts the exchange operation. User u_i inserts a new entry in the NIT accordingly as shown in lines 6–9. Otherwise, u_j is shifted

Algorithm 1 `BuildNIT` Protocol (Constructing Neighbors Information Table)

1: // Every user u_i creates his/her NIT Γ_i, $S_i = \oslash$; $R_i = \oslash$; and $\bar{S}_i = \mathcal{N}_i$
2: **while** $\mathcal{O}_i = 1$ **do**
3: Finds $\mathcal{N}_i^* \subseteq \{\bar{S}_i \cup S_i\}$
4: **if** $u_j \in \mathcal{N}_i^*$ & $u_j \in \bar{S}_i$ **then**
5: u_i initiates exchange operation & waits
6: **if** u_j accepts the exchange **then**
7: $S_i = S_i \cup \{u_j\}$
8: $\bar{S}_i = \bar{S}_i - \{u_j\}$
9: u_j's information is inserted in Γ_i
10: **else** $R_i = R_i \cup \{u_j\}$
11: $\bar{S}_i = \bar{S}_i - \{u_j\}$
12: **if** $u_j \in \mathcal{N}_i^*$ & $u_j \in S_i$ **then**
13: u_i synchronizes with u_j
14: when u_j initiates exchange/update operation
15: u_j's entries in Γ_i are updated

from set $\bar{S}_i$ to set R_i, if u_j does not show interest in information exchange after u_i's c numbers of requests time out as shown in lines 10 and 11. However, if $u_j \in \mathcal{N}_i^*$, and $u_j \in S_i$ then u_i first synchronizes with u_j to know what has been already exchanged. When user u_j has either new neighbors' information to exchange or wants to modify the already exchanged information, user u_j initiates the exchange/update operation with user u_i. User u_i then updates the corresponding entries in his/her NIT Γ_i as shown in lines 12–15. The exchange and update operations are performed regularly by each user u_i to build and maintain the NIT Γ_i.

Let r be the number of bytes required to store a neighbor's information. Then the space complexity at each user u_i is $\mathcal{O}(r|\mathcal{M}_i|)$ bytes, where $|\mathcal{M}_i|$ is the total number of k-hop neighbors of user u_i. The space complexity increases linearly as the number of k-hop neighbors $|\mathcal{M}_i|$ increases; however, each SOQAS user manages it by exchanging information with socially close neighbors only. This way, each SOQAS user significantly reduces the complexity by keeping a limited number of neighbors and their profile information. In our simulation setup reported in Sec. V-B, we considered neighbors socially close if their profile similarity is greater than a particular value. **Algorithm** 1 also has linear time complexity of $\mathcal{O}(|\mathcal{N}_i|)$.

B. SEARCHNIT

The `SearchNIT` protocol serves two main functions of finding: (i) high-quality answerers and (ii) optimal relays to build social referral chains to forward the question to. We discuss both in detail as follows.

1) FINDING HIGH-QUALITY ANSWERERS

When a question q is initiated at the asker u_a, it is represented as a vector of keywords. To find an answerer u_k for question q, the asker u_a matches the question keywords vector q with

each neighbor u_j profile keywords vector p_j in NIT Γ_a. To find a quantitative measure of matching between question q and a user u_k's profile vector p_k, there are several techniques, e.g., vector space model, the Euclidean distance, and correlation coefficient [40]. Similar to [9], [28], [31], we use the vector space model to find each user's expertise level based on the user profile keywords. Each user u_i creates a Keywords Score Table (KST) Υ_i which contains the term frequency and inverse document frequency (tf-idf) score for each keyword of NIT Γ_i [41]. The expertise level $\Psi_{k,q} \in [0, 1]$ of a user u_k determines how likely the question q vector matches with the profile vector p_k of the user u_k. It is calculated by:

$$\Psi_{k,q} = \frac{\sum_{l=1}^{|V|} q^l p_k^l}{\sqrt{\sum_{l=1}^{|V|}(q^l)^2}\sqrt{\sum_{l=1}^{|V|}(p_k^l)^2}}, \tag{1}$$

where q^l and p_k^l represent the tf-idf scores of the keyword l in question q, and in user u_k's profile vector p_k, respectively. $|V|$ denotes the total number of keywords in q in $|V|$ dimensional vector space.

IDs	B	C	D	E	F
Networks	3.3	0.0	0.0	5.0	8.4
Algorithms	0.0	3.3	0.0	5.0	8.4
Java	5.0	3.3	6.7	0.0	6.7
Database	6.7	8.4	0.0	3.3	0.0
Music	0.0	7.5	5.0	0.0	0.0

FIGURE 4. Sample Keywords Score Table (KST) for user *A* in Fig. 1.

User A's KST Υ_A for the NIT Γ_A is shown in Fig. 4, where each cell represents the tf-idf score (without log) for a particular keyword. The high-quality answerer to A's question regarding Networks and Algorithms is user F who has the highest score among all users.

Algorithm 2 `SearchNIT` Protocol (Finding High-Quality Answerers)

1: // Every user u_i create his/her KST Υ_i, and $\mathcal{W}_i \leftarrow \oslash$
2: **for** each q of asker u_a **do**
3: Find $\Psi_{k,q}$ using (1)
4: **if** $\Psi_{k,q} > \Psi_{a,q}$ *then*
5: $\mathcal{W}_i \leftarrow u_k$
6: Select top-$\mathcal{K}$ Answerers in $\mathcal{W}_i$

The pseudocode of `SearchNIT` protocol for finding top-$\mathcal{K}$ high-quality answerers is summarized in **Algorithm** 2. First, each user u_i creates KST Υ_i from NIT Γ_i and an empty set $\mathcal{W}_i$ that holds potential answerers' expertise levels for question q, as shown in line 1. For each question q, asker u_a finds the expertise level $\Psi_{k,q}$ for each user u_k in Υ_a by (1) in lines 2–3. We consider that a potential answerer u_k's expertise level $\Psi_{k,q}$ should be greater than the asker u_a's expertise level $\Psi_{a,q}$ on question q. Thus, we only consider user u_k as a potential answerer if $\Psi_{k,q} > \Psi_{a,q}$ as shown in lines 4–5.

Algorithm 3 `SearchNIT` Protocol (Finding Optimal Relays)

1: // Every asker u_a selects optimal relays for the top-$\mathcal{K}_a$ answerers
2: **for** each $u_k \in \mathcal{K}_a$ **do**
3: **if** $\Omega_k = 1$ **then**
4: Forward q at $[O^s_{a,n}, O^e_{a,n}) \cap [O^s_{k,n}, O^e_{k,n}) \neq \oslash$
5: **if** $\Omega_k \neq 1$ **then**
6: Find all u_js for $\{u_k \mid u_k \in \Gamma_j,\ u_j \in \mathcal{N}_i\}$
7: **if** $|u_j| = 1$ **then**
8: Forward q as line 4
9: **if** $|u_j|$s $\geqslant 2$ are online with same Ω_k **then**
10: Select u_j with min $d_k^T(u_a, u_j)$
11: **if** $|u_j|$s $\geqslant 2$ are online with diff. Ω_k **then**
12: Select u_j with smaller Ω_k
13: **if** $|u_j|$s $\geqslant 2$ are offline (same/diff. Ω_k) **then**
14: Select u_j who goes online first
15: Repeat until TTL expires or u_k is found

We rank all the potential answerers u_k by their expertise levels in line 6. We implement **Algorithm** 2 in linear time complexity of $\mathcal{O}(|\mathcal{K}|)$ [42] for selecting top-$\mathcal{K}$ high-quality answerers.

2) FINDING OPTIMAL RELAYS

In a distributive question answering system, it is challenging to find optimal relays that lead to multiple answerers, because relays drop the question q if it has already been forwarded to the answerer. This results in finding fewer answerers than required. Since the NIT Γ_i of user u_i does not keep structure of k-hop neighbors, it becomes crucial whom to select as a relay among the 1-hop neighbors who may lead to the identified answerer. The naive strategy is to forward the question to immediate online 1-hop neighbors like epidemic routing [43]; however, it may not lead the social referral chains to identified answerers.

We use rational strategies for choosing the optimal relays in the 1-hop neighbors in each user u_i's NIT Γ_i, as described as follows. After selecting the required number of top-$\mathcal{K}$ answerers with highest expertise levels, each asker u_a then finds the answerers' IDs in the NIT Γ_a. The asker u_a waits for the answerer u_k to go online if the answerer u_k is an 1-hop neighbor of asker u_a; otherwise, asker u_a finds among the 1-hop neighbors responsible for providing the answerer u_k's information. If asker u_a finds only one neighbor u_j in the 1-hop neighbors for answerer u_k, then the asker forwards the question to neighbor u_j at $[O^s_{a,n}, O^e_{a,n}) \cap [O^s_{j,n}, O^e_{j,n}) \neq \oslash$. However, if more than one neighbors in the 1-hop neighbors are found, the asker u_a breaks the tie with the following strategy. If all the online neighbors have the same Ω_k to the answerer u_k, the neighbor with smaller response time is selected. If all the online neighbors have different Ω_k to answerer u_k, the neighbor with smaller hop-distance is

selected. If all neighbors are offline, the neighbor who goes online first is selected, irrespective of Ω_k to the answerer u_k. The asker then forwards the question q and the identified answerer u_k's ID_k to the selected relays for answering or further forwarding. The relay finds the answerer u_k's ID_k in his/her NIT Γ_j, and selects the optimal relays as described before. This process is repeated at each relay and is terminated if either the answerer u_k is found or the question's Time-To-Live (TTL) expires. The TTL decreases by one whenever the question is forwarded in the social referral chain. In Fig. 1, user A finds that the answerer's ID is F and is 2 hops away from user A. The optimal relay to reach F is then user C.

The pseudocode of `SearchNIT` protocol for finding optimal relays is summarized in **Algorithm** 3. If the answerer $u_k \in \mathcal{N}_i$, i.e., $\Omega_k = 1$, then the question is forwarded to the answerer u_k at the online time, as shown in lines 2–4. However, if the answerer $u_k \in \Gamma_j$, $u_j \in \mathcal{N}_i$, i.e., $\Omega_k \neq 1$, then u_a finds the 1-hop neighbor $u_j \in \mathcal{N}_i$ via whom u_k can be accessed. If there is one user u_j who can reach answerer u_k, then the asker u_a forwards the question to user u_j at their common online time as shown in lines 5–8. However, if there are more than one user who can reach a particular answerer u_k, then the asker u_a selects the suitable user u_j as explained before and is shown in lines 9–15. **Algorithm** 3 also has linear time complexity of $\mathcal{O}(|\mathcal{K}|)$.

V. EVALUATIONS

In this section, we evaluate SOQAS's performance using trace-driven simulations.

A. DATASET COLLECTION

We consider Facebook as the representative dynamic social network, and collect three real datasets. The datasets contain each user's neighbors with neighbor links and time-stamped timeline posts. Because of privacy concerns, Facebook has strict rules for accessing users' data. For example, a user can only collect his/her 1-hop neighbors' data but not 2-hop neighbors' data. Therefore, data collection from Facebook via its Application Programming Interface (API) is infeasible to us. To circumvent this limitation, we use Octoparse [44], which is a web crawler software, to collect data. Octoparse models web browsing behavior of humans such as typing texts or clicking mouses, and allows us to extract data from its built-in web browser. After running Octoparse, we write scripts to extract users' information and their timeline's posts.

We performed data collection from early August 2017 to mid-October 2017. We recruited three university students as the seed Facebook users for collecting their social networks data using our developed scripts. These seed users are friends on Facebook and thus have plenty of common neighbors. Each user logs in to his/her Facebook and uses breadth-first search method for collecting all neighbors' publicly available posts from their Facebook-walls (Shen *et al.* [9] also collected Facebook data using the same breadth-first search method, but for 1000 users only). This way, each seed user collects 1-hop neighbors' data. Next, with similar procedure, 2-hop neighbors' data were collected via 1-hop neighbors. We get the complete overview of Facebook w.r.t. the seed users by repeating the same procedure. However, the success rate of getting k-hop neighbors' data significantly decreases. Each seed user collected 20,000+ users data from their social networks which constitute our raw datasets.

The raw datasets need to be processed before being used in simulations. The following processed data are required: (i) individual users' online times, (ii) individual users' profile keywords, and (iii) dynamic social networks based on neighbor links. To collect the disjoint online times for each user, we parse all the collected time-stamped posts. We consider a user online at time only, if the user writes (post) something on his/her Facebook-wall at that time. Since most of the collected users' posts are in Chinese, first we, therefore, translated all the contents of collected data to English using Microsoft Azure Service [45]. We parsed all the collected timeline posts of each user for the keywords. We used the keywords to represent a user profile.

To build a social network, we consider users as nodes and connect them with users having neighbor links in the dataset. We build three networks with small, medium, and large size datasets, to evaluate SOQAS. We pick one seed user' collected data and remove users with: (i) incomplete profile information, and (ii) 1 and 2 degrees, because they do not have sufficient number of neighbors for answering and forwarding questions. This dataset yields a small-sized network of 408 users. Similarly, we pick two and three seed users' collected data and connect neighbors with their neighbor links. We filter the users in the same way as the small-sized network, which results in two more datasets of different sizes. These datasets yield medium- and large-sized networks of 795 and 1252 users, respectively. The statistics of all three datasets are summarized in Table 2. We observe that all three networks' node degrees follow power law distribution as shown in Fig. 5. Thus, all three network users are well connected and each user has access to a greater portion of the network to search answerers.

TABLE 2. Statistics of all three datasets.

	Small	Medium	Large
No. of users	408	795	1252
No. of edges	2171	4917	6687
Average degree	10.64	12.37	10.68
Power law coefficient	−2.14	−2.44	−2.57
Online times (hours)			
Average	6.57	6.56	6.51
Std. deviation	1.78	1.79	1.75
No. of keywords			
Average	139.9	144.3	153.4
Std. deviation	65.1	65.8	69.5

B. SETUP

We implement SOQAS in Network Simulator-3 (NS-3) [46] to build: (i) NITs via information exchange and (ii) KSTs for identifying high-quality answerers and optimal relays at each

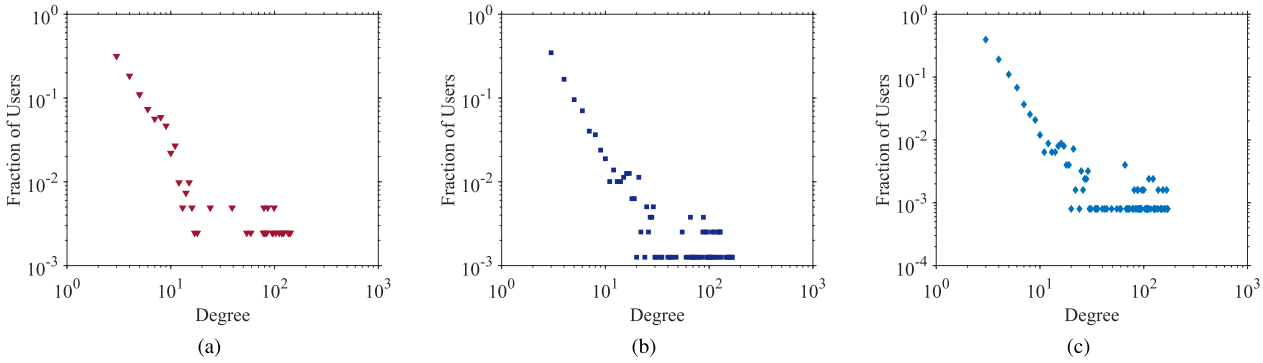

FIGURE 5. Degree distribution of the three considered networks: (a) small, (b) medium, and (c) large.

hop to forward the questions to, via social referral chains in the dynamic social networks. NS-3 is a detailed event-driven network simulator. We also implement three state-of-the-art schemes, i.e., similarity-based scheme [9], [11] (denoted as *Similarity* in figures), degree-based scheme [12] (*Degree*), and random scheme [47] (*Random*). All the three schemes share the same inputs; however, each scheme selects its relays among neighbors to explore users with highest expertise levels in the networks differently. The similarity scheme selects top-k users having the *highest question to profile similarity* among the neighbors. The degree scheme selects top-k users having the *highest degree* among the neighbors. The random scheme selects k random users among the neighbors. Invoking each scheme gives the k social referral chains. Each relay records the user with the highest expertise level and then sends the question 1-hop further by selecting a relay among the neighbors according to the scheme. This process continues until the question's specified TTL expires. When the search finishes, the user with the highest expertise level among all explored users via relays is returned to the asker. Due to the power law characteristic of the underlying social networks, all the schemes in comparison, including SOQAS, explore, on average, the same number of users in the search process. The *Upper Bound* is used to show the expertise levels of users in the *entire network* which may not be achievable because the askers may have no social referral chains to the identified answerers. We do not use Upper Bound for routing the questions, and therefore its response rate and response time are not shown in the figures.

We use a single bootstrap server in our implementation. We assume all users have broadband Internet connections in the simulator where the bandwidth distribution to network users is adopted from a dataset of Washington DC [48]. We assume that 1/10 of the bandwidth is used for social networks. Questions are randomly generated by the simulator from the union of all k-hop users' profile keyword vectors. To generate a question of k-keywords, we create a list of document frequency for each keyword and normalize it. We then divide the normalized list in k equal size groups where k is the number of keywords in a question. To generate a question of k-keywords, the simulator picks one word from each group. This way, each question contains both rare and common keywords.

Every user joins/leaves the network according to his/her online times. Due to privacy concerns, we did not collect the users' conversations in the datasets to measure their actual response times. We adopted the response time distribution from Shen *et al.* [9]. Their analysis showed that more than 60% of the questions were answered within 15 minutes, so they adopted a constant response time of 12 minutes for all users. Since users' response times vary in a social network, we slightly deviate the response times around 12 minutes. Specifically, we assigned integer response times uniformly distributed between 8 and 16 minutes (inclusive) with a mean of 12 minutes for each user in our simulation. Our simulator supports various system parameters including: (i) question arrival rate following a Poisson process, (ii) the number of keywords per question, (iii) the number of answerers per question, (iv) the number of hops allowed for searching high-quality answerers, and (v) *predictability*. The predictability is a real-valued number between 1 and 0 which is used to introduce some randomness in users' online/offline times and makes our simulations more realistic. The simulator matches a random number uniformly chosen between 1 and 0 against the selected value of predictability and let the user to go online only if the random number is smaller than the selected value of predictability. The system parameters with default values are given in Table 3.

TABLE 3. System parameters.

Parameter Name	Values	Default
Questions arrival rate (per hour)	$\{\frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{2}, 1\}$	$\frac{1}{4}$
No. of answerers per question	$\{1, 2, 4, 6, 8\}$	4
No. of keywords per question	$\{4, 6, 8, 10, 12\}$	8
No. of hops	$\{1, 2, 3, 4, 5\}$	3
Predictability	$\{0.6, 0.8, 1\}$	1

To make all the schemes and SOQAS operable, users exchange their 1-hop and k-hop profiles keywords,

respectively, to know each other expertise. We assume that neighbors with similar profiles are socially close and are willing to answer or forward each other's questions. We use cosine similarity [42] for measuring their social closeness which is a real number between 0 and 1. In our simulation settings, each neighbor is willing to respond (answer/forward a question, exchange information) if the neighbor profile cosine similarity is greater than or equal to 0.2. We report results after each user exchanges profile information with sufficient number of neighbors. We measure and report the following performance metrics:

- *Expertise level*, which shows the quality of an answerer to a given question.
- *Response rate*, which shows the percentage of questions that are responded.
- *Response time*, which is the time difference between an asker asking a question and receiving response from an answerer.

Each simulation is repeated 5 times for our solution and all other three schemes. Each simulation run lasts for one week. We report the average results with 95% confidence intervals whenever applicable.

C. RESULTS

Unless otherwise specified, the main take-away messages and simulation results for the medium-sized network are given in this section.

1) SOQAS FINDS HIGHER-QUALITY ANSWERERS

We compare the expertise levels of each question achieved by SOQAS and other schemes in Fig. 6. Fig. 6a reports the cumulative distribution function (CDF) curves of the highest-expertise levels from a sample run. Since all schemes report answerers with highest-expertise levels from their social referral chains, therefore, few questions have expertise levels in the middle. The questions receiving low-expertise levels are due to poor similarity match between questions and users profiles. However, the figure shows that SOQAS results in higher-expertise levels. For example, for 36% of the questions, SOQAS achieves expertise levels of 0.2 or higher, while Similarity, Degree, and Random achieve the same expertise levels for only 19.6%, 20%, and 19.8% of questions, respectively. SOQAS is only 14% away from the upper bound. Fig. 6b shows the average expertise levels across all questions which confirms the trend: SOQAS > Degree > Similarity $\cong$ Random. SOQAS achieves the average expertise levels of 0.38, which is 44.7%, 42.1%, and 44.7% higher than Similarity (0.21), Degree (0.22), and Random (0.21), respectively. SOQAS achieves 75.5% of the upper bound average expertise levels (0.503), while Similarity, Degree and Random achieve 41.7%, 43.7%, and 41.7% of the upper bound, respectively. This is because SOQAS leverages the askers' KSTs for identifying high expertise level answerers while the state-of-the-art schemes search the answerers according to their strategies in the k-hop dynamic social network. Since the simulator randomly generates questions for every asker, on average, Random achieves almost the same expertise levels to Similarity. By the virtue of more neighbors, Degree finds answerers with slightly higher expertise levels than Similarity and Random.

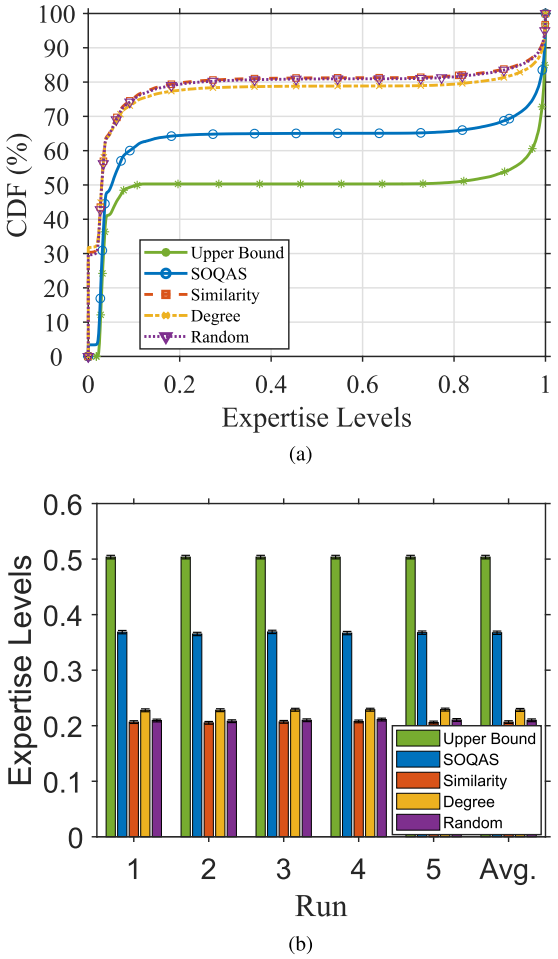

FIGURE 6. SOQAS finds higher-quality answerers: (a) a sample run and (b) overall results.

2) SOQAS PROVIDES MORE ANSWERERS

We report the response rates of individual questions in Fig. 7. Fig. 7a reports the CDF curves of the response rate from a sample run. This figure shows that the number of answerers each question finds are from 0–4 (0%–100%), therefore, all schemes curves are five-stairs. Furthermore, it shows that SOQAS results in much higher response rate. For example, SOQAS delivers 100% response rate for 93.4% of all questions, while Similarity, Degree and Random achieve the same response rates for 67.5%, 64.2%, and 68.8% of the questions, respectively. Fig. 7b presents the average response rate across all questions, which follows the trend: SOQAS > Similarity $\cong$ Random > Degree. This figure demonstrates that SOQAS achieves 26.9%, 28.7%, and 27.0% higher response rate on average as compared to other schemes, respectively. This is because SOQAS leverages **Algorithms** 2 and 3 to forward questions to relays leading to unique answerers.

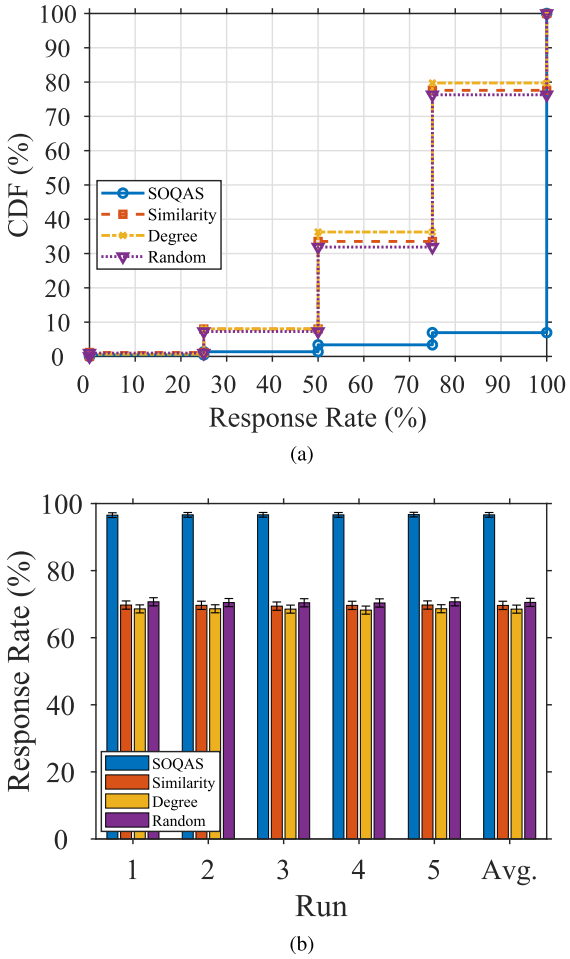

FIGURE 7. SOQAS provides more answerers: (a) a sample run and (b) overall results.

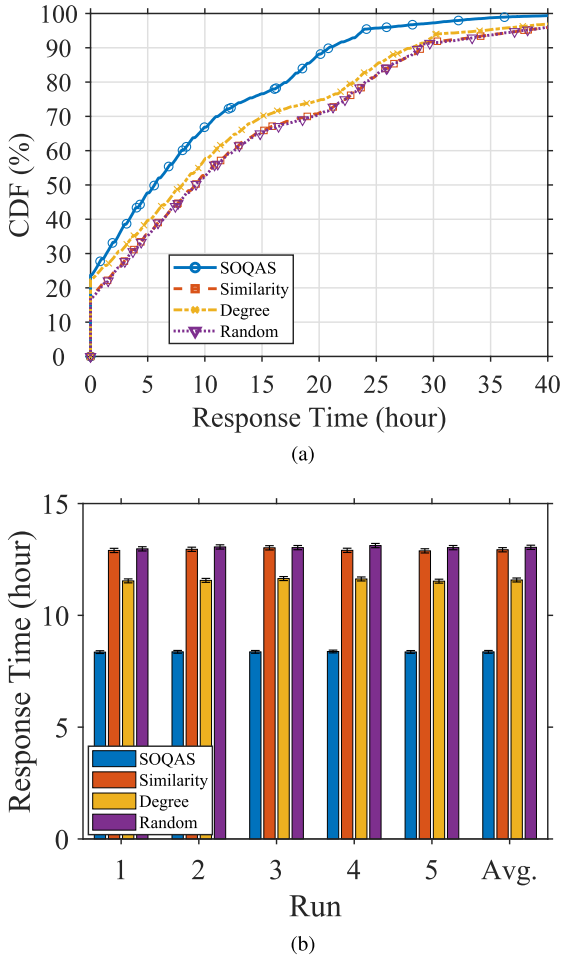

FIGURE 8. SOQAS explores answerers quickly: (a) a sample run and (b) overall results.

The other schemes often end-up on answerers who already responded the question, resulting in fewer unique answerers (lower response rate). Relays in Degree forward questions to users with more neighbors and end-up on same answerers more often as compared to Similarity and Random, resulting into slightly lower response rate.

3) SOQAS EXPLORES ANSWERERS QUICKLY

We plot the response time of each question from all answerers in Fig. 8. Fig. 8a gives the CDF curves of the response time from a sample run. This figure shows that SOQAS yields lower response time; half of the questions are answered within 5.5 hours by SOQAS, while half questions are answered within 9.1, 8.2, and 9.2 hours by Similarity, Degree, and Random, respectively. The long response time of some questions are due to relays who remain frequently offline in the social referral chains. Fig. 8b presents the average response time across all questions which follows the trend: SOQAS < Degree < Similarity $\cong$ Random. This figure shows that on average, SOQAS achieves 34.9%, 27.6%, and 35.4% reduction in response time as compared to other schemes, respectively. This is because, SOQAS utilizes **Algorithms** 3 to select optimal relays at each hop to reach the answerers. The other schemes select relays according to their strategies and then wait for them to go online. The Degree achieves a lower response time as compared to Similarity and Random because users with more neighbors are typically those who are online more frequently.

4) SOQAS GENERATES MODERATE OVERHEAD

We plot the seven days traffic generated by SOQAS in Fig. 9. Fig. 9a shows the CDFs of messages flow over all links. It can be observed that 50% of the links carry less than 10 messages per day that include questions and control messages. Fig. 9b shows a medium busy link having 86 messages in seven days. This include 46 questions (6.6 questions per day), which is reasonable. The peak and mean traffic over the link is 46.7 and 0.02 kbps, respectively. Fig. 9c shows the CDFs of peak traffic of all links over five runs. The figure reveals that 80% of the links have peak traffic of less than 30 kbps while very few links have high peak traffic.

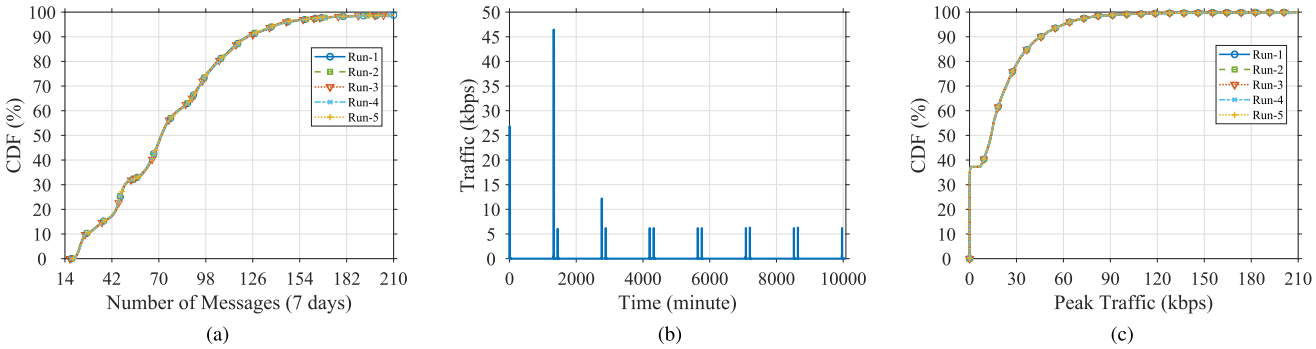

FIGURE 9. SOQAS generates moderate overhead: (a) CDFs of messages over links, (b) a sample traffic over medium busy link, and (c) CDFs of peak traffic.

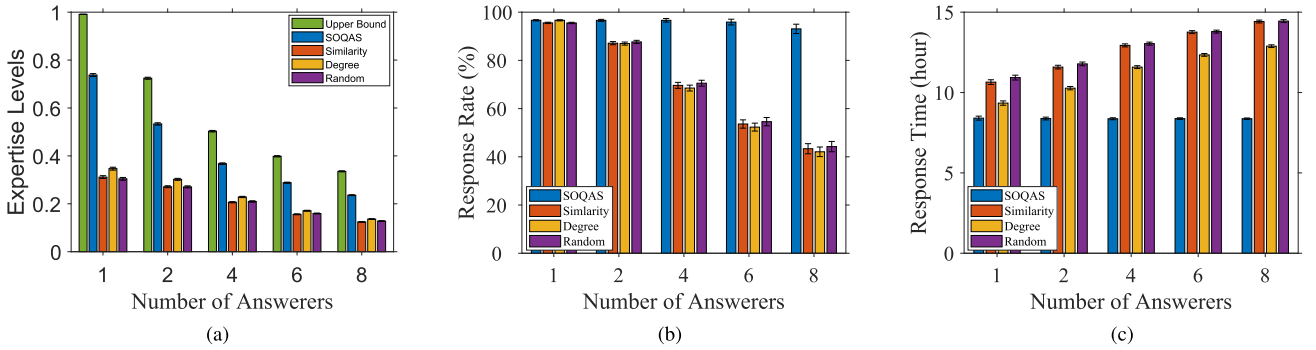

FIGURE 10. SOQAS is robust under various numbers of answerers: (a) overall results of expertise levels, (b) overall results of response rate, and (c) overall results of response time.

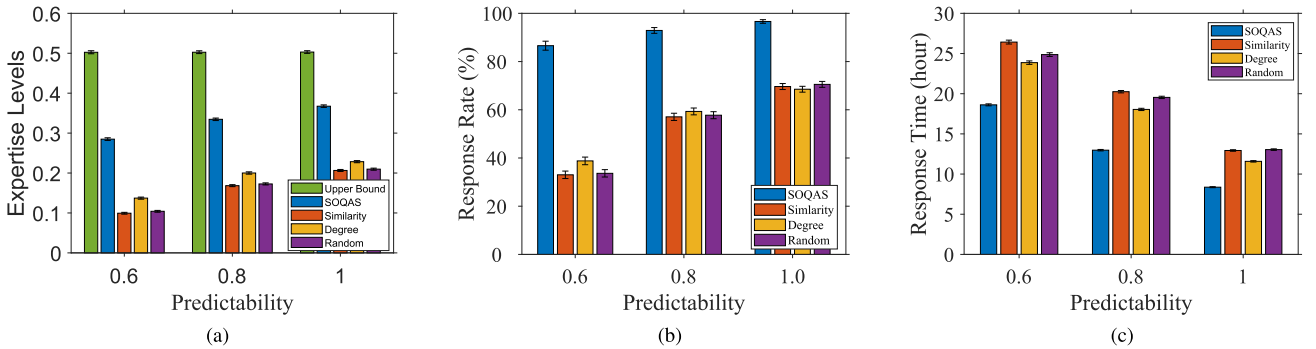

FIGURE 11. SOQAS is robust under various values of predictability: (a) overall results of expertise levels, (b) overall results of response rate, and (c) overall results of response time.

5) SOQAS IS ROBUST UNDER HIGHER LOADS

Since SOQAS supports several system parameters, we show its performance under different number of answerer and predictability values in Fig. 10 and Fig. 11, respectively. We do not report the results for parameters of keywords per question and question arrival rate, because we find that their average expertise levels, response rate, and response time do not change for different values of these parameters. Fig. 10a shows the expertise levels which follows the trend: SOQAS > Degree > Similarity ≅ Random. The expertise levels decrease as the number of answerers increases. This is because the expertise levels are averaged with increased number of answerers. Fig. 10b shows response rate with trend: SOQAS > Similarity ≅ Random > Degree. The figure shows that SOQAS has 45.9%, 48.9%, and 43.9% higher response rate than the other schemes, respectively, when the number of answerers is eight. Fig. 10c shows the response time which follows the trend: SOQAS < Degree < Similarity ≅ Random, for all numbers of answerers. It can be seen that SOQAS has 41.6%, 34.9%, and 42.5% lower response time than Similarity (14.2 hours), Degree (12.9 hours), and Random (14.6 hours), respectively, when the number of answerers is eight.

Fig. 11a reports the expertise levels which follows the trend: SOQAS > Degree > Similarity ≅ Random, for all values of predictability. Overall, the expertise levels decrease

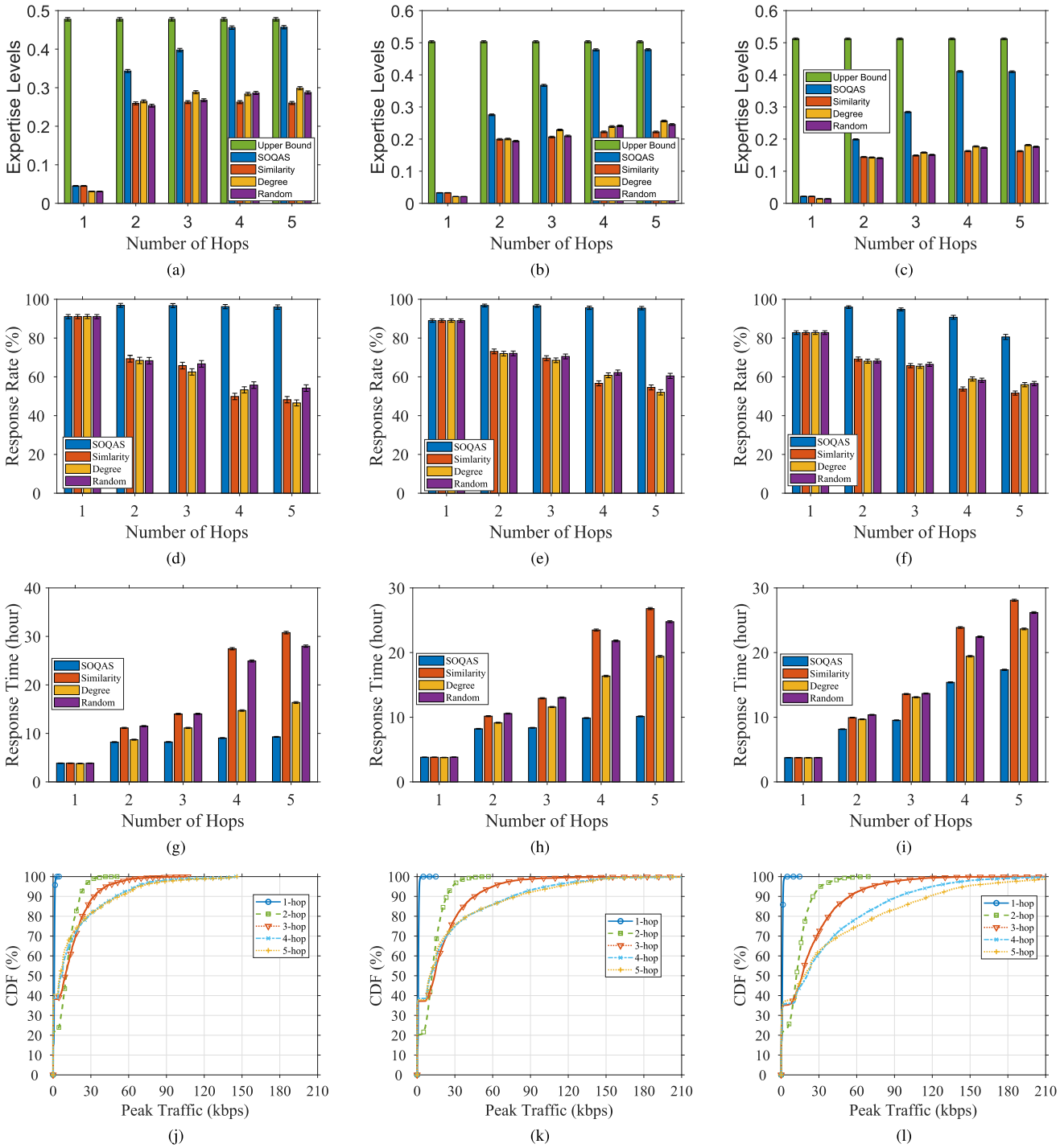

FIGURE 12. SOQAS performs efficiently under various number of hops: (a), (b), and (c) show overall expertise levels, (d), (e), and (f) show overall response rate, (g), (h), and (i) show overall response time, and (j), (k), and (l) show peak overhead, for small-, medium-, and large-sized networks, respectively.

with decreased values of predictability, because predictability affect the users online times, where social referral chains to answerers may get disconnected, and thus the average expertise levels decrease. However, SOQAS expertise levels of 0.28 is still higher by 64.2%, 50%, and 64.2% than Similarity (0.10), Degree (0.14), and Random (0.10), respectively, at the lowest predictability value of 0.6. Fig. 11b shows the response rate which follows the trend: SOQAS > Degree > Similarity $\cong$ Random. The response rate decreases due to unavailability of selected relays for the social referral chains. However, SOQAS response rate is higher by 33.1%, 38.7%, and 33.3% than Similarity, Degree, and Random,

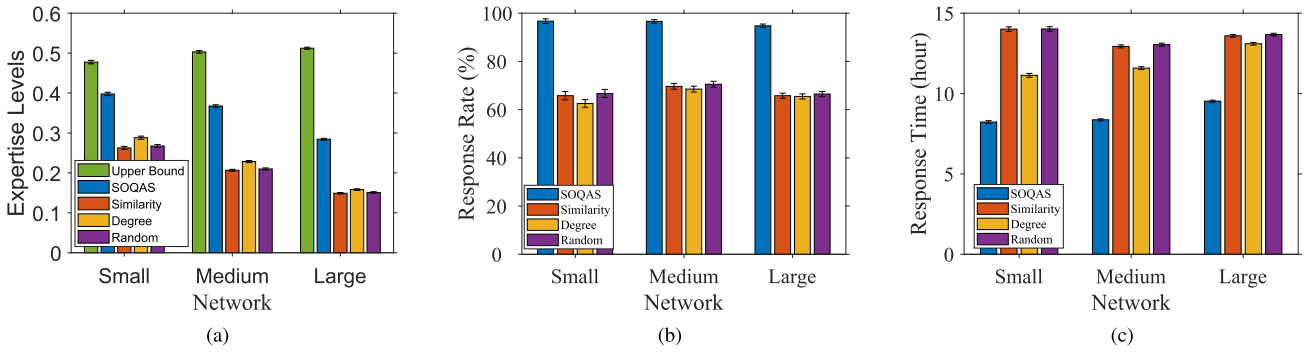

FIGURE 13. SOQAS is scalable under different networks sizes: (a) over all results of expertise levels, (b) over all results of response rate, and (c) over all results of response time.

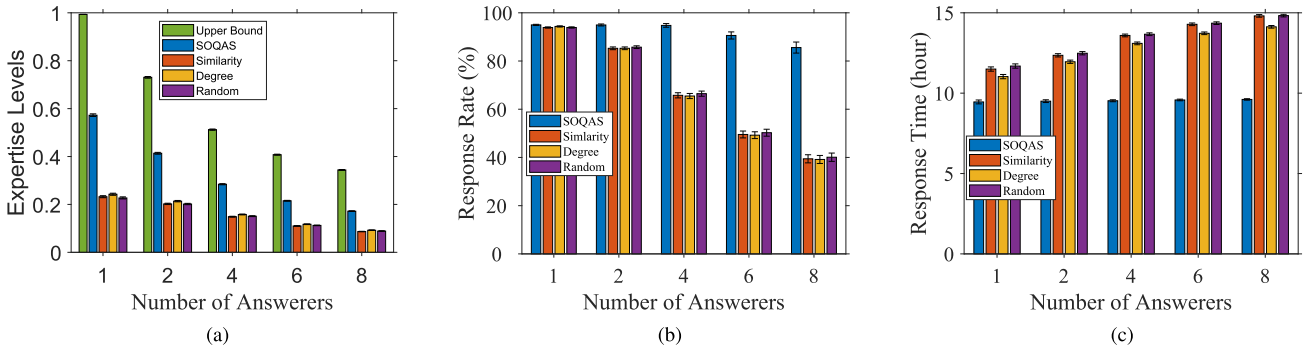

FIGURE 14. SOQAS performance on the large-sized network under various numbers of answerers: (a) overall results of expertise levels, (b) overall results of response rate, and (c) overall results of response time.

respectively, at lowest predictability of 0.6. Fig. 11c shows the response time with the trend: SOQAS < Degree < Random ≅ Similarity. SOQAS response time is lower by 29.6%, 21.9%, and 24.4% than Similarity, Degree, and Random, respectively.

6) SOQAS PERFORMS EFFICIENTLY UNDER VARIOUS NUMBER OF HOPS

SOQAS' performance for different number of hops is shown in Fig. 12. Fig. 12a–Fig. 12c show the expertise levels against various number of hops for the three networks, respectively. The figures show the trend: SOQAS > Degree > Similarity ≅ Random, for the number of hops being greater than one. When the number of hops increases, SOQAS achieves expertise levels comparable to upper bound. SOQAS' expertise levels for 4 and 5-hops have very slight difference because SOQAS covers most of the users' information in 4-hops. Fig. 12d–Fig. 12f show the response rate with the trend: SOQAS > Degree > Similarity ≅ Random, for the number of hops being greater than one. When the number of hops is one, all schemes have the same response rates because all select answerers among 1-hop neighbors. SOQAS's response rate increases with the number of hops because of k-hop neighbors availability. The slight decrease observed in the large-sized network is because of unavailability of answerers with nonzero expertise levels.

Fig. 12g–Fig. 12i show the response time for small-, medium-, and large-sized networks, respectively, that follow the trend: SOQAS < Degree < Random ≅ Similarity, for all number of hops. When the number of hops is one, all schemes have the same average response time because all schemes select answerers among the 1-hop neighbors. The response time increases with an increase in the number of hops, because each question has to go to k hops along the social referral chain. The slight increase in large-sized network's response time is attributed to low average online time as compared to other two networks. Fig. 12j–Fig. 12l show the CDF of peak traffic for SOQAS for small-, medium-, and large-sized networks, respectively. It can be seen that most links have low peak traffic for all networks. However, the peak traffic slightly increases when the number of hops increases.

7) SOQAS SCALES WELL UNDER DIFFERENT NETWORKS SIZES

SOQAS's potentials under different network sizes is shown in Fig. 13. Fig. 13a shows the expertise levels which follows the trend: SOQAS > Degree > Similarity ≅ Random, for all three networks. A close inspection reveals that most

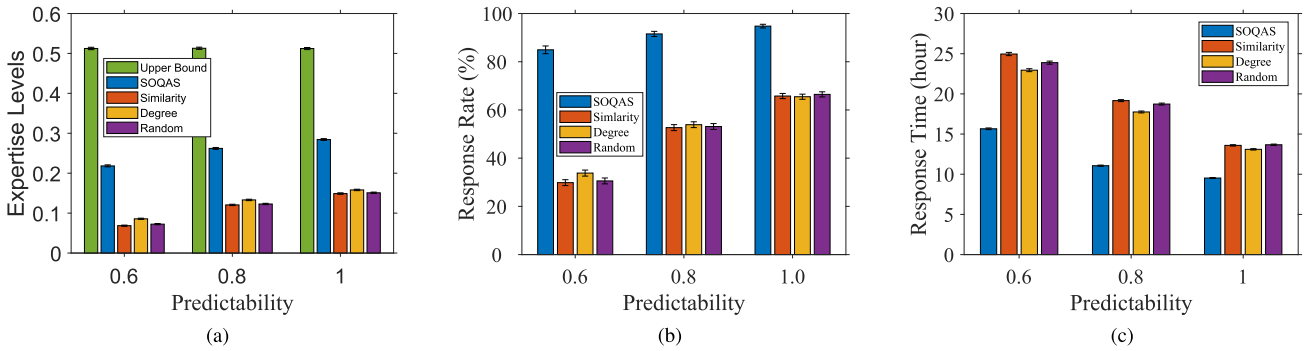

FIGURE 15. SOQAS performance on the large-sized network under various values of predictability: (a) overall results of expertise levels, (b) overall results of response rate, and (c) overall results of response time.

of the users' information in small-sized network is covered by SOQAS in 3-hops, while as the network size increases, large fractions of users are not covered. This explains why the average expertise levels decrease when the network size increases. Fig. 13b shows the response rate with the trend: SOQAS > Similarity $\cong$ Random > Degree. The response rate is not affected much with different network sizes because the answerers are still searched within 3-hop networks. Fig. 13c shows the response time which follows the trend: SOQAS < Degree < Similarity $\cong$ Random. The slight increase in large-sized network response time is due to the user average online time being slightly smaller than that in small and medium-sized networks.

To see how SOQAS performs on the large-sized network of 1252 users, we show its parametric results. Fig. 14 and Fig. 15 reveal that the trends in the number of answerers and predictability parameters for the large-sized network are similar to Fig. 10 and Fig. 11 for the medium-sized network, respectively. Our extensive experiments on the small-sized network show similar trends in all parametric results as in medium- and large-sized networks, and therefore are not reported.

VI. CONCLUSION

In this article, we present SOQAS, which is a distributed question answering system for finding high-quality answerers. SOQAS leverages the properties of dynamic social networks to relay a question via neighbors to answerers. To find high-quality answerers and forward the question to them, SOQAS uses two protocols that allow social network users to exchange information so as to know each other's expertise levels which are disseminated to k-hop neighbors with a moderate overhead. SOQAS efficiently finds high-quality answerers among the k-hop neighbors. To forward the question to the identified high-quality answerers via social referral chains, SOQAS selects optimal relays at each hop. Our extensive trace-driven experiments show that SOQAS outperforms other state-of-the-art schemes. In particular, SOQAS achieves: (i) higher average expertise levels by more than 42%, (ii) higher average response rate by more than 26%, and (iii) lower response time as high as 27% reduction. Furthermore, the results show that under diverse system parameters such as question arrival rate, keywords per question, answerers per question, number of hops, and predictability, SOQAS consistently outperforms the state-of-the-art schemes.

Future work includes incorporating a privacy-preserving mechanism to ensure user privacy during exchanges of information, conducting experiments via graduate students to evaluate the accuracy of answerers' expertise levels, developing an incentive mechanism to motivate users in forwarding or answering questions, and ultimately testing and releasing a real-world SOQAS.

REFERENCES

[1] M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub, "Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms," *Inf. Syst.*, vol. 56, pp. 1–18, Aug. 2016.

[2] G. Dror, Y. Koren, Y. Maarek, and I. Szpektor, "I want to answer; who has a question?: Yahoo! Answers recommender system," in *Proc. ACM SIGKDD*, Aug. 2011, pp. 1109–1117.

[3] *Yahoo!Answers*. Accessed: Feb. 2017. [Online]. Available: https://answers.yahoo.com/

[4] *Qoura*. Accessed: Feb. 2017. [Online]. Available: https://www.quora.com/

[5] *Answers.com*. Accessed: Feb. 2017. [Online]. Available: http://www.answers.com/

[6] *Stack Overflow*. Accessed: Feb. 2017. [Online]. Available: https://stackoverflow.com/

[7] M. R. Morris, J. Teevan, and K. Panovich, "A comparison of information seeking using search engines and social networks," in *Proc. ICWSM*, May 2010, pp. 23–26.

[8] D. Horowitz and S. D. Kamvar, "The anatomy of a large-scale social search engine," in *Proc. 19th Int. Conf. World Wide Web*, Apr. 2010, pp. 431–440.

[9] H. Shen, Z. Li, G. Liu, and J. Li, "SOS: A distributed mobile Q&A system based on social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 1066–1077, Apr. 2014.

[10] E. Pennisi, "How did cooperative behavior evolve?" *Science*, vol. 309, no. 5731, p. 93, Jul. 2005.

[11] G. Liu and H. Shen, "iASK: A distributed Q&A system incorporating social community and global collective intelligence," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1–14, May 2016.

[12] L. Zhang, X.-Y. Li, J. Lei, J. Sun, and Y. Liu, "Mechanism design for finding experts using locally constructed social referral Web," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2316–2326, Aug. 2015.

[13] I. Ali, R. Y. Chang, J.-C. Chuang, C.-H. Hsu, and C. M. Yetis, "Optimal question answering routing in dynamic online social networks," in *Proc. IEEE VTC*, Sep. 2017, pp. 1–7.

[14] Z. Zhao, L. Zhang, X. He, and W. Ng, "Expert finding for question answering via graph regularized matrix completion," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 4, pp. 993–1004, Apr. 2015.
[15] L. Nie, X. Wei, D. Zhang, X. Wang, Z. Gao, and Y. Yang, "Data-driven answer selection in community QA systems," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1186–1198, Jun. 2017.
[16] Z. Zhao, H. Lu, V. W. Zheng, D. Cai, X. He, and Y. Zhuang, "Community-based question answering via asymmetric multi-faceted ranking network learning," in *Proc. AAAI*, Feb. 2017, pp. 3532–3539.
[17] I. Srba and M. Bielikova, "A comprehensive survey and classification of approaches for community question answering," *ACM Trans. Web*, vol. 10, no. 3, pp. 1–63, Aug. 2016.
[18] S. A. Paul, L. Hong, and E. H. Chi, "Is Twitter a good place for asking questions? A characterization study," in *Proc. ICWSM*, Jul. 2011, pp. 578–581.
[19] R. W. White, M. Richardson, and Y. Liu, "Effects of community size and contact rate in synchronous social Q&A," in *Proc. ACM SIGCHI*, May 2011, pp. 2837–2846.
[20] L. Soulier, L. Tamine, and G.-H. Nguyen, "Answering Twitter questions: A model for recommending answerers through social collaboration," in *Proc. ACM CIKM*, Oct. 2016, pp. 267–276.
[21] B. J. Hecht, J. Teevan, M. R. Morris, and D. J. Liebling, "SearchBuddies: Bringing search engines into the conversation," in *Proc. ICWSM*, Jun. 2012, pp. 138–145.
[22] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proc. ACM SIGKDD*, Jun. 2009, pp. 467–476.
[23] S. Jiang, L. Guo, X. Zhang, and H. Wang, "LightFlood: Minimizing redundant messages and maximizing scope of peer-to-peer search," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 601–614, May 2008.
[24] G. Kukla, P. Kazienko, P. Bródka, and T. Filipowski, "SocLaKE: Social latent knowledge explorator," *Comput. J.*, vol. 55, no. 3, pp. 258–276, Sep. 2011.
[25] Y. Lin and H. Shen, "SmartQ: A question and answer system for supplying high-quality and trustworthy answers," in *Proc. IEEE Trans. Big Data*, Aug. 2017, pp. 744–751.
[26] H. Amintoosi and S. S. Kanhere, "A trust-based recruitment framework for multi-hop social participatory sensing," in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2013, pp. 266–273.
[27] L. Guo, C. Zhang, and Y. Fang, "A trust-based privacy-preserving friend recommendation scheme for online social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 413–427, Jul. 2015.
[28] H. Shen, G. Liu, H. Wang, and N. Vithlani, "SocialQ&A: An online social network based question and answer system," *IEEE Trans. Big Data*, vol. 3, no. 1, pp. 91–106, Mar. 2017.
[29] C.-Y. Lin, K. Ehrlich, V. Griffiths-Fisher, and C. Desforges, "SmallBlue: People mining for expertise search," *IEEE Multimedia Mag.*, vol. 15, no. 1, pp. 78–84, Jan. 2008.
[30] G. Chen, C. P. Low, and Z. Yang, "Enhancing search performance in unstructured P2P networks based on users' common interest," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 6, pp. 821–836, Jun. 2008.
[31] K. C.-J. Lin, C.-P. Wang, C.-F. Chou, and L. Golubchik, "SocioNet: A social-based multimedia access system for unstructured P2P networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 7, pp. 1027–1041, Jul. 2010.
[32] H. Shen, Z. Li, and K. Chen, "Social-P2P: An online social network based P2P file sharing system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 2874–2889, Oct. 2015.
[33] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao, "Spammer behavior analysis and detection in user generated content on social networks," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2012, pp. 305–314.
[34] X. Liu, W. B. Croft, and M. Koll, "Finding experts in community-based question-answering services," in *Proc. ACM CIKM*, Oct. 2005, pp. 315–316.
[35] L. Chen and R. Nayak, "Expertise analysis in a question answer portal for author ranking," in *Proc. IEEE/WIC/ACM WI*, Dec. 2008, pp. 134–140.
[36] O. Rottenstreich, Y. Kanizo, and I. Keslassy, "The variable-increment counting bloom filter," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1092–1105, Aug. 2014.
[37] L. Zhang, X.-Y. Li, K. Liu, T. Jung, and Y. Liu, "Message in a sealed bottle: Privacy preserving friending in mobile social networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1888–1902, Sep. 2015.
[38] M. Li, S. Yu, N. Cao, and W. Lou, "Privacy-preserving distributed profile matching in proximity-based mobile social networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2024–2033, May 2013.
[39] R. Zhou, K. Hwang, and M. Cai, "GossipTrust for fast reputation aggregation in peer-to-peer networks," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1282–1295, Sep. 2008.
[40] A. Huang, "Similarity measures for text document clustering," in *Proc. NZCSRSC*, Apr. 2008, pp. 49–56.
[41] D. M. Christopher, R. Prabhakar, and S. Hinrich, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008, p. 504.
[42] S. Zhu, J. Wu, H. Xiong, and G. Xia, "Scaling up top-K cosine similarity search," *Data Knowl. Eng.*, vol. 70, no. 1, pp. 60–83, Jan. 2011.
[43] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Dept. Comput. Sci., Duke Univ., Durham, NC, USA, Tech. Rep. CS-200006, Apr. 2000.
[44] *Octoparse Home Page*. Accessed: Feb. 2017. [Online]. Available: http://www.octoparse.com
[45] *Microsoft Azure Service Home Page*. Accessed: Feb. 2017. [Online]. Available: https://azure.microsoft.com
[46] *NS-3: Network Simulator 3*. Accessed: Feb. 2017. [Online]. Available: https://www.nsnam.org/
[47] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proc. ACM ICS*, Apr. 2002, pp. 84–95.
[48] *Akamai: State of the Internet*. Accessed: Feb. 2017. [Online]. Available: https://www.akamai.com/fr/fr/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-connectivity-report.pdf

IMAD ALI received the B.S. degree in telecommunication engineering from the University of Engineering and Technology, Mardan Campus, Pakistan, in 2008, and the M.S. degree in electrical engineering from the CECOS University of Information Technology and Emerging Sciences, Peshawar, Pakistan, in 2011. He is currently pursuing the Ph.D. degree with the Taiwan International Graduate Program in Social Networks and Human-Centered Computing, Institute of Information Science, Academia Sinica, Taipei, Taiwan, and the Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan. His research interests include question answering systems and social network analysis.

RONALD Y. CHANG (M'12) received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2000, the M.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2002, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2008. From 2002 to 2003, he was with the Industrial Technology Research Institute, Hsinchu. In 2008, he was a Research Intern at the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. In 2009, he was involved in the NASA Small Business Innovation Research projects. Since 2010, he has been with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, where he is currently an Associate Research Fellow (Associate Professor). His research interests include wireless communications and networking. He was a Visiting Scholar with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, in 2018. He was a recipient of the Best Paper Award from the IEEE Wireless Communications and Networking Conference in 2012 and the Outstanding Young Scholar Award from the Ministry of Science and Technology, Taiwan, in 2015 and 2017, respectively. He was an Exemplary Reviewer of the IEEE COMMUNICATIONS LETTERS in 2012, the IEEE TRANSACTIONS ON COMMUNICATIONS in 2015, and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS in 2017.

CHENG-HSIN HSU (S'09–M'10–SM'16) received B.S. and M.S. degrees from National Chung-Cheng University, the M.Eng. degree from the University of Maryland, and the Ph.D. degree from Simon Fraser University. In 2011, he joined the Department of Computer Science, National Tsing Hua University as an Assistant Professor and was promoted to an Associate Professor in 2014. Before accepting the teaching position, he was with the Deutsche Telekom Laboratory, CA, USA, Motorola Inc., IL, USA, and Lucent Technologies, MD, USA, for over six years. He has been a Visiting Scholar with the University of California Irvine (Summer 2013 and Summer 2018–Summer 2019), the Qatar Computing Research Institute (Summer 2014), and the University of Illinois Urbana–Champaign (Spring 2016).

His research interests are in the broad area of multimedia networking, mobile computing, broadcast/wireless networks, Internet-of-Things, networked games, cloud/fog computing, and computer networks. He and his colleagues received the Best Paper Award at the IEEE CloudCom'17, APNOMS'16, IEEE RTAS'12, and the IEEE Innovation'08, and the TAOS Best Paper Award at the IEEE GLOBECOM'12, and the Best Demo Award from ACM Multimedia'08. He was selected as one of the Multimedia Rising Stars by the ACM SIGMM in 2015, and he received the Best Associate Editor Award from the *ACM Transactions on Multimedia Computing, Communications, and Applications* (TOMM) in 2016. He helped organizing international conferences in various capacities, such as the Area Co-Chair at ACM Multimedia'17, the TPC Co-Chair at ACM MMSys'17, the Poster Co-Chair at the IEEE NOMS'18, and the Publicity Co-Chair at ACM MMSys'18. He has been an Associate Editor of TOMM since 2014. He was the IEEE MMTC E-Letter between 2012 and 2014.