Real-Time Community Question Answering: Exploring Content Recommendation and User Notification Strategies

Qiaoling Liu **Tomasz Jurczyk** **Jinho D. Choi** **Eugene Agichtein**
Mathematics and Computer Science Department, Emory University, USA
{qiaoling.liu, tomasz.jurczyk, jinho.choi, eugene.agichtein}@emory.edu

ABSTRACT

Community-based Question Answering (CQA) services allow users to find and share information by interacting with others. A key to the success of CQA services is the quality and timeliness of the responses that users get. With the increasing use of mobile devices, searchers increasingly expect to find more local and time-sensitive information, such as the current special at a cafe around the corner. Yet, few services provide such hyper-local and time-aware question answering. This requires intelligent content recommendation and careful use of notifications (e.g., recommending questions to only selected users). To explore these issues, we developed *RealQA*, a real-time CQA system with a mobile interface, and performed two user studies: a formative pilot study with the initial system design, and a more extensive study with the revised UI and algorithms. The research design combined qualitative survey analysis and quantitative behavior analysis under different conditions. We report our findings of the prevalent information needs and types of responses users provided, and of the effectiveness of the recommendation and notification strategies on user experience and satisfaction. Our system and findings offer insights and implications for designing real-time CQA systems, and provide a valuable platform for future research.

Author Keywords

Real-time; community; question answering; recommendation; mobile notification;

INTRODUCTION

Community-based Question Answering (CQA) services such as Yahoo! Answers [11] and Quora [8] provide an increasingly effective way of finding and sharing information online. Users turn to community help for various reasons, e.g., unsuccessful web searches or a need for answers from real humans. A key to the success of CQA services is the quality and timeliness of the responses that the users get. To reduce the response latency between questions and answers, several real-time question-answering systems have been introduced [18, 30, 36]. To improve the response quality, question recommendation has been explored [15, 34]. At the same time, as the use of mobile devices has increased, the information needs related to the content near the user's location have also become more prevalent [35]. To satisfy such location-related needs, location-based CQA systems have been introduced [4, 5, 27].

Although both real-time and location-based aspects are important, we are not aware of CQA systems that support both aspects. To investigate the interplay of relevance, location, and timeliness in real-time CQA, we built a research system called *RealQA*, utilizing a mobile application supporting instant notification and location detection, and a server backend handling recommendation and notification strategies. Our system supports common CQA functions such as asking and answering questions, voting for questions and answers, and retrieving a list of questions to answer. We also investigate novel features specifically designed for real-time receiving recommendation notifications of newly posted questions, and receiving new answers for subscribed questions. User locations are recorded while they interact with the system, which are later used for real-time question recommendation.

Our system was developed iteratively, incorporating insights from the analysis of two user studies - the pilot study and the main study, conducted with students at Emory University. We focused on evaluating the overall functionality of our system during the pilot study, and on comparing different recommendation algorithms during the main study. Specifically, we investigated: 1) the types of questions asked and quality of answers; 2) user preference for question recommendation strategies: letting users pull a list of questions from the main page vs. pushing questions to the users via notifications; 3) the effectiveness of question-ranking and user-ranking algorithms for different recommendation strategies; 4) the effectiveness of automated question tag recommendation algorithms.

The specific contributions of our work include:

- A novel, location-aware real-time CQA system named RealQA as a research platform (Section 3).
- Two user studies with two versions of the system, showing improvements with both qualitative survey analysis and quantitative behavior analysis (Section 4).
- Comparison of different algorithms for question ranking, user ranking, and tag ranking (Section 4).
- Insights, data, and code (`http://ir.mathcs.emory.edu/projects/realqa/`) for future studies (Section 5).

IUI 2015, March 29–April 1, 2015, Atlanta, GA, USA.
Copyright © 2015 ACM 978-1-4503-3306-1/15/03 ...$15.00.
http://dx.doi.org/10.1145/2678025.2701392

RELATED WORK

Real-time question answering

Real-time question answering (QA) systems have been designed to shorten the time for a question to be answered. These systems typically use synchronous communication channels for asking and answering questions. For examples, Aardvark automatically routes a question to people in the asker's extended social network who are most likely to answer [18]. The user ranking algorithm considers user's expertise in the question topic, connectedness to the asker, and availability to respond. IM-an-Expert provides an instant message service deployed in an organization to find experts for a question and automatically create dialog sessions for the asker [30, 36]. The expert ranking algorithm is based on matching question text with user profiles using TF-IDF, an established method in information retrieval. Mimir, a market-based real-time QA system, broadcasts a question to all other users [19].

Some systems allow users to locate potential responders for new questions.For example, the Quora service called "Online Now" enables an asker to find a list of experts who are currently online, to choose whom to ask [9]. [26] proposed a system which can find a set of Twitter friends for a query based on availability, willingness, and knowledge. [24] presented methods to locate targeted strangers on Twitter for information solicitations.

Our work is most relevant to Aardvark. Table 1 compares the statistics reported in [18] and the statistics collected in our main study, although the comparison is not quite fair due to different deployment settings and participation incentives. We found that the nature of questions collected by our system is more subjective and local-intent. Moreover, the proportion of users answered any questions (or recommended questions) is higher in our case. The main difference between the two systems is that Aardvark focuses more on social network while our system focuses more on location proximity when exploring recommendation and notification strategies.

	Aardvark	RealQA
% of subjective questions	64.7%	71%
% of questions with local intent	10%	77%
% of questions answered	87.7%	89.3%
% of users received rec. questions	86.7%	66%
% of users clicked rec. questions	70%	74%
% of users answered rec. questions	38%	48%
% of users answered any questions	50.0%	74.3%

Table 1: Comparative statistics for Aardvark and our system (RealQA).

Location-based question answering

Location-based QA systems have been designed to facilitate the information seeking and knowledge sharing about some geographic locations [4, 5, 27]. Typically, these systems allow users to post questions to users around a specific geographic location. Our system provides similar features, i.e., when posting a question, users can select "asking people around a specific location". Then, the question recommendation algorithm uses this location information to recommend the question to users who are not only interested in answering but also close to that location.

Mobile-based question answering

Asking and answering questions from mobile devices have become increasingly popular [20, 28]. Most web QA services also provide mobile-friendly websites or mobile apps such as Yahoo! Answers [11], Quora [8], and Stack Overflow [10].

The portability of mobile devices also makes accurate location detection and real-time interaction with users easier. This motivates our system to support real-time and location-aware question answering services based on a mobile app. Recently, [29] studied the effect of mobile phone notifications on the daily lives of mobile users, and showed that an increasing number of notifications correlated with negative emotions. From our study, we found that carefully sending notifications to users and let users have control resulted in better system performance and user satisfaction. [29] also found that silent notifications were not viewed slower than non-silent ones, which supports our design decision to send silent notifications for recommendations.

Question recommendation

There are two ways for users to retrieve questions: proactively request questions from QA services (PULL), or let the system push questions to them (PUSH). Accordingly, question recommendation can be done using two strategies. The PULL strategy computes a list of questions for a user to answer, e.g., by considering question content signals and user social signals [15], and question relevance, diversity, and freshness [34]. The PUSH strategy computes a list of users to route a question for getting answers, e.g., by considering user interests [17], authority [22], availability [21], and compatibility [14]. Real-time QA systems often use this strategy [18, 36]. A detailed survey of the question recommendation methods can be found in [16].

To the best of our knowledge, the only work supporting both the PULL and PUSH strategies in a single system is Aardvark [18]. In their study, more users answered via PUSH because users were willing to answer questions to help friends in their social networks. However, in our studies, PULL is preferred, when a closely knit social network may not exist.

Tag recommendation

A question in a CQA service may span multiple topics or be interesting to different users. Tags have become a de-facto standard way to organize such content in a meaningful way, but are not always provided by the users. Automatic tag recommendation has been studied for a variety of content (e.g., blogs [32], micro-blogs [23, 37], discussion forums [33]), and social recommender systems have been introduced [31]. We adapted and compared several methods in our system, including a method similar to the one in [37] in order to automatically suggest tags for users' questions.

SYSTEM OVERVIEW

Our system consists of two parts: a front-end mobile application and a back-end server system. Users can post, upvote, and downvote questions and answers using the mobile app. From the main page of the mobile app, users can browse questions recommended by our server system. Users may also get questions recommended via mobile notifications. Moreover, users can subscribe to questions they want to get notifications about when new answers are posted for the questions.

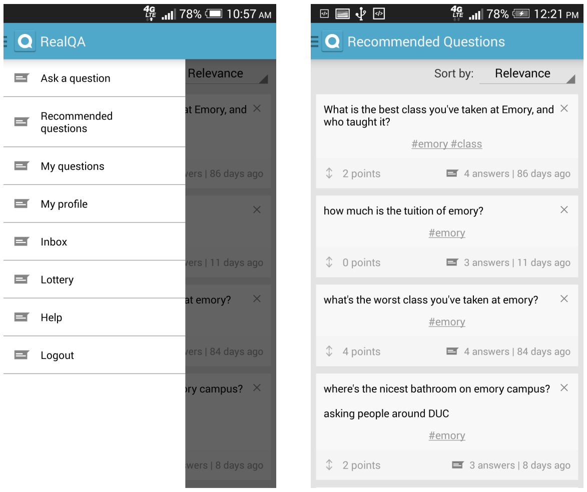
Our system was developed iteratively, incorporating insights from the analysis of two user studies. In both studies, we analyzed users' interactions with our system and their survey responses. From the pilot study, we received user feedback and improved our system for both the front-end mobile application and the back-end sever system based on their feedback. We also collected feedback from users from the main study, which will be used for future improvement. Table 2 shows main functions and comparisons between systems used for the pilot and the main studies.

Front-end: mobile application

Our mobile application has been designed and built for smart phones using an Android OS (version 4.0 or higher, occupying 87.9% of the whole Android device market). Our application is distributed through the official Android distribution platform, Google Play (`http://play.google.com`).

Registration, login, and navigation

During the registration process, each user is asked to enter a username, a password, and tags of interests. After registration, the user is logged into the system, and directed to the main page. The *navigation drawer* is used to view different pages in the application, and can be prompted when the user presses the application icon in the top-left corner (Figure 1a).



What is the best class you've taken at Emory, and who taught it?
#emory #class
2 points 4 answers | 86 days ago

how much is the tuition of emory?
#emory
0 points 3 answers | 11 days ago

what's the worst class you've taken at emory?
#emory
4 points 4 answers | 84 days ago

where's the nicest bathroom on emory campus?
asking people around DUC
#emory
2 points 3 answers | 8 days ago

(a) Navigation drawer.

(b) Recommended questions.

Figure 1: The main page of our mobile application.

Application main page

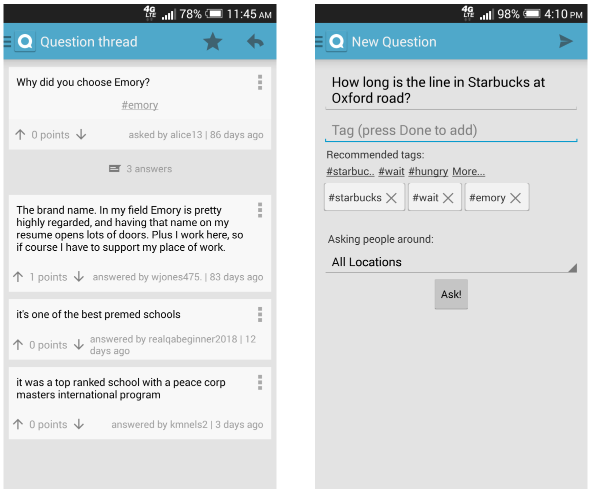
By default, the application main page shows a list of questions recommended to this user (Figure 1b). Each row consists of a question body, tags related to the question, the score of the question, the number of answers to the question, and the time since question is posted. The cross icon in the upper right corner of each question item can be used to dismiss this question.

By clicking the choice box in the upper right corner, users can choose different question ranking algorithms (see the 'server' section for more details about the algorithms). It also gives an option of retrieving all questions. Any click on either the question body or the number of answers displays the *question thread*, consisting of the question and all answers that have been posted for the question (Figure 2a). Additionally, two buttons are placed in the upper right corner; the left button lets the user subscribe to this question, and the right button lets the user post an answer to this question.

Posting a question

A key aspect of our system is the ability to post a new question in real-time (Figure 2b). When the user enters a question body, the system shows 3 recommended tags and an option to retrieve 10 recommended tags based on the question body (see the 'server' section for more details about the recommended tags).

Another important aspect is that our system allows the user to post a question to only people around a specific location. By default, each question is asked to people in all locations with no specific location annotated. If the user chooses "my current location", the question is annotated with the user's current location (using latitude and longitude). Other locations can be also chosen, including 21 popular locations on campus such as food courts or residence halls, in which case, the question is annotated with this particular location.



Why did you choose Emory?
#emory
0 points asked by alice13 | 86 days ago
3 answers

The brand name. In my field Emory is pretty highly regarded, and having that name on my resume opens lots of doors. Plus I work here, so if course I have to support my place of work.
1 points answered by wjones475. | 83 days ago

it's one of the best premed schools
0 points answered by realqabeginner2018 | 12 days ago

it was a top ranked school with a peace corp masters international program
0 points answered by kmnels2 | 3 days ago

How long is the line in Starbucks at Oxford road?
Tag (press Done to add)
Recommended tags:
#starbuc.. #wait #hungry More...
#starbucks × #wait × #emory ×
Asking people around:
All Locations
Ask!

(a) Question thread.

(b) Question post.

Figure 2: Question thread and question post.

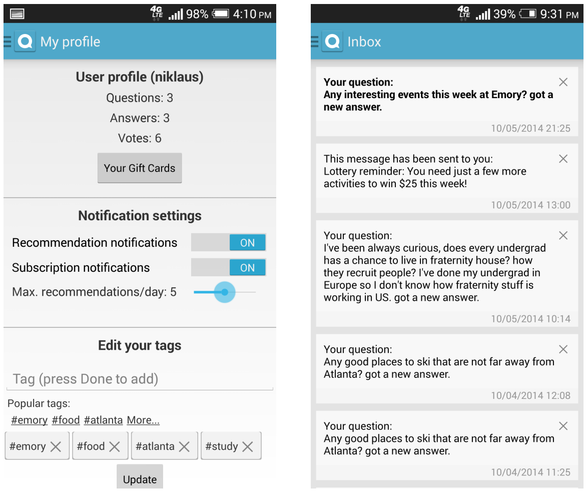
By clicking on either the *Ask* button or the right upper corner icon, the question gets posted. The author is automatically subscribed to this question, and receives a notification whenever a new answer is posted for this question.

	Pilot Study	**Main Study**
User related		
· Getting notifications (e.g., question recommendations, answer updates)	2 recommendation algorithms with sound & vibration	3 recommendation algorithms silent notifications
· Updating user's tags of interest	Y	Y
· Turning off notifications	Y	Y
· Setting maximum # of recommendations per day	**N**	**Y**
· Managing user's notifications in the inbox	**N**	**Y**
· Participating in the weekly lottery	**N**	**Y**
Question related		
· Browsing recommended questions in the main page	1 assigned ranking algorithm	5 selectable algorithms
· Browsing all questions in the system	via navigation menu	via ranking options in main page
· Subscribing questions automatically	for asked/answered questions	for asked questions only
· Subscribing questions manually	Y	Y
· Browsing user's asked/answered/subscribed questions	Y	Y
· Asking questions (optionally, annotating locations)	Y	Y
· Answering questions	Y	Y
· Voting for questions and answers	Y	Y
· Getting recommended tags based on question text	**N**	**Y**
· Dismissing questions from the main page	**N**	**Y**

Table 2: Functions in our systems used for the pilot and the main studies.

User profile and notification inbox

Users can update their profiles from the *user profile* tab consisting of user statistics, notification settings, and tags settings (Figure 3a). From the notification settings, users can control whether to receive subscription notifications (when a subscribed question receives a new answer) and recommendation notifications (when a newly-posted question is recommended for the user to answer), as well as the number of maximum recommendations that users receive per day (the default is 5).



User profile (niklaus)

Questions: 3

Answers: 3

Votes: 6

Notification settings

Recommendation notifications ON

Subscription notifications ON

Max. recommendations/day: 5

Edit your tags

Popular tags:
#emory #food #atlanta More...

(a) User profile.

Your question:
Any interesting events this week at Emory? got a new answer.
10/05/2014 21:25

This message has been sent to you:
Lottery reminder: You need just a few more activities to win $25 this week!
10/05/2014 13:00

Your question:
I've been always curious, does every undergrad has a chance to live in fraternity house? how they recruit people? I've done my undergrad in Europe so I don't know how fraternity stuff is working in US. got a new answer.
10/05/2014 10:14

Your question:
Any good places to ski that are not far away from Atlanta? got a new answer.
10/04/2014 12:08

Your question:
Any good places to ski that are not far away from Atlanta? got a new answer.
10/04/2014 11:25

(b) Notification inbox.

Figure 3: User profile and notification inbox.

The *notification inbox* tab contains notifications that have been sent to the user (Figure 3b). The user can view the associated question by clicking on any notification. Notifications that have not been clicked are shown in bold.

Miscellaneous

The *my questions* tab is similar to the *recommended questions* in Figure 1b, except that the dismissal icon is not present and the right upper choice box include options: asked, answered, subscribed. The *lottery* tab provides information about the status of weekly lotteries. The *help* tab lets users see the terms of service, run the tutorial, or report bugs.

Back-end: server system

For the implementation of the server system, we adapted the Django REST framework [1] and extended the Django models from the Open Source Q&A system (OSQA [7]).

Search component

We adapted Haystack [3] and Elasticsearch [2] for implementing our search component: Haystack provides modular search for Django and Elasticsearch provides scalable real-time indexing and search. Adapting these tools allows our system to create indices for questions, users, and tags dynamically, as well as to search them in real time. Each document in our index contains two fields: a question text field and a tag field. Detailed content of each type of documents is provided in Table 3. We used field level boosting to weigh the tag field twice more than the question text field. We also applied decapitalization, stop word removal, and stemming to the texts in documents and queries.

Document Type	**Question Text Field and Tag Field**
question	· Question body · Tags of this question
user	· Question bodies answered by this user. · Tags of questions answered by this user, and tags pre-entered in user's profile.
tag	· Question bodies with this tag. · The name of this tag.

Table 3: Index structure for questions, users, and tags.

Our system uses a vector space model for measuring the relevance between documents and queries. All documents and queries are represented in a multi-dimensional vector space, where each dimension stands for a unique term. The relevance between a document and a query is measured by Lucene's practical scoring function [6]: $score(q, d) = \sum_{t \in q}(tf(t, d) \cdot$

$idf(t)^2 \cdot norm(t,d)) \cdot\ queryNorm(q) \cdot coord(q,d)$, where $queryNorm(q)$ is the query normalization factor, $coord(q,d)$ is a score factor based on how many of the query terms are found in document d, $tf(t \in d)$ is the term frequency for term t in document d, $idf(t)$ is the inverse document frequency for term t, $norm(t,d)$ is the field length norm, combined with the index-time field-level boost.

Question ranking

Our system offers 5 different options for ranking questions when a pull (of questions) is requested by a user:

- **Relevance** - questions that are more relevant to the user's interests are ranked higher. To find relevant questions, search queries are extended with contents in both question text field and tag field of user document, which provide information about the user's interests. This is the default option.
- **Freshness** - more recent questions are ranked higher.
- **Location** - if questions are annotated with locations, ones closer to the user's current location are ranked higher.
- **Popularity** - questions viewed by more unique users are ranked higher.
- **Answer count** - questions with a fewer number of answers are ranked higher.

Previous studies have shown that factors such as relevance and freshness are important for question recommendation [34]. During the pilot study, our system assigned a random ranking algorithm to each user; however, it lets the users choose the ranking algorithm during the main study, which is an improvement made from the analysis of our pilot study.

User ranking

Given a newly posted question, our system randomly assigns one of the following algorithms for ranking users to send recommendation notifications for answering this question:

- **Matching questions** - This algorithm first finds the top 20 questions in the history that are similar to the new question using our search component, then ranks the answerers of these questions. The score of each answerer is measured by the sum of similarity scores between the new question and questions previously answered by the user [17].
- **Matching users** - This algorithm uses the new question as a query against the user documents, and returns the top ranked users using our search component [17, 36].
- **Location proximity** - This algorithm computes the distance between user's and the question locations, and returns the users with closest distances. This algorithm is used for only questions that are annotated with a specific location.

The top 5 ranked users will be sent recommendation notifications for the question. Our system during the pilot study always used the location proximity strategy for questions annotated with specific locations, and the matching users strategy for the rest. During the main study, random algorithms are assigned to new questions to evaluate the differences between these strategies.

Tag recommendation ranking

When the user moves onto the tag input field after completing a new question, a list of recommended tags is displayed. The following algorithms are used for ranking these tags:

- **Matching questions** - This algorithm is similar to the user ranking one, except the score of each tag is measured by the sum of similarity scores between the new question and questions annotated with this tag [37].
- **Matching tags** - This algorithm uses the new question as a query against the tag documents, and returns the top ranked tags using our search component.
- **Tag popularity** - This algorithm ranks the tags based on their use frequencies. A tag is ranked higher if more questions are annotated with this tag.

When a new question is entered, the user is randomly assigned one of the first two ranking algorithms, which returns the top 10 ranked tags. If the number of ranked tags is smaller than 10, then the third algorithm is used to fill in the rest.

USER STUDIES

To evaluate our system for real-time question answering, we conducted two user studies. During the pilot study, we focused on evaluating the overall functionality of the system, while we focused on comparing different recommendation algorithms during the main study. The key aspects of our studies are:

- Types of questions and quality of answers.
- Preference of question recommendation strategies: letting users pull a list of questions in main page (PULL) vs. pushing questions to users via notifications (PUSH).
- Effectiveness of question ranking algorithms for PULL.
- Effectiveness of user ranking algorithms for PUSH.
- Effectiveness of tag recommendation algorithms.

Statistics	Pilot	Main
# of registered users	27	35
# of qualified users	14	16
# of all questions	120	56
# of questions annotated with locations	5	6
% of answered questions	83%	89%
# of all answers	244	238
# of upvotes for questions	76	76
# of upvotes for answers	134	155
# of downvotes for questions	15	5
# of downvotes for answers	33	26
Avg. # of tags in user profiles	3.6	4.5
Avg. # of tags for a question	1.9	2.3

Table 4: Statistics of the data collected from our user studies.

Any student with an Android phone was eligible to participate in these studies. Each participant was asked to install and use our mobile app for a limited period of time. The runtime, minimum required activities, compensation, and initial database were different between two studies. The pilot study was run for a week in July, 2014. In this study, participants received $10 gift cards if they met the following requirements: 1) posted at least 5 questions, 10 answers, 10 votes; 2) performed 1 activity (ask, answer, or vote) per day for 5 days; 3) completed the survey. The initial database included 7 questions posted by the authors of this paper.

The main study was run for 3 weeks in Sep. 2014. In this study, participants received $5 gift cards if they met the following requirements: 1) posted at least 1 question, 5 answers, 5 votes; 2) performed at least 1 activity per day for 3 days; 3) completed the survey. Moreover, we encouraged users to keep using our system for winning lotteries even after they received the gift cards; participants who completed the minimum required activities in a week were eligible to play the lottery held at the end of that week, and each winner received a $25 gift card. The initial database included most questions, answers, tags, votes from the pilot study (some noisy data were discarded).

Statistics and survey responses

In both studies, we analyzed users' interactions with the systems and their survey responses. Table 4 shows statistics from our studies. Qualified users are the ones who finished the minimum required activities. The percentage of qualified users is lower in the main study; one reason is that the pilot study was conducted during the summer when students had more free time, whereas the main study was conducted during the regular school year. The difference in the number of all questions is most likely due to the different minimum required activities in each study. The percentages of answered questions are both high and comparable to the one provided by Aardvark (Table 1). The number of upvotes is much higher than of downvotes for both studies, similar as in Stack Overflow [13]. Users in the main study tended to utilize tags more, as they are generally younger than the ones in the pilot study.

Table 5 shows the survey questions and the responses from users collected during our studies. The average ratings for all questions increased from the pilot study to the main study. In fact, all the negative average ratings had turned to positive. This shows that the overall user satisfaction with our system in the main study had improved from the pilot study. From users' feedback from the pilot study, we found that being able to ask a question to the local community was most important, and pushing the notifications about new answers to their questions was also important. Moreover, we found that too many notifications, irrelevantly recommended questions, and the lack of options to sort recommended questions could annoy users. Based on the feedback, we improved both the mobile application and the server system as described in Table 2 (e.g., sending silent notifications, allowing users to set the maximum number of recommendation notifications per day, managing notification inbox, providing ranking options in the main page, sending answer updates to only askers, and using better recommendation algorithms).

From users' feedback from the main study, we found that the most significant issue was having not enough participants. Some users mentioned improving the quality of recommended questions, improving robustness of tag recommendation, as well as supporting discussion forums for answers.

Question types and answer quality

Types of questions

To understand what types of questions had been asked, we manually categorized all questions, shown in Table 6, based on the question types discussed in [18, 25]. Many questions were with local intent, and meanwhile more subjective questions (recommendation, conversational, opinion) were found than objective ones (factual). The main topics of the questions were about food, study, and relaxing activities. Some example questions and answers from the study participants are shown in Table 7. Such characteristics of the questions were related to the deployment of our system within a campus setting. Compared to Aardvark, the percentages of subjective questions are similar; however, the percentage of questions with local intent is much higher in our study (Table 1).

From the analysis of the question types, we found that many questions were with local intent; however, only 5 and 6 questions were annotated with specific locations in the pilot and main studies, respectively. One reason is that such local intents are often at the university or city level, while the locations provided by our system are at a finer granularity of the building or part-of-campus level. Therefore, users prefer to ask questions to people around all locations in the system, which is the default option. We plan to automatically detect the local intent based on the question text and tags, and dynamically suggest locations with various granularity in the future. As shown in Table 5 (Question 10), users found the function of "asking people around a specific location" somewhat useful in both studies ($0.57 \rightarrow 0.63$). This aspect can be improved if users have more freedom to select locations or areas.

Types	Pilot	Main
Local-intent	63.3%	76.8%
Time-sensitive	12.5%	14.3%
Recommendation	26.7%	50.0%
Conversational	48.3%	14.3%
Opinion	10.0%	7.1%
Factual	15.0%	28.6%
Food	10.8%	26.8%
Study	14.2%	14.3%
Entertainment	20.8%	19.6%
Annotated with locations	4.2%	10.7%

Table 6: Types of questions asked and their proportions.

Q1: my mom is coming into town and I need to entertain her for the weekend. where do people bring their parents for fun things to do? #hungry #music #parents
A1: I would definitely take her to the botanic garden, the aquarium, and walking around emory. Make it a walking weekend with a focus on being active!

Q2: are there any upcoming food festivals in or around ATL? #food #atlanta #festival
A2: Taste of Atlanta! it's next month. great event.

Q3: Where can I get the most food at Emory (for a meal) for the best value? #food #emory #hungry
A3: the DUC is unlimited (all you can eat) for a meal swipe or dooley dollars. If you're tired of the DUC, try the Woodrec in Woodruff hall. you get up to 4 items for a meal swipe or dooley dollars. it's pretty filling, but not all you can eat.

Q4: is there free parking anywhere on campus? #broke
A4: the parking lots on campus open at different times. I know peavine opens at 4, but then fishbourne doesn't open till later 6. and don't risk parking overnight, I've seen people in the morning get tickets

Q5: What class do you recommend for any major to take? #emory #class #professors
A5: Astronomy with the physics department. No prior knowledge needed and lots of fun and pretty easy. Chaucer with Professor Morey was also excellent.

Table 7: Example questions and answers from our studies.

	Survey questions	Pilot	Main
1.	How satisfied are you with the answers?	0.71	1.13*
2.	How would you rate the timeliness of receiving your answers?	0.57	0.94
3.	How satisfied are you with the questions recommended via notification?	-0.86	0.75*
4.	How satisfied are you with the questions recommended via main page?	-0.36	0.50*
5.	Which do you prefer, question recommendation via notification or main page?	$21\%, 71\%, 8\%$	$25\%, 56\%, 19\%$
6.	Which do you prefer for ranking recommended questions in main page?	–	relevant, fresh, popular
7.	How satisfied are you with the tags recommended when asking a question?	–	0.75
8.	How useful are the notifications about answer updates?	0.57	1.19
9.	How useful are the notifications about question recommendations?	-0.43	0.81*
10.	How useful is the “ask people around some place” feature?	0.57	0.63
11.	What did you like (like best) about the system?	...	...
12.	What did you dislike (dislike most) about the system?	...	...
13.	Comments/Suggestions.	...	...

Table 5: Survey responses. Ratings are scaled in {2, 1, 0, -1, -2}. Tuples in Question 5 represent percentages of notification, main page, and no preference. * indicates statistically significant difference according to the Mann-Whitney test at $p = 0.05$.

Response latency

For the analysis of the response latency, we measured the duration between questions and their first answers being posted. The median of this duration is 3.2 hours and 36 minutes in the pilot and the main studies, respectively. One reason for the faster latency in the main study can be found from the more active use of recommendation notifications (Table 9; 11% → 24%). Mobile notifications are usually checked within minutes [29], which largely shortens the time for a user to see and answer a newly posted question. The proportion of questions being answered within the first 10 minutes is 34% in our main study, which is lower than 57.2% as reported by Aardvark [18]. This number could be increased if we had a larger user base. The improvement on timeliness of receiving answers in the main study over the pilot study is also supported by the survey responses shown in Table 5 (Question 2; 0.57 → 0.94).

Quality of answers

From the survey responses in Table 5 (Question 1), we see that users are mostly satisfied with the answer quality in both studies, and the satisfaction had increased in the main study (0.71 → 1.13). Since the answer length is an important feature for predicting answer quality [12], we measured the average answer length in each study and found that the average answer length in the main study was longer (Table 8; 10.8 → 12.4). Compared to Aardvark (median answer length: 22.2 as reported in [18]), the answer length tends to be shorter in our main study (median answer length: 7). One reason is that the input device is limited to mobile devices in our study while users in Aardvark could also use desktop computers.

Measurements	Pilot	Main
Avg # of words in answers	10.8	12.4
Avg # of words in answers (score>0)	13.5	16.6
Avg time editing an answer	52s	40s
Avg time editing an answer (score>0)	66s	46s
% of answers with score>0	39%	31%

Table 8: Statistics related to the answer quality.

Another good indicator for the answer quality is the answer score, which is measured by subtracting the number of downvotes from the number of upvotes. We found that answers with positive scores tended to be longer in both studies (13.5 vs. 10.8, 16.6 vs. 12.4), and users spent more time for editing them (66s vs. 52s, 46s vs. 40s). An interesting observation is that users in the main study tend to type faster, probably because they are generally younger than users in the pilot study.

Question recommendation strategies: PULL vs. PUSH

Which leads to more answers? Which is preferred by users?

Table 9 shows which way the qualified users used the most to find questions they wanted. We consider only behavior of qualified users here because behavior of unqualified users is too sparse and unreliable to be analyzed. For both studies, users answer more questions from the main page than from notifications (39% vs. 11%, 61% vs. 24%). Besides, we see the percentage of answers from recommendation notifications increased in the main study (11% → 24%), implying the improvement of notification recommendation. On the other hand, the percentage of answers from subscription notification decreased (6% → 3%). In addition, the percentage of answers from all questions list also decreased (44% → 12%), as in the main study we removed the “all questions” choice from the navigation menu, but added it to the ranking options in the main page.

Source of answers	Pilot	Main
all questions	90 (44%)	25 (12%)
main page	80 (39%)	133 (61%)
recommendation notification	23 (11%)	51 (24%)
subscription notification	13 (6%)	7 (3%)
total	206 (100%)	216 (100%)

Table 9: Sources of answers (considering only qualified users).

The preference result is also supported by the survey responses (Table 5, Question 5), i.e., the majority of users prefer main page rather than notification for question recommendation (71% vs. 21%, 56% vs. 25%). Meanwhile, more users express no preference between the two recommendation strategies in the main study (8% → 19%), with the improvement made on these strategies.

We are aware that the interpretation of these results highly depends on how PULL and PUSH are performed. Yet, the implication here is that it is important to allow users to pull questions, even in a real-time QA system. This is also mentioned in Aardvark [18] that 16.9% of all users have proactively tried answering questions. However, in Aardvark more users answered via notification than via pulling questions. The explanation was that users were willing to answer questions to help their friends or connected people, but not everyone does so proactively. In our system, the users were not as connected. This might be one important reason for the different preference of PULL and PUSH in our system compared to Aardvark.

Recommendation from the main page: question ranking

Which ranking is viewed more and leads to more answers? How satisfied are users with main page recommendations? Which ranking is preferred by users?

Table 10 shows the results from different ranking algorithms. Users mostly viewed questions ranked by relevance, partially because it is the default option. When considering the likelihood of answering, ranking by relevance and freshness are among the best. This is consistent with the observation in [34] that freshness is an important factor besides relevance for question recommendation.

Ranking	Views	Answers	Answers per view
Relevance	336	87	0.259
Freshness	163	38	0.233
Popularity	21	4	0.190
Location	14	2	0.143
Answer count	8	2	0.250
all questions	114	25	0.219

Table 10: Question ranking in the main study.

From the survey responses (Table 5; Question 4), users are somewhat satisfied with the main page recommendations in the main study (0.50), better than in the pilot study (-0.36). One important reason is that in the pilot study, each user was randomly assigned a single question ranking algorithm, whereas users chose the ranking algorithm by themselves in the main study, which was more preferred by the users. From survey responses in the main study (Table 5; Question 6), ranking questions by relevance (31%), freshness (31%), and popularity (31%) are among the best. An interesting observation is that although users claimed to be interested in popular questions, they were less likely to answer these questions.

Recommendation via notification: user ranking

How many recommendation notifications are sent? How many are clicked and answered? How satisfied are users with this?

Table 11 shows the statistics about recommendation notifications. The average number of recommendations sent per question to qualified users in the main study is higher than in the pilot study (3.0 → 4.3). This implies that a larger proportion of recommendations was sent to unqualified users during the pilot study. Looking at the recommendation notifications sent to qualified users, the likelihood of users clicking on them and the likelihood of users answering the recommended questions had increased (0.40 → 0.52, 0.15 → 0.28).[1] This shows that our system performed better in notification recommendations during the main study, thanks to several improvements (i.e., managing notification inbox, setting max recommendation notifications per day, and the recommendation algorithm). First, as notification inbox was not supported in the pilot study, users could only see the latest one, and miss some previous ones since their last check of mobile notifications. This would affect clicks and answers. Second, we noticed that active users received more and more notifications per day in the pilot study. One reason is that the user ranking algorithm had a bias towards active users. Another reason is that we did not limit max recommendations per day in the pilot study in the first 3 days. To avoid the unbalance of user workloads getting worse, we set limitation to 5 in the last 4 days. Users were not aware of this number. However, in the main study users were allowed to reset this number at any time of the study.

	Pilot	Main
recommendations	356	239
avg rec. per question	3.0	4.3
clicks	142	124
click rate	0.40	0.52
answers	54	66
answer rate	0.15	0.28

Table 11: Statistics of question recommendation notifications.

The improvement of the system regarding notification recommendations is also supported by survey responses. As shown in Table 5 (Question 3), users' satisfaction with notification recommendations increased in the main study compared to the pilot study (-0.86 → 0.75).

Which algorithm is better for user ranking?

To evaluate the three algorithms used in our user ranking component, we use both click related metrics (considering users who clicked on a given question as the ground truth) and answer related metrics (considering users who answered a given question as the ground truth). Specifically, average precision, recall, and F1 scores are computed across all questions using both click and answer based ground truths.

Table 12 shows the results of comparing the three algorithms. When looking at questions that are not annotated with a specific location, the algorithm based on matching users performed better in both click and answer related metrics. First, this algorithm has a bias towards active users, because the document of an active user contains more answered questions and corresponding tags, which makes it more likely to match a new question. Meanwhile, active users are more likely to respond to recommended questions, e.g., by clicking on the question and answering it. The workload balance is handled using the maximum number of recommendations per day set by each user, therefore active users will not get over-annoyed. This is however different from the observation in [17] that matching questions is more effective than matching users for finding potential answerers for a question. First, questions in their setting, i.e., from Yahoo! Answers, is much more diverse in terms of topics than in our setting, i.e., posted by university students, where the main topics are food, study, and entertainment. Therefore, active users are able to answer a larger

[1] Note that all the questions recommended to a user that got answered by him/her are counted here. A user might have not clicked on the recommendation notification but answered it from the main page.

percentage of questions. Second, our results come from live user studies while their results are based on offline simulation experiments. Therefore, it is hard to predict how the results would change from offline simulation to online application.

When looking at questions that are annotated with a specific location, the algorithm based on location proximity showed better performance. This indicates that location proximity is an important factor other than relevance for ranking questions related to specific locations. Yet, more data is needed to do more meaningful analysis regarding the algorithms.

Tag ranking

Which algorithm is more effective for tag recommendation?

To evaluate the algorithms used in our tag ranking component, we use the actual tags entered by the asker as the ground truth, and compute the average precision, recall and F1 scores across all the questions for top 10 and top 3 results of each algorithm.

Table 13 shows the results of comparing the algorithms. For both top 3 and top 10 results, the live system combining the three algorithms achieved the best results on all metrics. The algorithm based on matching questions performed better than matching tags, as matching questions and then summing scores of questions for a tag would better filter out noisy tags in top results while matching tags directly are more likely to return noisy tags because their documents might well match the given question text. Our algorithm based on matching questions is similar to the SimilarityRank approach [37], with the distinction that we sum the scores of similar questions for a tag as its score while their approach chose the max score of similar questions for a tag as its score. Our precision and recall are comparable to the ones in [37].

As shown in Table 5 (Question 7), users are somewhat satisfied with tag recommendations (0.75). This may be improved by applying more state-of-the-art tag recommendation algorithms [23], which will be our future work.

Notification Settings

How many users changed their default notification settings?

How annoying are the notifications?

In the pilot study, three users turned off recommendation notifications. In the main study, one user turned it off, and three users reset the number of max recommendation notifications per day (from 5 to 10, 3, 3 respectively). One user in the pilot study and no user in the main study turned off subscription notifications. This improvement is mainly because we were more careful about sending notifications in the main study, e.g., notifications being sent in silent mode, answerers not getting answer updates on their answered questions, at most 5 recommendation notifications per day and allowing users to reset it, and the improvement of the recommendation algorithms.

From survey responses, we also observe improved user experience about notifications. As shown in Table 5 (Question 9), in the pilot study users rated recommendation notifications somewhat annoying (-0.43), but in the main study users rated them useful (0.81). Meanwhile, in the main study users rated notifications about answer updates more useful (Question 8; 0.57 → 1.19).

DISCUSSION AND IMPLICATIONS

By analyzing the types of questions posted in our studies, we found a large proportion of questions asked are with local intent, subjective, and about food, study, and relaxing activities. Previous research on question recommendation focused more on matching questions and potential answerers based on user interest, expertise and availability [16], while less effort was made on question recommendation considering question types, e.g., questions with local intent. Our preliminary study results indicate that location proximity is an important factor for finding potential answerers to answer such questions besides relevance. More investigation on how to integrate location proximity with relevance and other factors for question recommendation in such cases is needed.

From our studies, we learned that the majority of users prefer pulling questions to answer rather than being pushed questions to answer. However, there are some users who really like to receive notifications, for example, one user commented in survey that "I like the notifications. it's a good way to get people to look at questions without having to browse the app." Therefore, question recommendation systems would achieve better performance if such personal preferences are detected and supported. One solution is to ask users during registration about their preferred settings of recommendation strategies, and then learn based on user behavior to suggest change of the settings or even perform automatic change.

When users do a pull of questions to answer, we found that overall ranking questions by relevance and freshness leads to higher answer rate, which is consistent with the observation in [34]. However, users express different preferences of relevance, freshness, and popularity for ranking the questions for this purpose. Therefore, question recommendation systems need to consider the different importance of factors for ranking questions for different users in the pulling mode. We also found that the question dismissal function was not actively used (75% of the qualified users never dismissed any question from their pulled questions). A potential improvement is to consider users' feedback of dismissing a question in question ranking algorithms, and meanwhile inform users of the benefit.

For deciding which users to push a new question, we found that ranking users by matching user profiles leads to better performance than by matching questions, which differs from the observations in [17], due to the differences in user base (university students vs. Yahoo! Answers users) and in experiment type (live user studies vs. offline simulation experiment). We found that this better performing algorithm has a bias towards active users, who are more likely to answer questions. Therefore, question recommendation systems could rely on active users to contribute more answers, and will benefit from predicting active users before they actually behave actively. Yet, control is needed to avoid over-annoy active users.

Regarding interface design for asking a question, we learned that careful design of location annotation for questions is needed. We found a large gap between the number of questions with local intent and the number of questions that are annotated with a specific location by the asker, partially because the granularity of the predefined locations in our system

		Click based ground truth			Answer based ground truth		
	Algorithm	Prec@5	Rec@5	F1@5	Prec@5	Rec@5	F1@5
location not annotated	matching questions	0.49	0.30	0.35	0.20	0.36	0.24
	matching users	**0.60**	**0.37**	**0.44***	**0.29**	**0.38**	**0.32**
location annotated	matching questions	0.53	0.27	0.35	0.15	0.44	0.21
	location proximity	**0.55**	**0.32**	**0.39**	**0.2**	**0.58**	**0.29**

Table 12: Comparing algorithms for ranking users to send recommendations for a newly posted question. * indicates statistically significant difference according to the Mann-Whitney test at $p = 0.05$.

Algorithm	Prec@3	Rec@3	F1@3	Prec@10	Rec@10	F1@10
matching questions	0.283	0.358	0.300	0.102	0.419	0.158
matching tags	0.220*	0.301*	0.240*	0.092*	0.401*	0.145*
tag popularity	0.289	0.376	0.309	0.111	0.453	0.173
combined live system	**0.294***	**0.400***	**0.319***	**0.116***	**0.485***	**0.180***

Table 13: Comparing algorithms for tag recommendation. * indicates statistically significant difference according to the Wilcoxon signed-rank test at $p = 0.05$.

is not so useful. Meanwhile, Letting askers make manual annotation of locations is annoying, which takes extra effort of askers, especially to select diverse granularity of locations using a complex interface. Therefore, location-aware question answering systems might have user experience improved if they can automatically detect the local intent based the question text and tags, and dynamically suggest locations with various granularity to be annotated with the question.

Our system was designed to be highly scalable. The Django REST framework [1] and Open Source QA models [7] used have proven their success to build scalable server systems. The recommendation algorithms are based on query evaluation using Elasticsearch [2], a scalable distributed real-time search software. The system could be used for multiple communities as well after extending the current limited location choices to cover more places using a map.

The presented studies have a few limitations. First, the data collected is from 27 users in 7 days and 35 users in 3 weeks in the pilot and the main studies, respectively. This limited user base and study period makes it hard to draw definitive conclusions from the data. Further, since users were mainly university students, observations in our studies may not generalize to other populations. Second, we are aware that the two studies have similar but different settings, e.g., the pilot study was conducted in summer with more graduate students joined, while the main study was conducted in fall with more undergraduate students joined. Third, we used rewards as participation incentives. However, given that participants were rewarded for total activity, they could freely choose how to allocate their effort and time. Also, we found that qualified users performed more activities than the required minimum, not just being perfunctory for rewards. Therefore, comparing recommendation strategies and ranking algorithms in this setting is still meaningful. Thus, the findings and results would still be likely to apply in more realistic settings.

CONCLUSION

This paper presented RealQA, a real-time community based question answering system with a mobile interface. Our system provided two strategies for users to get recommended questions to answer: users doing a pull of questions in the main page or being pushed questions via mobile notifications. Two live user studies were conducted to test the effectiveness of the system. Based on user feedback from the first study, we improved both the front-end interface and back-end algorithms of the system. Both users' self-reported satisfaction and behavior related metrics were improved in the main study compared to the pilot one. Different algorithms for question ranking, user ranking, and tag ranking were adapted and compared. Our system is useful for conducting further research on location-aware real-time question answering, and is publicly available to other researchers.

To sum up, our main findings were: 1) a large portion of questions asked were with local intent, subjective, and about food, study, and relaxing activities; both answer quality and timeliness of receiving an answer were reported good by users; 2) the majority of users prefer pulling questions to answer rather than being pushed questions to answer in our setting, though some users like to receive notifications more; 3) for the pulling strategy, ranking questions by relevance and freshness leads to higher answer rate, which is consistent with the observation in [34]; 4) for the pushing strategy, ranking users by matching users leads to better performance than by matching questions, which is different from the observation in [17], due to differences in user base and experiment type; 5) for tag recommendation, the algorithm combining both relevance and popularity performed best.

In future, we plan to improve our system towards asking and answering questions with local intent, and to investigate more personalized question recommendation strategies. We are also building an iOS version of the mobile front end and a web front end for our system.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grant IIS-1018321 (EA, QL) and the Yahoo! Faculty Research and Engagement Program grants (QL, EA).

REFERENCES

1. Django rest framework. `http://www.django-rest-framework.org/`. Accessed: 2014-10-05.

2. Elasticsearch. `http://www.elasticsearch.org/`. Accessed: 2014-10-05.

3. Haystack. `http://haystacksearch.org/`. Accessed: 2014-10-05.

4. Localmind. `http://www.localmind.com/`. Accessed: 2014-08-28.

5. LOCQL. `http://www.locql.com/`. Accessed: 2014-08-28.

6. Lucene's practical scoring function. `https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html`. Accessed: 2014-10-05.

7. Osqa. `http://www.osqa.net/`. Accessed: 2014-08-28.

8. Quora. `http://www.quora.com/`. Accessed: 2014-08-28.

9. Quora online now. `http://blog.quora.com/Getting-Answers-Faster`. Accessed: 2014-08-28.

10. Stackoverflow. `http://stackoverflow.com/`. Accessed: 2014-08-28.

11. Yahoo! answers. `https://answers.yahoo.com/`. Accessed: 2014-08-28.

12. Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, ACM (New York, NY, USA, 2008), 183–194.

13. Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, ACM (New York, NY, USA, 2012), 850–858.

14. Chang, S., and Pal, A. Routing questions for collaborative answering in community question answering. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, ACM (New York, NY, USA, 2013), 494–501.

15. Dror, G., Koren, Y., Maarek, Y., and Szpektor, I. I want to answer; who has a question?: Yahoo! answers recommender system. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, ACM (New York, NY, USA, 2011), 1109–1117.

16. Furlan, B., Nikolic, B., and Milutinovic, V. A survey and evaluation of state-of-the-art intelligent question routing systems. *International Journal of Intelligent Systems 28*, 7 (2013), 686–708.

17. Guo, J., Xu, S., Bao, S., and Yu, Y. Tapping on the potential of q&a community by recommending answer providers. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, ACM (New York, NY, USA, 2008), 921–930.

18. Horowitz, D., and Kamvar, S. D. The anatomy of a large-scale social search engine. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, ACM (New York, NY, USA, 2010), 431–440.

19. Hsieh, G., and Counts, S. Mimir: A market-based real-time question and answer service. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, ACM (New York, NY, USA, 2009), 769–778.

20. Lee, U., Kang, H., Yi, E., Yi, M., and Kantola, J. Understanding mobile q&a usage: An exploratory study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, NY, USA, 2012), 3215–3224.

21. Li, B., and King, I. Routing questions to appropriate answerers in community question answering services. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, ACM (New York, NY, USA, 2010), 1585–1588.

22. Liu, M., Liu, Y., and Yang, Q. Predicting best answerers for new questions in community question answering. In *Proceedings of the 11th International Conference on Web-age Information Management*, WAIM'10, Springer-Verlag (Berlin, Heidelberg, 2010), 127–138.

23. Ma, Z., Sun, A., Yuan, Q., and Cong, G. Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, ACM (New York, NY, USA, 2014), 999–1008.

24. Mahmud, J., Zhou, M. X., Megiddo, N., Nichols, J., and Drews, C. Recommending targeted strangers from whom to solicit information on social media. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, ACM (New York, NY, USA, 2013), 37–48.

25. Morris, M. R., Teevan, J., and Panovich, K. What do people ask their social networks, and why?: A survey study of status message q&a behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (New York, NY, USA, 2010), 1739–1748.

26. Nandi, A., Paparizos, S., Shafer, J. C., and Agrawal, R. With a little help from my friends. In *ICDE* (2013), 1288–1291.

27. Park, S., Kim, Y., Lee, U., and Ackerman, M. Understanding localness of knowledge sharing: A study of naver kin 'here'. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, ACM (New York, NY, USA, 2014), 13–22.

28. Pelleg, D., Savenkov, D., and Agichtein, E. Touch screens for touchy issues: Analysis of accessing sensitive information from mobile devices. In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.* (2013).

29. Pielot, M., Church, K., and de Oliveira, R. An in-situ study of mobile phone notifications. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, ACM (New York, NY, USA, 2014), 233–242.

30. Richardson, M., and White, R. W. Supporting synchronous social q&a throughout the question lifecycle. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, ACM (New York, NY, USA, 2011), 755–764.

31. Song, Y., Zhang, L., and Giles, C. L. Automatic tag recommendation algorithms for social recommender systems. *ACM Trans. Web 5*, 1 (Feb. 2011), 4:1–4:31.

32. Sood, S., Owsley, S., Hammond, K. J., and Birnbaum, L. Tagassist: Automatic tag suggestion for blog posts. In *Proceedings of the First International Conference on Weblogs and Social Media, ICWSM 2007, Boulder, Colorado, USA, March 26-28, 2007* (2007).

33. Stanley, C., and Byrne, M. D. Predicting tags for stackoverflow posts. In *Proceedings of the the 12th International Conference on Cognitive Modelling ICCM 2013* (2013).

34. Szpektor, I., Maarek, Y., and Pelleg, D. When relevance is not enough: Promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, International World Wide Web Conferences Steering Committee (Republic and Canton of Geneva, Switzerland, 2013), 1249–1260.

35. Teevan, J., Karlson, A., Amini, S., Brush, A. J. B., and Krumm, J. Understanding the importance of location, time, and people in mobile local search behavior. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, ACM (New York, NY, USA, 2011), 77–80.

36. White, R. W., Richardson, M., and Liu, Y. Effects of community size and contact rate in synchronous social q&a. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, ACM (New York, NY, USA, 2011), 2837–2846.

37. Zangerle, E., Gassler, W., and Specht, G. Recommending #-tags in twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, vol. 730 (2011), 67–78.