Using Stranger as Sensors: Temporal and Geo-sensitive Question Answering via Social Media

Yefeng Liu
Department of Computer Science,
Waseda University, 3-4-1 Okubo
Shinjuku-ku, Tokyo, Japan
yefeng@dcl.cs.waseda.ac.jp

Todorka Alexandrova
Department of Computer Science,
Waseda University, 3-4-1 Okubo
Shinjuku-ku, Tokyo, Japan
alexandrova@aoni.waseda.jp

Tatsuo Nakajima
Department of Computer Science,
Waseda University, 3-4-1 Okubo
Shinjuku-ku, Tokyo, Japan
tatsuo@dcl.cs.waseda.ac.jp

ABSTRACT

MoboQ is a location-based real-time social question answering service deployed in the field in China. Using MoboQ, people can ask temporal and geo-sensitive questions, such as how long is the line at a popular business right now, and then receive answers that crowdsourced from other users in a timely fashion. To obtain answers for questions, the system analyzes the live stream from public microblogging service Sina Weibo to identify people who are likely to currently be at the place that is associated with a question and sends them the unsolicited question through the microblogging service from which they were identified. MoboQ was deployed in China at the beginning of 2012, until October of the same year, it was used to ask 15,224 questions by 35,214 registered users, and it gathered 29,491 answers; 74.6% of the questions received at least one answer, 28% received a first response within 10 minutes, and 51% of the questions got first answer within 20 minutes. In total, 91% of the questions successfully found at least one answer candidate, and they were sent to 162,954 microblogging service users. We analyze the usage patterns and behaviors of the real-world end-users, discuss the lessons learned, and outline the future directions and possible applications that could be built on top of MoboQ.

Categories and Subject Descriptors

H.5.m **Information interfaces and presentation** (e.g., HCI): Miscellaneous. General; H.4.0 **Information Systems Applications**: General

General Terms

Design; Human Factors

Keywords

Social media; Stranger-sourcing; Crowd-sourcing; Human as sensors; Microblogging; Temporal and geo-sensitive question answering

1. INTRODUCTION

Recently social media platforms including microblogging services are opening their resources to third parties via Application Programming Interfaces (APIs). As a result, other applications, such as Location-Based Services (LBS) systems, have started sharing their users' activities through these social media platforms. Consequently, microblogging services have become an information hub of users' daily experiences, where people publicly share their activities such as where they are, what they are doing, and whom they are with. In China[1] for example, the major LBS applications Jiepang[2] and Dianping[3] both synchronize their users' check-in data with the microblogging service Sina Weibo[4]. This phenomenon creates a unique opportunity for identifying who is very likely to be at a given location at any specific moment by analyzing only their publicly shared information. Furthermore, unlike strong-tie social networks (e.g., Facebook), the weak-tie social networks (e.g., Twitter) have a more open culture that makes it a more desirable environment for supporting unsolicited conversation between strangers.

Since the birth of the Ubiquitous Computing (UbiComp), one of the key challenges is how to extract context information from the environment. Without such information, UbiComp applications cannot provide genuine context-aware services. Recently, the design concept of a human-in-the-loop has been introduced to environment sensing, which has resulted in novel concepts such as Human Probes Sensing, People-centric Sensing, or Participatory Sensing [4, 5]. These concepts refer to the design patterns of utilizing sensors-equipped mobile phones for sensing and monitoring urban areas, which takes advantage of the rapid growth and ubiquity of today's smartphone users. The capacity of these machine sensors is still limited to gathering somewhat low-level physical environmental data, e.g., speed, temperature, and pressure, but for context-aware applications, it is important to also have higher-level information, for example, human activity, group emotions, status of non-electronic objects, and social environments. Inspired by the notion of Human Computation [2], we extended the original idea of Human as Processor to Human as Sensor, where human users become part of sensor networks and report local information around them using portable devices. Using such an approach, a system can collect information that is difficult (if not impossible) to obtain by machine sensors, thus offering the ability to generate a richer contextual model.

We present MoboQ, the location-based real-time question answering service that is built on top of a microblogging platform. In MoboQ, end users can ask location- and time-sensitive

WWW 2013, May 13–17, 2013, Rio de Janiero, Brazil.
ACM 978-1-4503-2035-1/13/05.

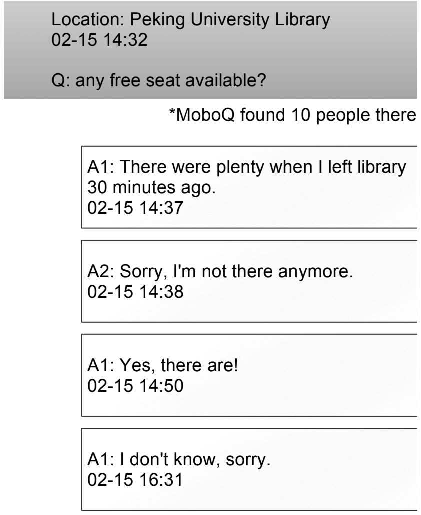
[1] In this paper we focus on China market.
[2] http://www.jiepang.com
[3] http://www.dianping.com
[4] http://www.weibo.com

questions, such as whether a restaurant is crowded, whether a bank has a long waiting line, or if any tickets are left for an upcoming movie at the local cinema; these are questions that are difficult to answer with ordinary Q&A services. MoboQ analyzes the real-time stream of microblogging service Sina Weibo, searches for the Weibo users who are most likely to be at the given location at this moment based on the content of their microblog posts, and pushes the question to those strangers. Note that the answerers in this system are Sina Weibo users, not MoboQ users, and might not even be aware of the existence of MoboQ. This design takes advantage of the popularity and furious growth rate[5] of Weibo, which provides us with the confidence to foresee that microblogging users will be regularly available at any reasonably popular Point-of-Interest (POI) in the near future. The real-time nature of microblogging platforms also makes it possible to expect a faster response time than traditional Q&A systems. To some extent, MoboQ utilizes the Weibo users as local human sensors and allows a question asker to extract context information on any given location by asking the local "human sensors" about what is happening right now around them.

The main challenges of building MoboQ include: how to identify social media users' locations, how to design motivational mechanism to incentivize strangers to answer questions. After addressing these challenges and building the system, there are also interesting questions to ask about users' behaviors in using the service, such as what types of questions are popular and what kinds of usage scenarios are useful. To the best of our knowledge, there is no other similar system that has been deployed in the field before, thus we are the first to evaluate real-world usage of the fundamental concepts of MoboQ.

Figure 1 shows a sample inquiry and its answers from our database[6]. In this example, an asker was checking whether there was a free seat available at a university library. This is a common issue for Chinese students to have difficulties in finding seats in the university libraries before or during exams seasons. MoboQ successfully found ten potential answerers and then opened 10 channels with candidate answerers. Eventually, the system yielded 4 answers. Two of the answers provided possibly helpful and useful information to the asker. At least three of the answerers were very likely at the library at that moment, or just left for a few minutes.

MoboQ was designed and implemented during the autumn of 2011, and was released early in 2012, until the beginning of October 2012, we collected 15,224 questions. With regard to answers, in total we received 29,491 answers from 162,954 Weibo candidate answerers. These observations indicate an 18.0% reply rate for an individual Weibo stranger. Because MoboQ sends one question to up to fifteen candidate answerers, this overall average response rate for a question is 74.6%, where 28% of the answers arrived within 10 minutes, 51% arrived within 20 minutes, and 72% arrived within an hour. This result demonstrates that the approach is feasible under a real-world conditions. We also analyze the asker's usage patterns in MoboQ. We have surprisingly found that very few of the questions actually required a real-time response, although our user interface, sample question and instructions are all designed to emphasis this unique feature of MoboQ. The most popular topics are Restaurant & Bar, and Travel & Location Information. We also found that applications with a more specific usage scenario can be easier to guarantee a stable user experience than a system with a general purpose.


Location: Peking University Library
02-15 14:32

Q: any free seat available?

*MoboQ found 10 people there

A1: There were plenty when I left library 30 minutes ago.
02-15 14:37

A2: Sorry, I'm not there anymore.
02-15 14:38

A1: Yes, there are!
02-15 14:50

A1: I don't know, sorry.
02-15 16:31

Fig. 1. Example question and answers

The main contributions of this paper can be measured from three aspects:

- Designed and implemented a stranger-sourcing system MoboQ and solved a series of system challenges that were related to identifying potential answers, addressing privacy and providing sufficient motivation.
- Deployed MoboQ system in the field, and demonstrated that it is effective across nine months of usage data.
- Analyzed real-user's usage data of the system, discussed the user behavior and lessons learned, and suggested design guidelines for other system designers.

The rest of the paper is organized as follows. In Section 2, some important related works are described. In Section 3, the system design and implementation is described in details. In Section 4 we analyze the usage data and present a number of interesting discussions in Section 5 and Section 6. And finally, the conclusion is given in Section 7.

2. RELATED RESEARCH

There have been some earlier efforts that were related to question answering via social media. In this section we present some of the previous research that is relevant to the system that is proposed in this paper.

2.1 Location-based Q&A

The most similar to our work is a study conducted by Nichols et al. [22] exploring the feasibility of a system that obtains answers by sending unsolicited questions to people based on their public status updates on social networks. The paper discusses the feasibility of answering questions by asking strangers, and the questions asked in the system were auto-generated and were limited to only two specific domains: airport security wait time tracking and digital camera product reviews. There are two major differences between their work and the present study MoboQ. The first difference is that MoboQ provides a system that is deployed in the real world and that allows arbitrary users to ask questions and to receive answers, rather than limited to certain designed

[5] According to companies' official report, Sina Weibo gathered more than 300 million users in 30 months. Tencent Weibo reached 372 million users within 24 months.

[6] Translated into English from Chinese

scenarios. The second significant difference is that their candidate answerers were artificially selected, whereas MoboQ applies a ranking model to automatically handle the filtering and ranking process.

An alternative approach that is used by some location-based services is to rely on experts (such as the rating and review services Zagat[7]) or a large number of users (such as Yelp[8] reviews). However, the former solution usually incurs a high monetary cost and lags in timeliness. The latter approach is challenged by problems such as how to distinguish genuine users from marketers.

Most recently, another approach for building location-based Q&A services is Social Network Mining [14], which refers to providing information that is based on the analysis and mining of past tweets from a given location, and summarizing those tweets to extract information about the location in an implicitly manner. An example of such a service is the Social Telescope proposed by Shankar et al. [29], which uses geo-tweets by Twitter users to learn about places. Moreover, there are a number of research studies [27, 8, 6] on how to detect and summarize events in real-time by analyzing Twitter feeds. Most of such applications are designed for crisis event reporting such as earthquake detection. Basically, our approach is different because MoboQ attempts to identify "a targeted social media user" rather than existing information, and thus an asker can directly obtain the answer that he/she is looking for from another human.

Another related study involves a location based information system for storing user-generated content at a geographical position, such as GeoNotes [26]. GeoNotes is a Location-based Post-it Notes that allows a user to store digital information at physical locations for other users to access while using mobile devices. This service is different from MoboQ because it is a database for the future re-usage of local information, while MoboQ provides a solution for extracting real-time local information in an on-demand fashion.

2.2 Social Search

Typical crowdsourcing and human computation systems treat contributors as interchangeable "cogs in the machine". In contrast, services that can be categorized as social search systems seek to leverage each contributor's unique skills and knowledge. These services help users to identify and connect with relevant experts who can offer solutions in a timely and human manner.

Macdonald and Ounis [19] proposed a data fusion-based approach for predicting and ranking candidate expertise with regard to a given question. The effectiveness of using adapted data fusion techniques was demonstrated by their evaluation results. Social Q&A Services such as the services proposed in [25] and [10] aim to perform social search by asking questions within the asker's social network, where they establish a community with "tagged" users, and the search is performed within that community. Panovich et al. [25] have explored who is likely to contribute valuable information in social network Q&As, and they found that strong ties offer more valued contributions compared with weak ties. Aardvark [10] was one of the most popular community-based social search engines: it was launched in early 2008 and was later acquired by Google in 2010. Aardvark's users ask questions similar to other on-line question and answer sites, but the Aardvark server "routes the question to the person in the requester's extended social network who is most likely to be able to answer the question." Aardvark prioritizes friends-of-friends for responses, which appears to be a good way to increase the quality of the response. A similar idea is Bernstein, M.S. et al.'s "friend-sourcing" approach [3], a form of crowdsourcing that is aimed at collecting information that is available only to a socially connected group of individuals within a strong-relation social network, such as Facebook. Our approach is a "stranger-sourcing" in weak-relationship social network such as Twitter, which makes it different from these approaches. Moreover, the previous approaches do not aim to solve real-time and location-related questions, such as in MoboQ, which is the proposed goal of our work.

2.3 Mobile Crowdsourcing

Crowdsourcing is an approach that seeks to use human intelligence to overcome the limitations of computing technologies. Extending this model to mobile devices and users can potentially further enhance the availability of the contributors.

In [17, 18], Liu et al. proposed a social media crowdsourcing platform that leverages the Twitter and mobile email to rapidly allocate microtasks to a wide network of crowdworkers, and that visualizes the geotagged result data to create a type of visual information pool. Yan et al. [32] proposed a crowdsourcing-based approach to improve the quality of real-time image searches on mobile phones. Their system combines an automated mobile image search with the real-time human validation of the search results. Crowdworkers from Amazon Mechanical Turk[9] perform the validation tasks. Moreover, CrowdSearch algorithms were proposed to optimize the delay and the money constraints. Their experiment reported a result of 95% search precision for several categories of images. The Ushahidi platform [24, 16], a Google map-based mash up tool to visualize crowdsourced information by allowing participants to submit information through text messaging using mobile phones, e-mails or the Web, is another successful mobile crowdsourcing initiative. After the Haiti earthquake in January 2010, people and organizations in Haiti posted thousands of their needs and requests on the Ushahidi Haiti map; volunteers with the ability to answer the requests then took actions and handled the requests.

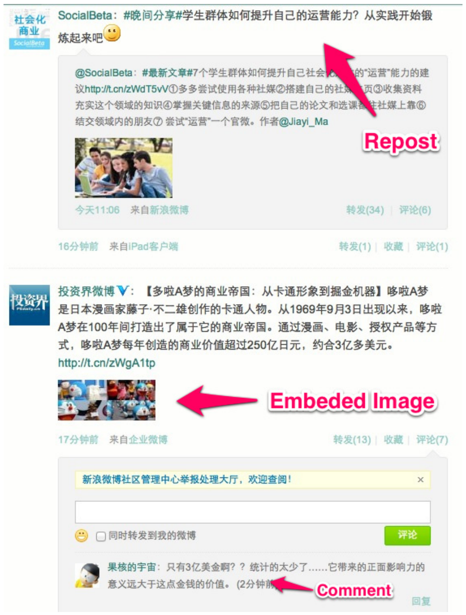
2.4 Context-awareness and Sensing

Context-aware applications are essential components of the ubiquitous computing vision. A. Schmidt et al. [28] proposed a classical model that suggested that context is composed of two aspects: the physical environment (e.g., light, pressure, acceleration, temperature, etc.) and human factors (e.g., users, social environment, activities, etc.). Machine sensors have become the main stream for extracting context information from physical environment, although they are not as ubiquitously deployed in the environment as needed for many applications. Sensing the human factors of context is much more difficult however. Burke et al. introduced the concept of Participatory Sensing [4], which uses everyday mobile phones that are carried by users to form sensor networks that gather local information, because, currently smart phones are typically embedded with rich sensors and the mobile users would be ubiquitously available in any modern city. For example, Jiang et al. [10] explore the ways

[7] http://www.zagat.com

[8] http://www.yelp.com

[9] http://www.mturk.com



SocialBeta：#晚间分享#学生群体如何提升自己的运营能力？从实践开始锻炼起来吧

@SocialBeta：#最新文章#7个学生群体如何提升自己社会[illegible]的"运营"能力的建议http://t.cn/zWdT5vV①多多尝试使用各种社媒②搭建自己的社[illegible]页③收集资料充实这个领域的知识④掌握关键信息的来源⑤把自己的论文和选课[illegible]往社媒上靠⑥结交领域内的朋友⑦ 尝试"运营"一个官微。作者@Jiayi_Ma

今天11:06 来自新浪微博 转发(34) | 评论(6)

16分钟前 来自iPad客户端 转发(1) | 收藏 | 评论(1)

投资界微博：【多啦A梦的商业帝国：从卡通形象到掘金机器】哆啦A梦是日本漫画家藤子·不二雄创作的卡通人物。从1969年9月3日出现以来，哆啦A梦在100年间打造出了属于它的商业帝国。通过漫画、电影、授权产品等方式，哆啦A梦每年创造的商业价值超过250亿日元，约合3亿多美元。http://t.cn/zWgA1tp

17分钟前 来自企业微博 转发(13) | 收藏 | 评论(7)

新浪微博社区管理中心举报处理大厅，欢迎查阅！ ×

□ 同时转发到我的微博 评论

果核的宇宙：只有3亿美金啊？？统计的太少了......它带来的正面影响力的意义远大于这点金钱的价值。(2分钟前

回复

Fig. 2. Screenshot of the main timeline of Sina Weibo (up: repost; down: post with image and its comment)

in which human-centered sensing can be applied to support a variety of analytic campaigns in the context of crisis response and management. However, the performance of Participatory Sensing by machine sensors is still limited. To some extent, MoboQ can be seen as a combination of two concepts: humans as sensor and participatory sensing.

3. DESIGN AND IMPLEMENTATION

In this section, we present the design of the MoboQ system: the overview of the main components, the basic workflow of the usage, the core algorithm for selecting and ranking of the potential stranger answerers from Weibo, the motivation mechanism for incentivizing answerers, and the user interface design.

3.1 Background: Microblogging Services and LBS in China

Sina Weibo is a Chinese microblogging service. Sina Corporation[10] launched this service in August 2009. The basic functionalities of Sina Weibo are similar to those of Twitter[11]: messaging, re-posting, direct messageing, mention (@), hashtag (#), and more. In addition, Weibo users can comment on a post, which means that the comments will be inserted into the original post rather than into their own timelines. This feature allows different Weibo users discussion with each other under one topic (post). A screenshot of Weibo UI is given in Figure 3. Moreover, Weibo officially supports posting images, videos, music via local files or on-line URLs, and locations. Sina Weibo allows up to 140 Chinese characters per post, which means that one Sina Weibo message could contain more information than in Twitter. Furthermore, unlike Twitter, Chinese microblogging sites normally support a large amount of "verified accounts" under their real name systems (i.e., users must link their national identification number to a microblogging account), for people to declare their professional identities and titles. Figure 2 gives a screenshot of the main UI of Weibo. By the middle of 2012, Sina Weibo reached 368 million registered users, which makes it the second largest microblogging service in China.

There are a number of Foursquare[12]-like location-based sharing services in China, including Jiepang, Dianping, and Sina Place[13]. They each work in a similar way: people check-in when they arrive at a location and earn badges or other rewards (sometimes coupons from local shops). Users can also give comments on the location or upload multimedia content of the location. According to an industry report from EnfoDesk[14], the total number of location-based services' users had reached 30 million by the end of 2011 and was growing quickly.

MoboQ can take the advantages of both Sina Weibo and various location check-in services, because almost all of the check-in services provide their users with the functionality of broadcasting their check-in activities on social media platforms such as Sina Weibo.

3.2 Main Components

The main components of MoboQ are:

- *Communication Module*: A REST Web Service, with an open API to client applications and the Sina Weibo API. This module handles the communications between an asker from MoboQ and an answerer from Sina Weibo
- *Ranking Engine*: Searches and selects the best candidates on the Weibo platform to answer a question.
- *Client Applications*: Includes a Web site, mobile web, and a native mobile application to present the questions and answers to the user in an accessible and interactive form.

The MoboQ server, which comprises the communication module and the ranking engine, is implemented in Ruby on Rails. The mobile web is implemented using HTML5 technologies.

3.3 The Life of a Query

Before discussing MoboQ in detail, it is useful to describe what happens behind the scenes when a user asks a question (Figure 3).

- A user begins by asking a question, which ideally is both location and time-sensitive. To ask a question, the user must search and select a specific location or POI (Point-of-Interest), the location data are available via Sina location API. Then the user should inputs the question about the POI. This input can be accomplished through different client applications.
- Once the new question is delivered to the server, the *Communication Module* calls the Sina Weibo Search API to obtain a raw result data set of real-time conversations about the location (i.e., Weibo posts having this location keyword in the content). These raw data are then sent to the *Ranking Engine*.

  Raw results are re-ordered based on a series of principles in the *Ranking Engine*. For example, a high voting weight is given to a post that has the following characteristics: a check-in message, posted recently, from a Weibo user whose profile shows he/she lives in the same area of the question, etc.

[10] http://www.sina.com
[11] http://www.twitter.com

[12] http://www.foursquare.com
[13] http://place.weibo.com
[14] http://www.enfodesk.com

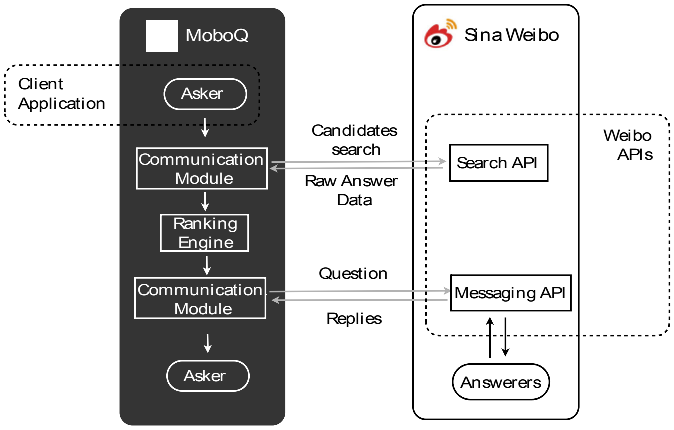


Fig. 3. The life of a query in MoboQ

- Based on the ranking result, the *Communication Module* selects at most fifteen Weibo users to answer the question. The original question is then pushed to each user via the Sina Weibo Messaging API as a normal Weibo post. The post body describes that this request is a real request from a real person who needs help, rather than a spam message. Links to the asker's MoboQ profile and the question page are also given in the content. The features are designed for helping Weibo users to understand why they have received such a message from a stranger. The Weibo users are notified by the mention function (i.e., @), and they can repost or comment on the question message to respond to the request. Either way, the answer will be caught by the Communication Module and will be delivered back to the MoboQ asker.
- Optionally, after receiving an answer the asker can "thank" or start further conversation with the answerer. A one-to-one communication channel is established via Sina Weibo to support this possibility.
- After 24 hours, a question is marked as "closed" and no further responses are collected. Such a "deadline" is added because of the real-time character of the MoboQ service and the limited access rights of Weibo API.

3.4 Ranking Algorithm

Ranking in MoboQ is performed by the *Ranking Engine*, which determines an ordered list of candidate answerers who should be contacted to answer a question, from the raw data returned from the Sina Weibo search API.

The main factors that determine this ranking of users are Current Location, Expertise, and Availability:

Current Location: First, the Ranking Engine finds the subset of users who have recently checked-in at a target location: those users that have posted content that matches the canned format of the check-in information that is shared by the other LBS applications. The posting times of check-in posts are also considered, and old check-in messages are ignored.

Location Expertise: Second, the Ranking Engine matches the users' profile with the given location to verify whether they are local residents. In the next stage, the tag of the user and the classification of the question will be considered, and recent historical posts of the user will be analyzed to determine if this user is a "local expert". The overall goal is to score each user according to the degree to which he/she is a good "match" for this location information query.

Availability: Third, the Ranking Engine prioritizes candidate answerers in such a way so as to optimize the possibility that the present question will be answered. This task involves factors such as prioritizing users who are the most likely currently online (e.g., have posted a new message in the last 5 minutes); who are active Weibo user (e.g., have reasonable amounts of Weibo posts); or who live or have lived in the same area of the asked POI; etc.

Given this ordered list of candidate answerers, the Ranking Engine then filters out Weibo users who should not be considered. These filters operate largely as a set of rules such as do not consider users who have more than 3000 followers. This rule is used because such users most likely have vast numbers of mention ("@") messages every day and thus are unlikely to respond. Another rules is: do not consider people without a profile picture or have a default display name (e.g., "Weibo User 1672832") because there is a chance their accounts are used infrequently or are automated robots.

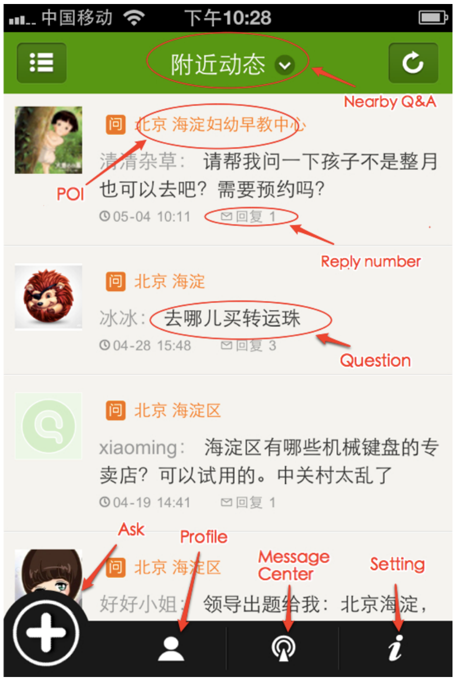
Finally, the top candidate answerers (at most ten in the beginning, which is increased to fifteen later) who have survived this filtering process are sent to the Communication Module. This model then proceeds with opening channels via the Weibo Messaging API to each of the candidates, serially, inquiring whether they would like to help the MoboQ user by answering the present question; a polling service checks regularly for new notifications and returns any answer or conversation to the asker.

3.5 Motivation Mechanisms

Because MoboQ is a Q&A system between people who are likely to be complete strangers, it is a challenge to motivate the potential answerers from Weibo to answer the strangers' questions. From a design point of view, we concentrate on two aspects simultaneously: 1) how to establish trust among the answerer, the MoboQ platform, and the asker, and 2) how to provide appropriate incentives to the candidate answerers.

Early studies suggest that a lack of trust can be identified as one of the greatest barriers that inhibits Internet transactions [8, 13]. For supporting online trust building, the content of a MoboQ query message has been specially designed. This message includes the reason why the candidate answerer has been selected (e.g., "hi, we found that you just visited #Location#"); the URL of the asker's profile page on the MoboQ site to show that the asker is a real person; and another URL to the question's page on MoboQ to help the answerer learn about the service. All of the necessary information is open to the candidate answerers, and we hope that this option helps the answerer to understand that this question is not a spam message but instead is a real question from a real person who is seeking help.

We also utilize findings from social psychology as incentives in our system. Social Facilitation [33] and Social Loafing [15] are two commonly cited behaviors that can affect contributions on social sites. The Social facilitation effect refers to the tendency of people to perform better on simple tasks while someone else is watching, rather than while they are alone or when they are working alongside other people. The Social loafing effect is the phenomenon of people making less effort to achieve a goal when they work in a group than when they work alone, because they feel that their contributions do not count or are not evaluated or valued as much. This consideration is seen as one of the main



中国移动 下午10:28

附近动态

问 北京 海淀妇幼早教中心

清清杂草： 请帮我问一下孩子不是整月也可以去吧？需要预约吗？

05-04 10:11 回复 1

问 北京 海淀

冰冰： 去哪儿买转运珠

04-28 15:48 回复 3

问 北京 海淀区

xiaoming： 海淀区有哪些机械键盘的专卖店？可以试用的。中关村太乱了

04-19 14:41 回复 1

问 北京 海淀区

好好小姐： 领导出题给我：北京海淀，

Fig. 4. Screenshot of MoboQ iPhone App, in the picture it is the page of recent questions

reasons that groups are less productive than the combined performance of members working alone. Different mechanisms is employed in MoboQ to take the advantage of the positive social facilitation and to avoid the negative social loafing:

- A public thank you message is provided thus individuals' efforts can be prominently displayed.
- The query is a public message thus individuals should know that others can easily evaluate their work.
- Every question is sent to up to fifteen people in a separate message, to allow the unique value of each individual's contribution to be emphasized.

Some other popular incentive mechanisms such as monetary incentives or game mechanisms are not implemented in the current version of MoboQ. The reason is that the relationship between the answerers and the MoboQ platform is different than in previous systems: the answerers (who need to be incentivized to perform) are not yet registered users thus it is difficult to offer points, currency, or scores directly to them.

3.6 Client Applications

MoboQ provides various client-side programs that people use to ask questions: Web site, mobile web (adopted by major mobile devices), and an iOS native application. The screenshot of an iOS Application is shown in Figure 4. Moreover, we foresee that the future user interface for asking a question on MoboQ, in its simplest form, can be any type of lightweight text input mechanism, along with a mechanism for displaying textual messages returned from MoboQ, e.g., SMS, Microblogging, IM, E-mail.

In this initial version, the MoboQ production system has no strong design for supporting in-community question answering. Although current MoboQ users can answer questions in the system, their answers are specially marked to notify the original asker that these answers are from a different source other than from Weibo.

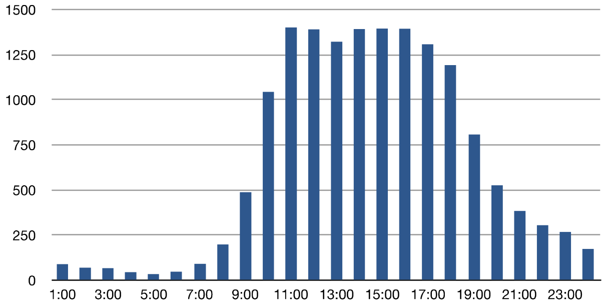


Fig. 5. The temporal patterns associated with the questions.

4. ANALYSIS

The Web version of MoboQ was deployed and made available publicly in January 2012, and the iOS version App was published in the iTunes store in March 2012.

By the beginning of October 2012, there were a total of 35,214 users registered on MoboQ, and a total of 15,224 questions were asked. Those questions were sent to 162,954 strangers on Weibo, which resulted in 29,491 answers and 3,408 follow-up conversations. Most of the questions are about locations in an urban area. We provide an overview of the data in the following.

We found only 4909 registered users (13.94%) posted at least one question. But we assume it is somehow normal for such on-line community to have a power law distribution, with only a small minority contributing. For instance, [1] reported in the P2P music sharing service Gnutella 10% users provided 87% of all the music. In the open source software community, 4% of participants likewise contributed 88% of the new code [20].

4.1 Demographics

Aside from data stored in the server, we also use Google Analytics[15] to monitor the traffic of the MoboQ Web site, and HTML5 App. Not spuriously, 94.06% of the traffic was from mainland China, followed by U.S. (2.19%) and Japan (1.38%). Moreover, most of the accesses were from the major Chinese cities, namely Shanghai (20.09%), Beijing (13.71%), and Guangzhou (6.81%).

4.2 Temporal Patterns and Answer Rates

The MoboQ users were more active during the daytime, from 10 am to 6 pm. Compared to major social media sites such as Facebook (10am to 10pm)[16] or Twitter (9am to 8pm)[17], users appeared to use MoboQ more during typical working hour and the usage faded quickly after 6 pm. Figure 5 provides an overview of the hourly distribution of the questions.

Of the 15,224 questions asked in MoboQ, 1383 questions (9% of the total questions) could not be sent out because the server failed

[15] http://www.google.com/analytics

[16] Anatomy of a Facebook Post
http://www.vitrue.com/educational-resources

[17] Time is on your side.
http://blog.bitly.com/post/22663850994/time-is-on-your-side

Category	%Question Number
Restaurants & Bars	48.6%
Travel & Local Information	32.6%
Transportation	6.6%
Vote & Opinion	2.2%
Social	1.6%
Miscellaneous	7.2%

Table 1. Categories of 500 Questions in September

to find any candidate. The main reasons were an uncommon POI/City and mistyping of the location name. For the rest of the remaining questions, the average question found 8.17 candidate answerers (SD=3.34), and 10319 of the questions eventually received at least one answer, which resulted in a 74.6% overall answering rate.

4.3 Desktop and Mobile Usage

Accesses from mobile devices (i.e., using mobile Web and/or an iPhone App) accounted for 25.3% of the total traffic. This number is lower than our expectation. Because the majority of the usage scenarios that we pictured were for mobile users, we believed that microblogging-style short query and answer is more useful on a phone, and the tasks are more likely to have time constraint when people are in a mobile context.

We believe that the absence of an Android App may be part of the reason, because a recent report[18] shows that Android phones have already achieved more than 90% of the smartphone market share in China. In fact, even without a native App more than 10% of the mobile traffic was from Android devices.

4.4 Questions Classification

We randomly selected 500 questions that were sent to MoboQ between September and October of 2012, and analyzed all of the questions' contents via human coding. Eventually, we classified the questions into six major topics (see Table 1 for an overview). Particularly, Reviews, Advices, comments and recommendations, queries regarding restaurants and bars are very popular (for example, "Is the food good? Can you recommend some nice dishes?"). A large number of questions are looking for insights, recommendations, and advices from local residents/experts (for example, "Can you recommend any good place nearby this university that is suitable for a business meeting?"). Some of the questions are socially oriented (e.g., "Hi, is there anyone there who wants to have a café with me?"), where users were trying to attempt a conversation rather than asking a question.

Surprisingly, we found that MoboQ users actually did not ask many time-sensitive questions. MoboQ was originally designed for the situations in which a user is in the middle of an on-going activity and requires an answer to continue. However, only 27 (5.4% of the sample questions) of the questions required real-time information at the given location and accordingly demanded a rapid response. Most of such queries ask about traffic conditions or parking. We consider that one problem could be that there is a general-purpose system design without a clear and helpful usage scenario, which failed to guide new users to better understand the new concept of MoboQ.

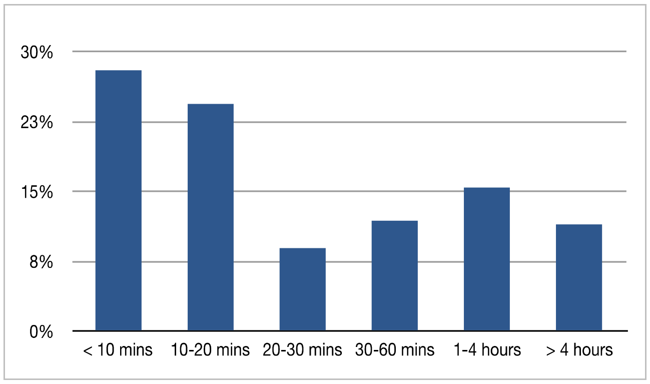


Fig. 6. Response time (the first-answers)

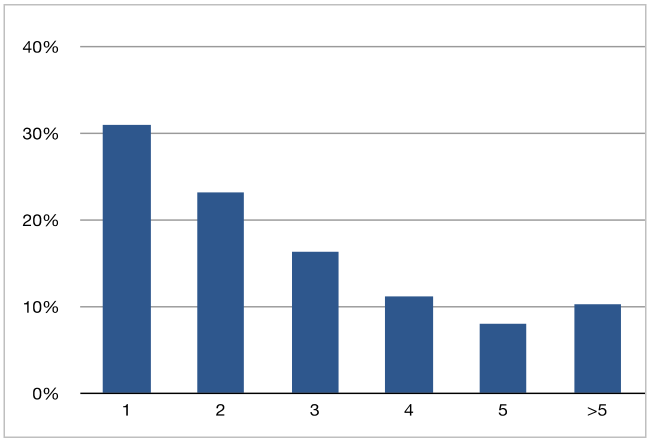


Fig. 7. Number of answers per question

4.5 Answers Quality

For a system that comprises human and machine sensors, the delay between these two sensors is one of the most important problems. Furthermore, how fast human participants can provide an answer largely determines the asker's experiences.

With regard to the response time of MoboQ (see Figure 6), we found that 2868 first answers (28%) arrived within 10 minutes, 5,366 (51%) arrived within 20 minutes, and 7494 of them (72%) arrived within an hour. In contrast, for asking questions using Facebook or in a friend-sourcing manner, only 24.3% of the queries received a response in 30 minutes, and 42.8% received a response in one hour or less [22].

If there is any type of additional process, it will delay the response time in MoboQ; for the current version we do not emphasis quality assurance. This constraint means that all of the answers were delivered to askers immediately for them to make the judgment, i.e., to select the alternatives that most likely are the correct answers. As a result, the quantity of an answers also indirectly determines the quality of the answer.

Of the 10,307 answered questions, an average of 2.86 answers were received per question (SD=27.57). As Figure 7 shows, approximately 69% of the questions receive multiple answers, and 10.2% of the questions receive more than 5 replies.

Because of the push-based nature, MoboQ has fewer threads that result from spammer issues. Most of the meaningless answers are "I don`t know" or "I`m not familiar", rather than malicious answers (i.e., answers that intentionally provide wrong answers for their own good, such as advertising their own business).

[18] http://www.techinasia.com/android-market-share-china-2012/

4.6 Askers as Answerers and Vice Verse

Unlike traditional Question Answering site, by design the MoboQ users generally just play the Asker role. We found of the 29,487 answers, 2051 MoboQ registered users (5.8%) contribute 4731 responses (16%). This observation might also arise because the users typically are able to answer the questions about the locations that they are familiar with. When the number of questions is relatively small, the users might hardly ever see a question that they could answer. As a comparison, 27% of the Yahoo! Answers users have ever provided a single response [12].

Surprisingly, even if 22,882 Weibo strangers have answered at least one question, less than 0.5% of them joined MoboQ afterwards. Every question contains two URL links to MoboQ site. One link is the asker's page and the other is the question's detail. However, we tracked very few incoming accesses from those links. After receiving answers, 6,206 thank you messages were sent by askers (i.e., 21% of the answers received a thank-you message that also contains an implicit invitation to join the community). This result might be indicate that the strangers could be willing to help by sending a simple reply sentence within Weibo, but clicking an additional URL to leave for another site might be too costly or risky for them.

5. DISCUSSION

In this section we focus on the implications of the analysis of the usage data and discuss the findings.

5.1 Privacy

The privacy issue has been widely discussed ever since the birth of social media, and the stranger Q&A feature of MoboQ pushes further on social media users' understanding of on-line privacy. In [30], the authors propose a number of guidelines for creating a location-sharing service. In [31], the authors discuss how a socially driven sharing (e.g., the location-based service Foursquare) is fundamentally different from purpose-driven sharing (e.g., Google Maps).

In MoboQ we define several fundamental rules for protecting a user's private information, for example, the collected data that are used to predict user's location must be publicly shared by the user himself/herself. However, we had the concerns that a Weibo user`s understanding of the service might be different from the actual model that is used by the site, for example, users might not be aware that their check-in message is publicly posted. Therefore we maintain our attention on any user feedback that is useful for analyzing how much of an issue this concern is for the MoboQ system. Overall, based on the feedback data, there were only two answerers who complained that the sudden question was annoying to them and that is violated their privacy. However, we do not have the data on whether users have marked the question messages as spam in Weibo. From the community management point of view, it is also important to guarantee that all of the questions are real and sincere. Spam questions should be strictly filtered out before being sent to social media users; thus, a comprehensive monitoring system should be established in the next stage.

5.2 Additional Quality Control Method

As explained in the previous section, there is no explicit quality control method that is implemented in the current version of MoboQ. However, we observed people who were not at the location or did not understand the MoboQ service rather than malicious users who generated a large number of irrelevant answers. We believe that the push-based question assignment mechanism also helped to reduce the possibility of having malicious answerers because all of the answerers were selected by the algorithms and thus spammers had less opportunity to insert themselves into the process.

To further improve the answer quality, additional components might be included. In the future we are interested in building classifiers that could detect extremely bad answers, and filter out obviously useless messages and spam messages using Natural Language Processing technologies.

5.3 Augmenting MoboQ through Social Media Mining

Systems such as Social Telescope [29] collect location-related information via Social Network Mining. Although many questions that are asked in MoboQ could not be satisfied through such an implicit way of collecting a review, such an approach could significantly enhance the response time for the party of the queries. When introducing a human into the loop, one of the drawbacks is that humans increase the time-cost of the whole process. Hence, data mining in the social media could potentially augment the weakness of MoboQ.

Therefore, we consider a combined Social Network Mining approach, which presents the results to an asker as reference data; this approach could be another interesting future direction. Simultaneously, we would consider adding a new feature of "report" into MoboQ, with which the user can "report" what activities are occurring at his current position and share these interesting stories with the community. By accomplishing these changes, the system could generate a "potential answers pool" before anyone is actually asking a question, which could be used as a reference for the potential askers.

5.4 Trust and Incentives

When discussing the stranger-sourcing concept, many people would first think of what incentive to provide to the answerers; however, we found that perhaps a more important question is how to build trust between strangers via the Internet. In MoboQ, a point system and a simple overall leaderboard are implemented to motivate community members to answer questions, but neither of these approaches appears to be effective because few MoboQ users contributed answers. However, the Weibo users (to whom we can hardly provide any strong incentives) actively responded to the requests. We believe that this result might demonstrate that once the strangers consider that the message is coming from a real person who is in trouble, they would very likely offer their help. However, without further study, we cannot provide a conclusion as to which factor might drive such altruistic activity.

Nevertheless, we designed the system as openly as possible to support online trust between strangers: non-registered strangers can easily verify the asker's profile and history, as well as the MoboQ's introduction information. All of the messages between MoboQ and Weibo are sent in public channels rather than private messaging, to allow all of the information to be open to everyone. Furthermore, the Weibo user's effort to help the asker is minimized to the task of simply replying to the tweet.

5.5 Support Region Oriented Questions

We found that such a location-based system should be able to search for a larger region instead of only a specific POI. Current MoboQ is designed for asking questions to very specific POIs, e.g., a shop, a restaurant, a bar, etc. One of the reasons for this is because current location-based sharing services users normally check-in at POIs rather than at regions.

However, we found that a MoboQ user often sets a whole area as a target location, and then asks a question about one specific place in that area. For example, if Dom Coffee is one shop in a large shopping mall called Nordstrom's, then many users might ask "#City, Nordstrom shopping mall# is Dom Coffee crowded right now?" rather than "#City, Dom Coffee# is it crowded over there". This action is logical and is an understandable way of thinking; however, it adds new requirements when searching for appropriate answerer candidates. This type of query might may suggests that in a future version of MoboQ, we should implement an additional sematic parser to retrieve possible POI names out from the question content, and consider these POIs to be another factor in the ranking algorithm.

5.6 User's Expectations and Demands

In traditional Q&A sites such as Yahoo! Answers, there is already a large database of questions and answers; thus, a user can search for the answers immediately. On the other hand, MoboQ attempts to solve real-time questions, which means that, even for the same question at the same location, the answer would most likely be different depending on the time and the context; thus the previous questions and answers are much less useful than in other Q&A services. As a result, compared to other Q&A sites, MoboQ does not have a particularly well-supported function to display and search historic questions and answers. However, such a design leaves a user almost nothing to do other than asking/answering questions and waiting. From the user's feedback, we received several suggestions about providing more intelligent questions and answers feed that contains only the locations that they are interested in, which implies that there would be a demand for browsing historical data.

We found that another core issue is that users have different expectations of the response time, with regard to the type of question or the context. However, MoboQ cannot guarantee the waiting time to users while adjusting these expectations. In the future, we believe that it could be very helpful to give an estimated waiting time for each question based on historical data. Additionally, a notification should be given when it is highly likely that the user's expectation for the response time would be higher than what the service could offer, for example, in the case of asking about a question to an uncommon location or at the mid-night. An appropriate guide should be provided to suggest possible alternative tasks or choices.

6. Future Applications for Specific Scenarios

MoboQ is designed to be a common platform for supporting location-related real-time question answering. In our initial deployment we focused on a wide range of questions, but the results show that it is difficult for a new user to understand such a general-purpose application thus the user adoption is slow. We believe that it is important to repurpose our general infrastructure to support a more focused applications that are difficult to be implemented by traditional methods. Nonetheless, from the asker's usage data and the classifications, we can could see the potential of narrowing down the usage scenario and building ubiquitous computing applications for more specific purposes on top of the MoboQ platform. Moreover, it is reasonable to assume that such applications' performances could be better than the current MoboQ system if the ranking algorithm and the data sources can be adjusted for the specific application domain.

The most successful applications built on top of the MoboQ would need to have the following properties:

- Utilize information that cannot be collected via traditional sensor technologies;
- Be able to tolerate a reasonable delay (e.g., 5 minutes to half an hour), because human sensors cannot provide a response as fast as machine sensors;
- Location and time-sensitivity should be required components of the solution; and
- The problem targeted is difficult for existing social search systems to solve.

With the guidelines and the question classification results in mind, we can at least foresee the following potential applications directions:

6.1.1 Tourism Information

Visitors might want to obtain get immediate support from local residents, e.g., recommendations for restaurants, advises for transportation, questions about directions, and weather condition. This possibility is different from existing users' reviews and local information search services such as Yelp or Dianping because people can have asynchronous conversation with another people to ask information with certain requirements and to obtain specific answers for their demands. For example "how cold is Beijing, what should I wear, is a jacket enough" instead of having a number for the temperature, or "is there a café suitable for business meeting" instead of checking comments on Yelp to find out the answer.

6.1.2 Emergency Management Information

A stranger-source system could also be used for information collection during emergency situations, such as detecting the current state of a given location after a natural disaster. During natural disasters, such as earthquake, it is common for the telephone voice network to be destroyed or inoperative. Some of the authors witnessed the situation that occurred at the nearest major cities to the epicenters of the 2011 Tohoku Ms9.0 earthquake: the telephone network was completely down after the first shock waves, but the Internet connection survived, although the speed was slow. During the important first 24 hours, many people relied on social networks to report, communicate and receive the latest news. Previous researches [27, 9, 11, 7] studied how to utilize Twitter feeds to real-timely detect crisis events such as earthquakes, however there was no method like MoboQ to push questions to a specific location for extracting needed information. Thus, the information could be greatly helpful for assessing the extent of the disaster and then for effectively allocating the valuable rescue and relief resources to the hardest hit areas.

6.1.3 Shopping Information

Real-time (or so-called semi-real time) information about shops, restaurants, and entertainment facilities can be helpful for supporting customers' decision-making processes. Often, people need such additional information for making plans to visit a location. For example, "which department store has the larger sale today", "is there a ticket left for the movie", or "which restaurant has no line waiting outside." We discovered that this type of questions that were accounted for a large proportion of the questions asked in the current MoboQ system, which suggests that it is an especially useful application of this technology.

6.1.4 Event report

Check-ins sometimes occur at an event rather than at a location. It is therefore possible to find people who are currently at big on-going events such as conferences, sports games, music concerts, and so forth. Although there are alternative studies [23, 6] about mining social media data to summarize on going or past events such as football games, for people with specific questions or purpose, question answering type of service could be more useful. Moreover, for small events while it is less likely that there will be enough information to generate a summary, Q&A approach could be a good alternative.

6.1.5 Context Acquisition Systems

Stranger-sourcing technicalities could also support an existing context sensing system for extracting data that cannot be collected otherwise. For example, a computer system can automatically and autonomously generate questions, translate system needs into human language, and extract data from answer content (i.e., natural language). Thus, such system can constantly fetch high-level context data such as a social environment or human activities, and can store the information that is needed to build a rich context information database.

7. CONCLUSIONS

We investigated the concept of stranger-sourcing on Social Media via a deployed application called MoboQ. Using MoboQ, a user can ask temporal and geo-sensitive questions, such as how long is the line at a popular business right now, and then receive answers that are crowdsourced from other users in a timely fashion. To obtain answers for questions, the system analyzes the live stream from the public microblogging service Sina Weibo to identify people who are likely to be currently at the place that is associated with a question and sends them the unsolicited question through the microblogging service from which they were identified.

Through nine months of real-world operation data, we verified the feasibility and advantages of building real-time social question-answering services on top of an existing microblogging platform, we analyzed users' behavior and preferences, we discussed lesson learned, and we proposed a list of future applications that could be built on top of the MoboQ platform. We foresee that there is great potential for MoboQ to become an important infrastructure component for building future context-aware applications.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Jeffrey Nichols at IBM Research Almaden for providing valuable suggestions and comments and Gang Chen, Rui Tao, and Shengyu Huang at Diggerlab Inc. for their great efforts on the system implementation.

REFERENCES

[1] E. Adar, et al. 00. Free riding on gnutella, First Monday vol. 5, no. 10-2 (2000).

[2] L. von Ahn. 2007. Human computation. In Proceedings of the 4th international conference on Knowledge capture. K-CAP '07. ACM.

[3] M. S. Bernstein, et al. 2010. Personalization via friend-sourcing. ACM Transactions on Computer–Human Interaction. vol. 17, no. 2 (2010), 1-28.

[4] J. Burke, et al. 2006. Participatory sensing. In Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications. WSW'06, ACM, Boulder, Colorado.

[5] A. T. Campbell, et al. 2008. The Rise of People-Centric Sensing. IEEE Internet Computing. 12, no. 4 (2008): 12-21.

[6] D. Chakrabarti, et al. 2011. Event Summarization using Tweets, In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media. ICWSM '11. 66-73.

[7] A. T. Crooks, et al. 2012. #Earthquake: Twitter as a Distributed Sensor System, Transactions in GIS. (2012).

[8] J. Feng, et al. 2004. Empathy and online interpersonal trust: A fragile relationship. Behavior and Information Technology. 23, no.2 (2004). 97-106.

[9] M. Guy, et al. 2010. Integration and Dissemination of Citizen Reported and Seismically Derived Earthquake Information via Social Network Technologies. Advances in Intelligent Data Analysis IX, Lecture Notes in Computer Science, 6065(2010), 42-53.

[10] D. Horowitz, et al. 2010. The anatomy of a large- scale social search engine. In Proceedings of the 19th international conference on World Wide Web. WWW '10. 431- 440.

[11] M. Jiang, et al. 2010. Human-centered Sensing for Crisis Response and Management Analysis Campaigns. In Proceedings of Information Systems for Crisis Response and Management, ISCRM '10. Seattle, WA, U.S.A.

[12] G. Koutrika, et al. 2007. Questioning Yahoo! Answers. Technical Report. Stanford InfoLab.

[13] H. Kim, et al. 2004. A comparison of online trust building factors between potential customers and repeat customers. Journal of Association for Information Systems. 5, no. 10 (2004): 392-420.

[14] J. Kleinberg. 2007. Challenges in Mining Social Network Data: Processes, Privacy, and Paradoxes. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '07. ACM. 4-5.

[15] B. Latane, et al. 1979. Many hands make light the work: The causes and consequences of social loafing. Personality and Social Psychology. 37, no. 6 (1979): 822.

[16] S. Liu, et al. 2010. Ushahidi Haiti & Chile: Next Generation Crisis Mapping. American Congress on Surveying and Mapping Bulletin, ACSM Bulletin 246.

[17] Y. Liu, et al. 2010. A crowdsourcing based mobile image translation and knowledge sharing service. In Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia. MUM '10. ACM, 6-15.

[18] Y. Liu, et al. 2012. Drawing on mobile crowds via social media. Multimedia Systems. vo.18, no.1, (2012): 53-67.

[19] C. MacDonald and I. Ounis. 2006. Voting for candidates: adapting data fusion techniques for an expert search task. In Proceedings of the 15th ACM international conference on Information and knowledge management. CIKM '06. ACM, 387- 396.

[20] A. Mockus, et al. 2001. Two case studies of open source software development: Apache and mozilla. ACM Trans. Softw. Eng. Methodol. vol. 11, no. 3, (2001): 309–346.

[21] M. R. Morris, et al. 2010. What do people ask their social networks, and why? A Survey study of status message Q&A behavior. In Proceedings of the 28th international conference on Human factors in computing systems. CHI '10. ACM, 1739-1748.

[22] J. Nichols and J. H. Kang. 2012. Asking questions of targeted strangers on social networks. In Proceedings of the ACM

2012 conference on Computer Supported Cooperative Work. CSCW '12, ACM, 999-1002.

[23] J. Nichols, et al. 2012. Summarizing sporting events using twitter. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces. IUI '12. ACM, 189-198.

[24] O. Okolloh. 2009. Ushahidi or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Participatory Learning and Action. 59, no. 1 (2009): 65-70.

[25] K. Panovich, et al. 2012. Tie strength in question & answer on social network sites. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work. CSCW '12. ACM, 1057-1066.

[26] P. Persson, et al. 2003 GeoNotes: a location-based information system for public spaces. Designing information spaces: The social navigation approach (2003): 151-173.

[27] T. Sakaki, et al. 2010. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In Proceedings of the 19th international conference on World Wide Web. WWW '10. ACM, 851-860.

[28] A. Schmidt, et al. 1999. There is more to context than location. Computers and Graphics. 23, no. 6 (1999): 893-901.

[29] P. Shankar, et al. 2012. Crowds replace Experts: Building Better Location-based Services using Mobile Social Network Interactions, In Pervasive Computing and Communications, IEEE International Conference on. PerCom '12. IEEE, 20-29.

[30] I. Smith, et al. 2005. Social disclosure of place: From location technology to communication practice. In Pervasive Computing (2005): 151-164.

[31] K. Tang, et al. Rethinking location sharing: exploring the implications of social-driven vs. purpose- driven location sharing. In Proceedings of the 12th ACM international conference on Ubiquitous computing. UbiComp '10. ACM, 85-94.

[32] T. Yan, et al. 2010. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In Proceedings of the 8th international conference on Mobile systems, applications, and services. MobiSys '10. ACM, 77-90.

[33] R. B. Zajonc. Social facilitation. Science 17 (1956): 65.