

QuestCrowd: A location-based question answering system with participation incentives

Venkata S. K. Pulla, Chandra S. Jammi, Prashant Tiwari, Minas Gjoka, Athina Markopoulou
CallT2, University of California, Irvine
{vpulla, cjammi, tiwarip, mgjoka, athina}@uci.edu

Abstract—We present QuestCrowd, a location-based system that uses crowd-sourcing to answer simple day-to-day questions. QuestCrowd supports realtime questions along with questions that can be answered using factual information by people not present at the location. Key design mechanisms that set it apart from similar location-based systems include (i) a reputation system that motivates quality contributors by rewarding high quality answers (ii) a forwarding mechanism that leverages existing social graph relations to increase participation.

I. INTRODUCTION

The extensive usage of smart phones and tablets coupled with widespread data connectivity has recently enabled the development of a large number of apps. Many of these apps fall under the category of location-based services that operate with knowledge of the user’s location. Apps like Foursquare and Google Places are gaining popularity among the masses. Apart from location based services, people also use their smart devices for information retrieval. They typically use web browsers to access search engines, which are mostly based on the library paradigm i.e. search results appear from matched content on the web. Thus there is a lack of dynamicity of the information that could be delivered to the users.

The crowd-sourcing paradigm has been successfully adopted in various services that provide efficient and dynamic information sharing among the individual members of the crowd. In this paper we present QuestCrowd, a location-based system that uses crowd-sourcing to answer simple day-to-day questions. The goal of QuestCrowd is to enable users to ask a question tied to a specific location and get a reply close to real-time. Questions that require low effort to respond are expected i.e. the reply can consist of a small message or even a picture. Some examples include “What is the current line outside IMAX 3D HFR for ‘The Hobbit’ at Irvine Spectrum?”, or “How crowded is Laguna Beach?”. Location subscription and tracking are used to identify users in the locations of interest. That avoids excessive load on the network and enables information delivery system with a fine granularity i.e. questions are sent to people that are likely to offer a good answer.

Some of the aspects of QuestCrowd bear resemblance to existing systems and services. Localmind [1] is a mobile location-based Q&A service. However, it lacks participation incentives and mechanisms to leverage existing social interactions. Aardvark [4] is a social search engine that aims to find the right person to answer a question. It is not location-based and also lacks strong participation incentives. SocialTelescope [6] is a location-based search system for places. The text and

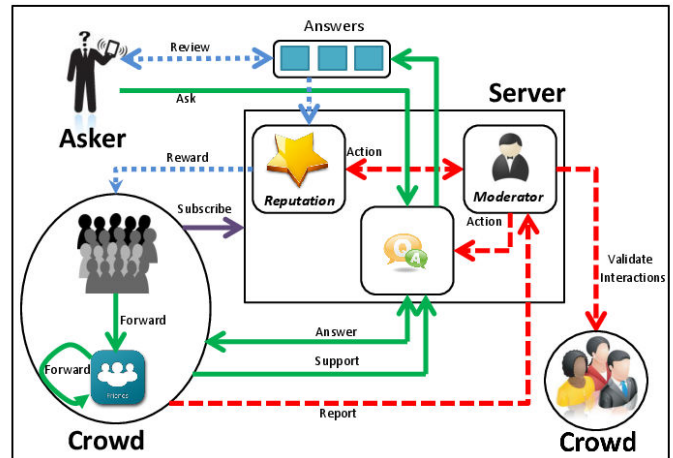


Fig. 1. System Architecture

location from tweets in combination with frequency of visits is used to rank places. [2] builds a question answering system on top of Twitter where questions are forwarded to randomly selected Twitter users that live in a certain area. Although the reply percentage is quite low (5%), the answer rate is very high (75%). [5] presents a survey of the KiN Q&A system and finds that users are largely divided into askers and responders, participation is intermittent and a reward system is a good motivator.

Compared to other Q&A systems, QuestCrowd contains the following novelties: (i) it uses a reputation system that aims to incentivize individuals to contribute their part by answering questions and helping the crowd. A detailed description about the incentives model is discussed later. (ii) it leverages the existing social networks to increase participation. The user is encouraged to forward a question to friends that are likely to answer it.

II. SYSTEM DESIGN

Three components, *Asker*, *Server*, and *Crowd* collectively define the entire system as shown in Fig. 1. The *Asker* is the user who asks a question to all the users in the system or the *Crowd*. The *Server* is an important component that provides a well defined bridge between the asker and the crowd with the help of three modules - *Reputation*, *Q&A* and *Moderator*. Along with these, every user in the system subscribes to locations for which they can answer questions, and which the *Server* stores.

The *Q&A* module is the central part of the server which takes the questions posted by the asker and directs it to the crowd, based on their subscriptions or the social network linkage. When the crowd responds back with an answer to a question or a support to a provided answer, the *Q&A* module

holds the responsibility of storing and delivering these to the asker. When the *Asker* receives an answer, he can review it. The review will be tracked by the *Reputation* module that rewards the deserved users with their corresponding reputation values. The different user actions such as answer, support etc. have different weights and are used in the calculation of user reputation values (see section III). Apart from these, when the crowd reports that a question or an answer is inappropriate, it is handled by the *Moderator* module of the server. It studies the reports and takes appropriate action on both the question or answer and the reputations of the involved users.

In addition to the above mentioned set of user actions, we support two unique and powerful actions - *Forwarding* and *Validations*. Hence, when a user in the crowd gets a question and knows that another person in his social network would be able to answer it, he can forward it to that user with minimal effort. On top of that, the forwarder will also be rewarded when the asker is satisfied with the response from the forwarded person. Also, when the reputation module detects a strange behavior in the reputations of a set of users, it will ask the *Moderator* module to verify their credibility. Then the *Moderator* will choose a highly reputed, random subset of crowd that is least connected to the given set of users and request them to volunteer for the validation process. Hence, this module leverages the crowd to prevent malicious activities in the system.

III. REPUTATION

The reputation system is a powerful mechanism that acts as a motivator for users to contribute and as an inhibitor for malicious activities in the system. The reputation model of our system is primarily dependent on the weights of different types of user activities described earlier and is inspired by the sociogram model in [3]. It is calculated in periodic intervals, i.e. daily.

The process of the calculation of a new reputation value $R(w)$ for a user w starts with the estimation of the influence of all other users in the system on his reputation. For this, we calculate the average user reputation in the system R_{avg} and then use it to find the influence of any other user v on user w as follows:

$$Infl(v \rightarrow w) = 1/2 + \arctan(R(v) - R_{avg})/\pi$$

For each user pair (v, w) we take the average of weights $S(v, w)$ of all the different interactions that happened between v and w in the last one day so as to prevent collusion attacks. Then the new reputation value of a user is calculated as:

$$R(w) = R_{old}(w) + \sum_v S(v, w) \cdot Infl(v \rightarrow w)$$

The reputation of a user is important because it decisively influences the display ranking of a question asked by the user. It can make a question get noticed faster or push it down for a less probable viewing by the crowd. The display ranking is different for different users and is based on three key metrics.

1) Reputation quality: We normalize the reputation value to quantify the relative priority w.r.t. the rest of the crowd $R_{qual}(w) = R(w)/max_v(R(v))$

2) Response probability: In order to enhance the likelihood of response for a question asked, we consider two factors - the popularity of the location at which the question was asked and the responsiveness of a user in answering questions about that

location. For a question q viewed by user w at location λ we compute the metric $P_{resp}(q, w) = Resp(w, \lambda) \cdot Popularity(\lambda)$.

3) Time: We compute the staleness of a question q as $Stale(q) = (1 - t_{elapsed})$

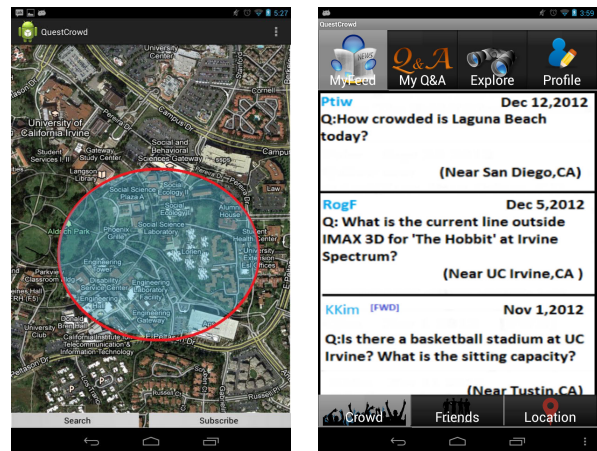
Using these metrics, the score of a question q asked by user v and viewed by user w is calculated by

$$Score(q, v, w) = [R_{qual}(v) \cdot P_{resp}(q, w) \cdot Stale(q)]^F \quad (1)$$

where $F = 0.5$ if q is forwarded from v to w ; else $F = 1$.

$Score(q, v, w)$ is calculated regularly for every question that a user is subscribed to answer. Questions are ranked in descending order of $Score$. Hence the calculation of $Score$ is a key design mechanism that will make the system give priority to the users who are contributing to the crowd while inhibiting inactivity and maliciousness in the system.

IV. ANDROID APP



(a) Location Area Subscription (b) Feed of Questions

Fig. 2. Screenshots of the Android app

Here we present two interfaces from our Android app prototype. Fig. 2(a) allows users to subscribe to an area of expertise, typically home, work. Fig. 2(b) is the default screen of the app and shows a stream of recent questions that require an answer, ordered by Eq.(1).

V. CONCLUSION

We presented QuestCrowd, a crowd-sourcing Q&A system suitable for real-time questions with mechanisms that incite participation. We plan to release QuestCrowd as an Android app and execute a pilot project around the University of California, Irvine.

REFERENCES

- [1] Localmind (<http://www.localmind.com>).
- [2] M. Bulut, Y. Yilmaz, and M. Demirbas. Crowdsourcing location-based queries. In *Proc. of PerCom Workshops '11*, 2011.
- [3] W. Chen, Q. Zeng, L. Wenyin, and T. Hao. A user reputation model for a user-interactive question answering system. *Concurrency and Computation: Practice and Experience*, vol. 19, 2007.
- [4] D. Horowitz and S. Kamvar. The anatomy of a large-scale social search engine. In *Proc. of WWW '10*, 2010.
- [5] K. Nam, M. Ackerman, and L. Adamic. Questions in, knowledge in?: a study of naver's question answering community. In *Proc. of CHI*, 2009.
- [6] P. Shankar, Y. Huang, P. Castro, B. Nath, and L. Iftode. Crowds replace experts: Building better location-based services using mobile social network interactions. In *Proc. of PerCom '12*, 2012.