Crowds replace Experts: Building Better Location-based Services using Mobile Social Network Interactions

Pravin Shankar[1], Woody Huang[2], Paul Castro[2], Badri Nath[1], and Liviu Iftode[1]

[1] Dept. of Computer Science, Rutgers University.
{spravin,badri,iftode}@cs.rutgers.edu
[2] IBM T J Watson Research Center, Hawthorne, NY.
{ywh,castrop}@us.ibm.com

Abstract. Location-based services are growing in popularity due to the ubiquity of smartphone users. The relevance of location-based query results is very important, especially for mobile phones with limited form factor. Location-based data is real-time and highly dynamic; this introduces challenges in indexing and ranking places. The growing popularity of mobile social networks, such as Twitter, FourSquare and Facebook Places, presents an opportunity to leverage user interactions on these networks to build better location-based services. In this paper, we present SocialTelescope, a location-based service that automatically compiles, indexes and ranks locations, based on user interactions with locations in mobile social networks. We implemented our system as a location-based search engine that uses geo-tweets by Twitter users to learn about places. We evaluated the coverage and relevance of our system by comparing it against current state-of-the-art approaches including page-rank (Google Local Search), expert-based (Zagat) and user-review based (Yelp). Our results show that a crowd-sourced location-based service returns results that are at least as relevant as those returned by current approaches, at a substantially lower cost.

1 Introduction

Modern mobile devices contain global positioning systems (GPS) and Internet connectivity via 3G and WiFi, which allows them to make location-based queries. A location-based service allows users to search for places based on keywords (such as restaurants, bars, museums, jogging trails, etc.). Due to the growing number of smartphone users, location-based services are growing in popularity.

Location-based services face two key challenges, viz. how to collect up-to-date information about places, and how to rank places. Information about any place typically changes all the time. For example, given a restaurant entry in the yellow pages, its menu, chef, management, quality of food, prices, and, as a result, its popularity can change periodically. It is crucial for a location-based service to maintain this up-to-date information about places. Because of the limited real-estate in mobile phones, ranking of location-based query results becomes very important, since a mobile user would find it very hard to scroll beyond the top few results.

Popular web ranking schemes such as Pagerank [27] and HITS [19] cannot be trivially applied for ranking locations, since there is no inherent link structure between locations. As a result, current popular location-based services, such as Google Places (formerly Google Local), Yahoo! Local and Bing Local, use a combination of several unpublished criteria for ranking places[32]. Further, these criterion are revised periodically so that the systems continue to serve relevant results.

An alternate approach used by some location-based services is to rely on experts, for example, Zagat ratings of restaurants [42]. Such an approach incurs high monetary cost. It also lags in timeliness, because the time of review could be several months in the past. It is also susceptible to expert bias. Another alternate approach, which has seen lot of popularity, is to rely on reviews by all users instead of just experts, for example Yelp reviews [39]. Relying on reviews introduces several problems, such as, how to distinguish genuine users from marketeers [28] and how to handle negative reviews [40, 41].

Fortunately, people are good judges of the places which they frequent. The interactions of users with location in mobile social networks can act as a rich source of information about people's preferences for places, and presents a novel approach to build a location-based service. In the recent past, there has been a huge popularity of location-based mobile social networks such as FourSquare and Gowalla, which allow users to checkin to locations. Foursquare reports 200 million checkins [11], which doubled from 100 million in just over two months [12]. As a result, the traditionally popular online social networks such as Facebook, Twitter and Google have now added location as a standard feature.

Our work is motivated by the question: *Can we build a location-based service by making use of mobile social network interactions?* In this paper, we introduce SocialTelescope, a novel location-based service that is built using user interactions with locations in mobile social networks. User interactions act as implicit feedback about locations, and are used to maintain and index information about locations.

On receiving a location query, the system first computes a candidate set of places based on matching user tags. Next, this candidate set of places is ordered based on popularity. SocialTelescope does not consider all users to be the same, when computing popularity of a location. Instead, users are assigned a score based on their expertise on the search keyword. For a given search keyword, SocialTelescope assigns expertise scores to users as the number of times that user has visited any place that matches the search keyword. The intuition behind computing expertise is that, say, when ranking restaurants that serve seafood, people who visit lots of seafood restaurants can be considered seafood connoisseurs, and so their choices can be given a greater weight.

For mobile social network interactions, we make use of a dataset we crawled, which comprises geo-tweets by all Twitter users in New York city over a period of two months. With this dataset as a starting point, we build a location-based service, SocialTelescope, that can answer location-based queries for different types of locations, such as food, nightlife, shops and parks.

Two key measures of the success of the proposed location-based service are coverage (*how complete and up-to-date is the information about locations?*) and relevance (*how interesting are the top results to a query?*). We evaluated the coverage and relevance of our system by comparing it against current state-of-the-art approaches including expert-based (Zagat), user-review based (Yelp), and hybrid (Google Local Search). Our results show that a crowd-sourced location-based service has coverage and relevance comparable to existing approaches at a substantially lower cost.

The key contributions of this paper are:

1. We show how a location-based service that supports rich, dynamic queries can be built by just using user interactions in a mobile social network as the building block.
2. We design an algorithm for ranking places based on their popularity among users, by giving weights to users proportional to their expertise.
3. We conduct and present results from an evaluation study that compares a crowd-sourced location-based service with current state-of-the-art systems including expert-based (Zagat), user-review based (Yelp), and hybrid (Google Local Search).

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 presents the system design in detail, including a motivating scenario, the system architecture and the ranking algorithm. Section 4 describes the details of the system implementation, including the mobile social network dataset that we crawl to bootstrap our implementation. Section 5 presents the results from our evaluation study. We discuss open issues in Section 6, and conclude and present directions for future work in Section 7.

2 Related Work

The immense popularity of social networks has spurned a new wave of research in various areas, including mobile computing, sensing systems, data mining, recommender systems, security and privacy. Our work is primarily motivated by this growing popularity of social networks, as well as the slew of interesting research in this field.

People sensing: Mobile sensing has now transformed into people-sensing, by being cognizant of the person carrying the sensing device. Visionary works such as participatory sensing [3], Mobiscopes [1], and MetroSense [9] proposed the idea of using mobile phones to perform people-centric sensing. People-centric sensing systems have been designed for various tasks, such as the CenceMe system [24] for activity sensing, ICartel [35] for sensing road traffic using mobile phones, and Micro-Blog [14] for sharing in-situ multimedia content in real-time.

Leveraging social interactions: The Reality Mining project showed how cellphone interactions can be used to learn about social connections of humans [8]. The Senseable City project [29] is investigating how digital traces of human activities can be used to better understand how a city functions. For example, aggregate mobile phone network activity can help estimate the presence of visitors [15, 18]. More recently, researchers have started looking at user activity in social networks as another source of social interactions which can be leveraged [30].

Crowdsourcing: Crowdsourcing has emerged as a powerful model to problem-solving by outsourcing tasks to a large group of people, as engagingly articulated in [37]. PeopleNet [25] was an early system that answered queries by leveraging social networks. Amazon Mechanical Turk [23] is a popular commercial marketplace for outsourcing tasks that require human intelligence. Crowdsourcing has been used for various tasks such as image search [38], mining to identify significant events [4], tourism [44], and improving product reviews using social networks [22].

Geo-social networks: Most of the popular social networks such as Facebook [10] and Twitter [36] have now added location as a primary entity, inspired by the popularity of location-based social networks such as FourSquare [13] and Gowalla [17]. Several recent works have modeled and analysed these geo-social networks [5, 2, 31]. Mapping meaningful names to places is an interesting problem that arises out of these geo-social networks [26, 21].

Location-based services: Location-based services can be improved by leveraging user interactions in social networks, as we do in our work. Closely related to our work are two recent works[7, 20]. In [7], the authors mine the community-authored reviews in Yelp [39] to identify potential activities supported by different locations. Further they show how a context-aware location-based service can leverage this information. Hapori [20] is another context-aware location-based service that makes use of context such as user activity, time, and weather, to return results that relate better to the personal taste of the user.

Privacy issues in location sharing: The act of sharing the current location by users of mobile social networks can potentially raise several privacy concerns. In [33], the authors describe useful guiding principles for creating a location-sharing service based on their pilot study. More recently, [34] explains how social-driven sharing (such as FourSquare) is fundamentally different from purpose-driven sharing (such as Google maps).

3 SocialTelescope Design

3.1 Motivating Scenario

Consider a tourist carrying a smartphone who has just reached New York city. He makes several queries to a location-based service from his smartphone. First, he wants to indulge in some tourism, so he queries for an interesting place to visit close to where he is. He next makes a query for a good seafood place to have dinner. Finally he searches for a bar which plays good live music.

In an ideal scenario, the tourist would either ask his friends for their suggestions, or go to each place and see how popular it is. In our scenario, the tourist

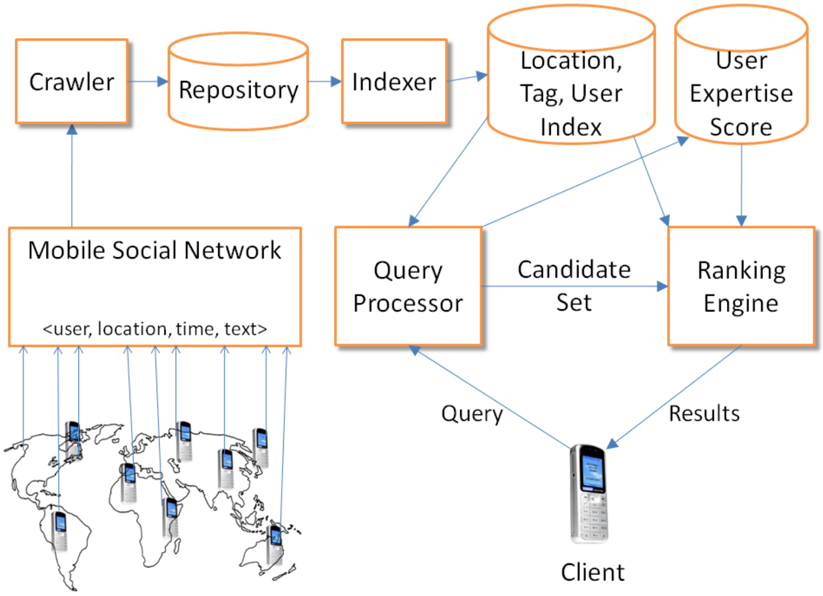


Fig. 1. High-level Architecture of SocialTelescope.

does not have any friends who are familiar with places in New York city. In addition, he does not have the luxury of time to actually visit multiple places before making his decision. The goal of the location-based service is to provide the most relevant results to all the queries by the user, which would match the results he would obtain in the above-mentioned ideal scenario.

3.2 Architecture

SocialTelescope is a centralized system that receives location-based queries from users, and replies to these queries with a list of places. Figure 1 shows the high level system architecture. Incoming queries from clients are handled by the query processor which computes a candidate set of results. The ranking engine then sorts these results by relevance and returns the best matches back to the requesting client.

In the background, the crawler continuously crawls a mobile social network for user interactions and stores them in a repository. These interactions are in the form of $< user, location, time, text >$ tuples. The indexer module updates the indexes each time a tuple is added to the repository. The indexer module maintains indexes corresponding to location (R-tree), tags and user names.

Each time a query is received, the ranking module computes a user expertise score for each user based on the query tags. For efficiency reasons, the user expertise scores can be pre-computed for popular queries, and they can also be cached for repeated queries. The ranking engine uses the location index and the user experise score to rank the candidate set of results by popularity weighted by user expertise.

We now explain each module of the system in more detail.

3.3 Crawling

The goal of the crawler is to record all public user interactions in mobile social networks, and store them in a repository. The main technical challenges are scalability and robustness to service failure. The crawler can scale to a large geographic region by parallelizing it. Spacial regions provide a natural hierarchy for parallelizing, while not needing to maintain any shared state.

Crawling also involves social challenges. The crawler needs to be cognizant of the social network service that it is crawling, as well as the users who's public information it is crawling. Unlike in the case of web crawling, social network services typically provide well-defined APIs for external applications to crawl their public data. The crawler is expected to adhere to the rate limits specified by the service it is crawling.

In our work, we assume that users of mobile social networks do not mind if the location-based data that they make public is used by an external service in an aggregate form to provide additional utility to all users. Prior research has studied privacy issues involved in sharing location [33, 34]; we consider location privacy beyond the scope of this paper, while acknowledging that users of social networks could unintentionally make private location data public.

3.4 Indexing

Text entered by a user in a mobile social network is converted into a set of tags. Term extraction using linguistic techniques is a well studied problem in the Information Retrieval field, and several open-source tools are available for term extraction. These tags serve as user-specified annotations for the user location.

User locations in a mobile social network could be represented either as geographic coordinates, or as semantic places (such as building, restaurant, outdoor). Geographic coordinates is converted into semantic places to better index the properties of a geographic region. Most of the popular social networks (such as Facebook, Twitter, FourSquare) provide functionality to make this transformation.

The system maintains an index for three types of entities: locations, tags and users. The corpus of data crawled from the mobile social network activities can be accessed using user, tag or location as index. By using the user index, it is possible to find a user's mobility profile as well as interests profile. Similarly a location's profile can be found by using the location index.

3.5 Query Processing and Ranking

On receiving a location query as request, the system performs two tasks: first a candidate set of places is computed, and then this set is ordered using

```
Get the list of places $L$ which have the search keyword $q$ as a tag
foreach user $u$: UserExpertise($u$) = number of visits by $u$ to any venue in $L$
Order each place in $L$ by number of user visits weighed by UserExpertise($u$)
Return the top $k$ results in $L$
```

Algorithm 1: **SocialTelescope Ranking Algorithm**

a ranking algorithm. Algorithm 1 describes the ranking algorithm. For ease of explanation, we will restrict a query to just one keyword. We will later remove this restriction and explain how multiple keywords are handled.

The query processor module finds a candidate set of places which have the keyword in the user query as a tag. This set of location now needs to be ranked. The ranking module finds the list of people who have ever visited the places in the candidate set. A naive algorithm could simply order the set by user count, thereby ranking places based on their popularity.

When computing popularity of a location, SocialTelescope does not consider all users to be the same. Instead, when a query arrives, users are assigned a score based on their expertise on the search keyword. Users' expertise on a search keyword is automatically inferred, based on their past interactions with locations of this type. The expertise score of a user signifies how valuable that user's vote is in determining the venue's popularity. The venues are ranked by popularity weighted by the user expertise score.

We define user expertise score for a query keyword as the number of times that user has visited any place that matches that keyword. User expertise can be learnt in several ways. We use the above definition because it is a good approximation and can be computed automatically from our dataset. The intuition behind this definition is that, say, when ranking restaurants that serve seafood, people who visit lots of seafood restaurants can be considered seafood connoisseurs. When answering queries about seafood restaurants, these users can be considered the authorities, and so their choices can be given a greater weight.

Multiple keywords in the query string can be handled by minor extensions to the above algorithm. The candidate set of places is chosen as the venues that contain all the query keywords as tags. Similarly, the ranking algorithm computes the combined user expertise score for all the keywords in the query string.

4 Implementation

Crawling.

To build our location-based service, we need a dataset of user interactions on mobile social networks. We built this dataset by crawling the popular social network Twitter. Twitter recently inroduced geo-tagging feature which allows users to add their current location to their tweets. Our crawler collected all geo-tweets made by any Twitter user in the New York Manhattan region for a period

Start Date	June 14, 2010
End Date	August 20, 2010
Bounding box for geo-tweets	(40.703,-74.022),(40.879,-73.899)
Total geo-tweets in the region	198919
Total number of distinct users	15659
Total FourSquare checkins corresponding to the geo-tweets	43461
Total number of distinct FourSquare users	6451

Table 1. Details of the SocialTelescope dataset.

of two months. Table 4 describes the details of our dataset. In a little over two months, about 200,000 geo-tweets were made in New York city, by about 15,000 users.

The crawler made use of Twitter's documented API to get the geo-tweets by specifying a bounding box. The twitter web service, as would be expected of any service of its scale, goes down quite frequently, and the crawler needs to be robust to handle all the errors gracefully. Twitter rate limits all accesses to its API; we contacted Twitter and our username and IP addresses were whitelisted by Twitter so that our crawler is not limited by the default rate limits.

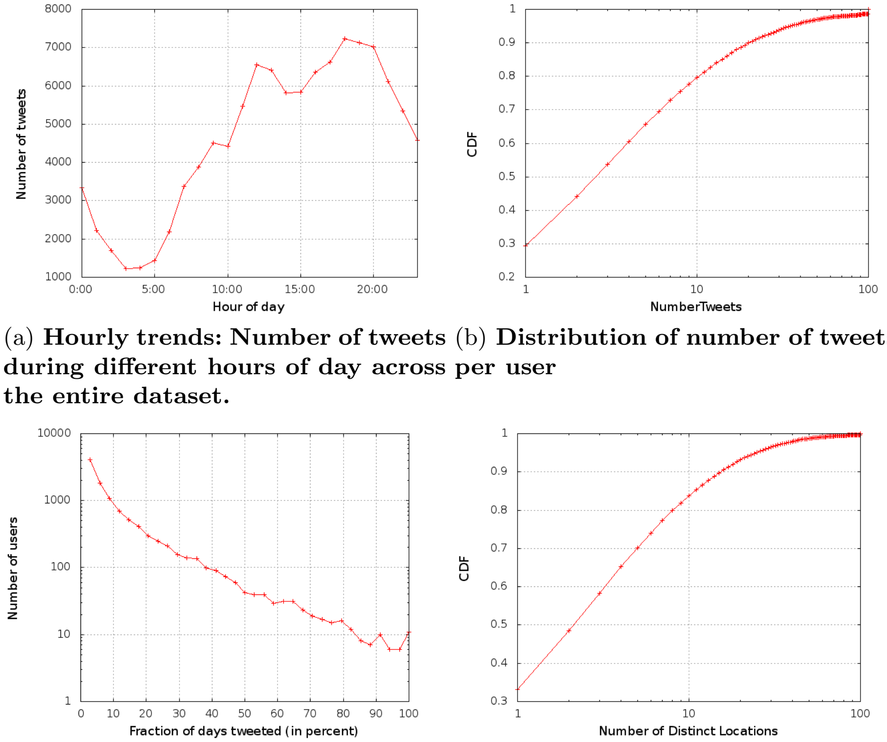
Despite the whitelisting, the crawler needs to be parallelized to scale to a large geographic region. Our implementation runs four parallel crawlers, which we observed to be sufficient for crawling the entire region of New York city. We chose not to perform any sanity checks for duplicate or erroneous data during crawling, so that the crawler could be easily parallelized. The crawler converts each user tweet into records of the form $< user, location, time, text >$.

Figure 4 describes the properties of the mobile social network interactions in our dataset. The temporal frequency of tweets follows a clear trend based on typical work hours, as shown in Figure 2(a). Most tweets are made in the evenings after work hours, with a small peak at lunch time. We analysed the day-of-week-trends and did not find any significant differences between weekdays and weekends. We believe that this is probably due to the large number of tourists who are present in New York city almost every day.

Out of the about 15,000 users in our dataset, not all users are equally active. As shown in Figure 2(b), the number of tweets by a user follows a clear power law, with a few power users making a large number of tweets and most users making less than 10 tweets in these two months. We analyse the number of days an average user is active during our two month period of data collection. Figure 2(c) shows that there is a small set of power users who are active every single day during this period. Finally, we analyse how mobile a typical user is in Figure 2(d). We once again notice a power law with a very sharp exponent. The figure shows that half the users tweet from not more than three distinct locations.

Converting Locations to Places using FourSquare.

We need to obtain semantic place names for each location in our crawled dataset. As shown in Table 4, we observed that almost one out of four tweets in



(a) **Hourly trends: Number of tweets during different hours of day across the entire dataset.**

(b) **Distribution of number of tweets per user**

(c) **Distribution of active days per user. An active day refers to a day when a user made at least one tweet.**

(d) **Mobility profile per user: number of distinct locations where a user tweeted from.**

Fig. 2. Properties of the New York city geo-tweet dataset

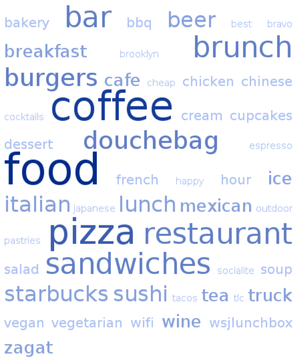
our dataset were made using the FourSquare client. These users can be considered active users of FourSquare, since they linked their FourSquare accounts to their Twitter accounts.

To convert the locations in our dataset to semantic place names, we leverage this trend of the immense popularity of FourSquare. This is done by simply looking up FourSquare for the venue id as specified in the tweet text. FourSquare has a large repository of place names with a list of categories and tags corresponding to each place. This entire repository is created by the users, and verified for accuracy by other users. This vast and accurate repository is just another evidence of the immense power of crowds.

Generating Tags for Locations.

We used two sources to generate tags for locations, viz. the user-specified tags in Foursquare and the text of the tweets. We obtain the list of user-specified tags for locations from FourSquare to populate our tag repository, and for each of these tags, we augment the frequency by parsing the text of each tweet in our dataset. Text in a tweet is limited to 140 characters. Due to this restriction, twitter users commonly use abbreviations and spelling/grammar errors. In our implementation, we ignore such errors.

Figure 3 shows the distribution of tags in our dataset, in the form of a tag cloud (Figure 3(a)) and a top-ten list of tags (Figure 3(b)). Most of the popular tags are as expected. One clear anomaly we found was that the third most popular tag is not a typical tag to describe a place. On further investigation, we found that this is a known problem to FourSquare [6]. The reason, as we later found, was a social *meme* which encouraged users to tag places that they did not like with this keyword.



(a) **User tags for locations in the form of a tag cloud**

bar	385
coffee	376
douchebag	355
pizza	340
beer	242
food	223
restaurant	206
brunch	181
gym	166
brooklyn	162

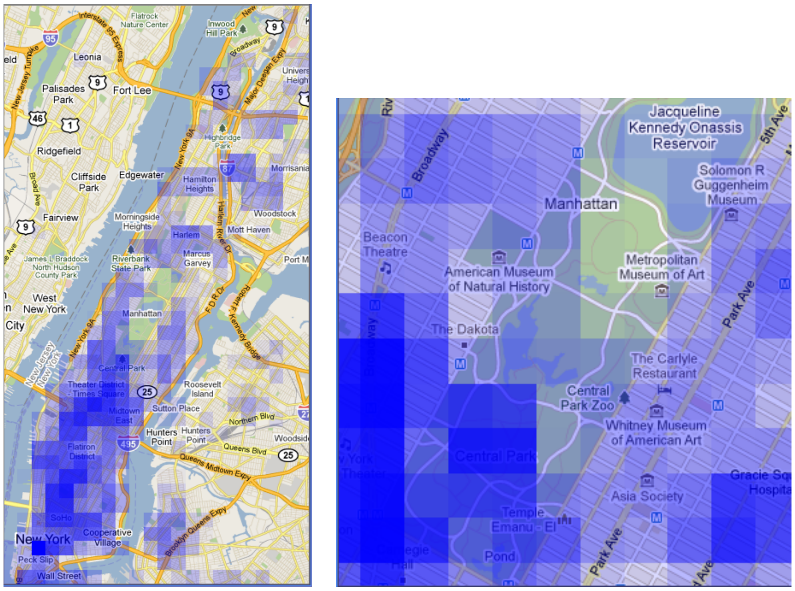
(b) **Top ten most popular user tags for locations**

Fig. 3. Trends in user tags for locations

Indexing.

The indexing module creates three indexes for our dataset: locations, tags and users. An R-Tree is used to index the locations by geographic coordinates. Our implementation uses the Postgresql database for storing and indexing the data, and the PostGIS spacial extensions for indexing the geographic coordinates.

Figure 4 shows the geographic distribution of tweet frequency in the form of a heatmap. The color intensity in the heatmap corresponds to the frequency of tweets in the corresponding location block. Figure 4(a) shows the heatmap corresponding to the entire region of interest of our dataset. Some obvious trends that can be seen are that New York city is a lot more popular than the neighboring New Jersey; Manhattan is lot more popular than all the other burroughs; and within Manhattan, the tourist regions are the most popular locations. Figure 4(b) shows a heatmap zoomed in to lower Central Park, which shows slightly more non-obvious popularity trends. The location granularity in our dataset cannot be reduced to below 100 m due to the imprecision of GPS location in smartphones.



(a) **Heatmap of the entire region.**

(b) **Heatmap zoomed in to Central Park to show trends at a finer granularity.**

Fig. 4. Heatmap of the New York city region based on geo-tweets.

User Interface.

The location-based search interface is implemented as a web-based client using PHP and PostgreSQL. The user interface is minimalistic; users just need to enter a keyword, and can optionally select a location, category and subcategory. Locations are resolved to geographic coordinates using Google geocoding API [16]. We construct a list of categories and sub-categories based on data fetched from FourSquare. Users can optionally prune their searches by specifying category/sub-category. The web interface is publicly accessible at http://www.socialtelescope.com/socialtelescope.

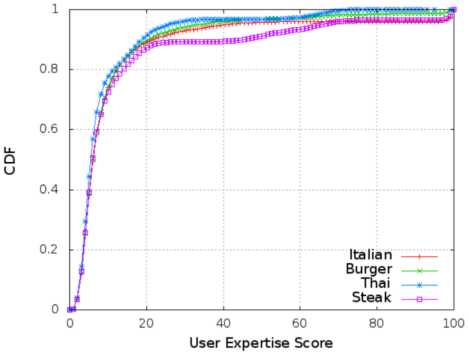
5 Evaluation

5.1 Evaluation Goals and Metrics

The goal of a location-based service is to return relevant results efficiently. Current location-based services are quite efficient and can scale to millions of users

Query	# Matches	# Experts
Barbecue	65	116
Burger	166	238
Japanese	237	182
Indian	70	61
Seafood	85	60
Mexican	212	140
Chinese	165	84
Steak	78	28
Thai	102	31
Italian	332	175

(a) **Number of place and expert matches in dataset for the 10 test queries.**



(b) **Distribution of User Expertise Score for different queries.**

Fig. 5. Details of test queries using Zagat Top Places.

easily. In our evaluation, we focus on the quality of the results returned by the service.

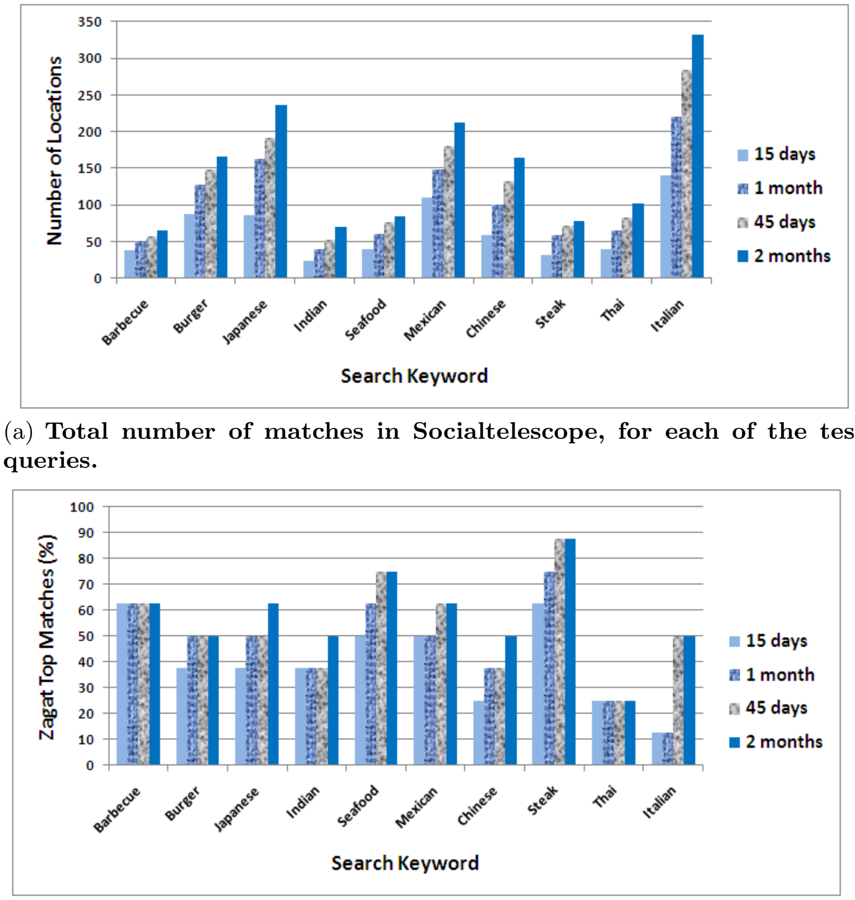
There are two key measures of the quality of a location-based service: coverage (how complete and up-to-date is the information about different locations) and relevance (how relevant are the top results to the query). Coverage and relevance can be measured quantitatively, only if we have ground truths available. Rank order of results to a query tends to be subjective, and it is hard to define ground truths.

5.2 Methodology

We evaluated the precision and recall of our system by focusing on restaurant search. We choose ten specific queries that are commonly used when searching for restaurants, as described in Figure 5. For each of these ten restaurant queries, we get the Top Spots list from Zagat [43]. This list is compiled by hand for different categories of restaurants by Zagat editors. In order to quantitatively measure coverage of SocialTelescope, we assume Zagat Top Spots list to be the ground truth.

We compare our results against two state-of-the-art approaches, viz., user-review based (Yelp), and page-rank based (Google Local Search). To make the comparison, we consider the results from an expert-based approach (Zagat) as the baseline.

Finally, to gain more insights into the relevance of results, we collected user feedback by interviewing a cross-section of users who used our system to query for restaurants in New York.



(a) **Total number of matches in Socialtelescope, for each of the test queries.**

(b) **Relevance of results returned by SocialTelescope, measured as fraction of Zagat Top Places that are contained in the result set.**

Fig. 6. Coverage and relevance of locations in SocialTelescope.

5.3 Coverage and Relevance Results

Table 5(a) lists the ten queries that we choose for our evaluation. The table shows the number of candidate locations that are returned from our dataset for each of these queries. The table also shows the *number of experts* in our dataset for each of these queries. All the ten queries result in a reasonably large number of candidate locations, which is the starting point for our ranking algorithm.

We computed the *user expertise score* for each (user,query) pair, as the number of times that user has visited any place that matches that query string (as described in Section 3). We considered users with a score greater than 5 as experts. Figure 5(b) shows the distribution of user expertise scores for a sample of

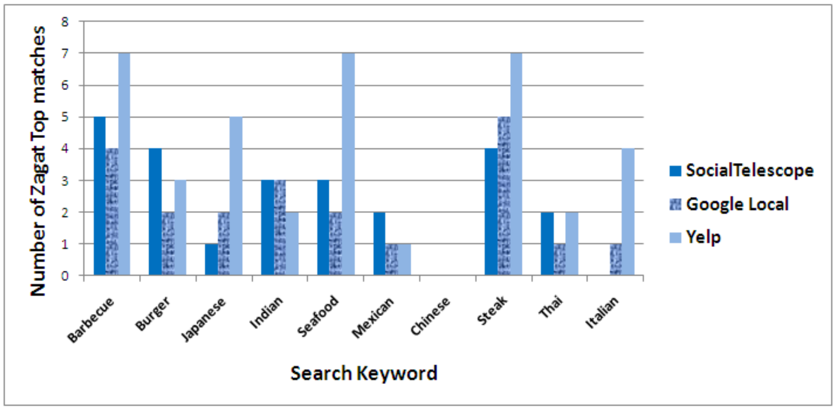


Fig. 7. Comparison of ranking results of SocialTelescope, Google Local Search and Yelp, by asuming the Zagat Top Places list to be the ground truth. The figure shows that there are overlapping results, as well as disagreements, between these approaches.

the test queries. The figure shows that the distribution of user expertise score follows a power law, and it is similar for queries with different number of experts.

Figure 6 shows the coverage and relevance results. We measure coverage as the number of candidate locations that are returned by our system for a test query. Recall that our system is based out of just two months of user interactions on Twitter. We further analyse how the results are affected by the quantity of user interactions in our dataset, by considering just the first 15 days of user interactions, first month, first 45 days, and the entire 2 month dataset, respectively.

Figure 6(a) shows that the coverage of our system increases with the quantity of user interactions. This is an expected result, since, over time, users will travel to more locations. We expect that the coverage will eventually saturate, but further experiments based on a larger time span of data are required to confirm this.

To measure relevance, we find out, for each of the ten queries, how many of the Zagat Top Places are present in SocialTelescope's candidate set of locations. Figure 6(b) shows the results of this analysis. We notice that for all but one query, SocialTelescope contains over 50% of the Zagat Top Places. Even more interestingly, a small quantity of user interactions (just 15 days) is enough to get highly relevant results. This result shows that there is a high degree of correlation between the experts' view (Zagat) and the crowd's view (SocialTelescope). This lends credence to our claim that user interactions in mobile social network can serve as a low cost alternative to current approaches for building a location-based service.

To gain some insights into how different ranking algorithms compare against each other, we analyze how SocialTelescope, Google Local Search and Yelp rankings perform in comparison to each other. To make this comparison, we assume Zagat Top Places to be the ideal results, and measure how many matches are returned by the different approaches. Figure 7 shows the results of this comparison. The figure shows that there is both a high degree of overlap, as well as disagreement, between results from different approaches. This is not unexpected, since ranking tends to be subjective in nature.

5.4 User Feedback

To better understand how relevant are the results from SocialTelescope, compared to other approaches, we gave our system to users for testing and then interviewed them. The users could see three sets of results (SocialTelescope, Google Local Search and Yelp), without knowing which set corresponded to which approach. We now report some interesting feedback from this study:

- SocialTelescope performs better than the other approaches for queries which are subjective in nature. For example, the keyword *view* led to good matches of restaurants with a nice view. This shows that, while Zagat and Yelp are great for getting ratings of places for fixed categories, SocialTelescope is able to generate dynamic categories using crowdsourcing.
- A type of query for which SocialTelescope did not perform well was when a user wanted only places with a speciality cuisine. For example, one of the highly ranked results for the keyword *sushi* is Whole Foods Market, which is a popular location but not a speciality sushi restaurant. The reason is that SocialTelescope is naturally biased towards locations that get a lot of visitors. To remove this bias, we need a way to normalize large and small places.

6 Discussion

Explicit vs implicit user feedback: Approaches that depend on explicit feedback in the form of user reviews (such as Yelp) face several challenges. Businesses could pay reviewers to write positive reviews about themselves [28]. Negative reviews impact businesses, so businesses could be forced to pay money to hide the negative reviews [40]. In addition, users who post negative reviews about an establishment could be susceptible to lawsuits [41].

SocialTelescope does not rely on explicit user reviews; instead we leverage implicit actions by users in social networks. We believe this is a more natural way to gather user feedback about locations. Users do not have to do anything special; they simply check in to locations and tag the places while socializing with their friends. Many of the problems with explicit user feedback can be solved by using implicit user feedback. However, spam detection and reputation tracking of users in social network users remains an important problem.

Cost of building and updating a location-based service: Current location-based services incur a significant cost in collecting and updating information about locations. The cost involved is both monetary, as well as in terms of timeliness. Our approach is a contrast to the existing approaches, leveraging interactions that users already perform on location-based social networks.

We showed in our work that by leveraging two months of user interactions in a densely populated region, we are able to achieve results that are comparable to existing services such as Yelp, Zagat and Google Local Search. We believe that, with user interactions over a larger period of time, the performance of our approach will further increase. To give more importance to recent user interactions, a time decay function can be added.

Personalized results: In our approach, the querying client does not need to reveal any personal information to the service. Several related works, such as Hapori [20], make use of user context such as the current activity performed, user likes and dislikes, to return results that are personalized to the user. While this is a promising idea, it requires the querying clients to reveal their identity to the service. Purpose-driven sharing is different from social-driven sharing [34]. This is also evident from the low adoption of personalized services such as Google Personalized Search. If privacy concerns of users can be alleviated, personal context-information of the querying client could be used to return personalized results.

7 Conclusion and Future Work

We proposed a novel way to build better location-based services by leveraging user interactions in location-based social networks. We introduced an algorithm for ranking places based on their popularity among users, weighted by user expertise score in the query string. We built a location-based service based on the above ideas, using user interactions on Twitter and FourSquare. Our evaluation results comparing our approach to existing approaches, such as user-review based, expert-based and hybrid schemes, shows that our approach returns results that are atleast as relevant as current approaches at a substantially lower cost.

Future Work.

As future work, we are extending our location-based service to support on-demand queries that are forwarded to the mobile phones of other users. The responses are aggregated and returned in real-time, thus allowing a user to have a real-time view of the world using the mobile phones of other users. We have built a prototype of this extension, and plan to conduct a user study.

References

1. Tarek Abdelzaher, Yaw Anokwa, Peter Boda, Jeff Burke, Deborah Estrin, Leonidas Guibas, Aman Kansal, Samuel Madden, and Jim Reich. Mobiscopes for human spaces. Pervasive Computing, IEEE, Vol. 6, No. 2. (April 2007), pp. 20-29, 2007.
2. Anthony Bonato, Jeannette Janssen, and Pawel Pralat. A geometric model for on-line social networks. Proceedings of the 3rd Workshop on Online Social Networks (WOSN), 2010.

3. J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. Proceedings of World Sensor Web Workshop, ACM Sensys, 2006.
4. F Calabrese, F C Pereira, G Di Lorenzo, L Liu, and C Ratti. The geography of taste: Analyzing cell-phone mobility and social events. Proceedings of the 8th International Conference on Pervasive Computing, 2010.
5. Justin Cranshaw, Eran Toch, Jason I. Hong, Aniket Kittur, and Norman Sadeh. Bridging the gap between physical location and online social networks. Proceedings of the 12th ACM International Conference on Ubiquitous Computing, 2010.
6. Does Foursquare have a douchebag problem? http://techcrunch.com/2010/01/31/foursquare-douchebag/.
7. David Dearman and Khai N. Truong. Identifying the activities supported by locations with community-authored content. Proceedings of the 12th ACM international conference on Ubiquitous computing, 2010.
8. N Eagle and A Pentland. Reality mining: Sensing complex social systems. Personal Ubiq. Comp., pp. 255-268, 2006.
9. Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. Metrosense project: People-centric sensing at scale. Proc. of Workshop on World-Sensor-Web (WSW 2006), pp. 6-11, 2006.
10. Facebook Places. http://www.facebook.com/places/.
11. Fouesquare Blog: 200 millionth check-in. http://blog.foursquare.com/2010/10/06/quite-the-way-to-celebrate-our-200-millionth-check-in/.
12. Techcrunch: It took Foursquare only 2.5 months to double check-ins to 200 million. http://techcrunch.com/2010/10/06/foursquare-200-million/.
13. FourSquare. http://foursquare.com/.
14. Shravan Gaonkar, Jack Li, Romit Roy Choudhury, Landon Cox, and Al Schmidt. Micro-blog: Sharing and querying content through mobile phones and social participation. Proceedings of ACM MobiSys, 2008.
15. F Girardin, A Vaccari, A Gerber, A Biderman, and C Ratti. Towards estimating the presence of visitors from the aggregate mobile phone network activity they generate. Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management, 2009.
16. Google Geocoding API. http://code.google.com/apis/maps/index.html.
17. Gowalla. http://gowalla.com/.
18. Sibren Isaacman, Richard Becker, Ramon Caceres, Stephen Kobourov, James Rowland, and Alexander Varshavsky. A Tale of Two Cities. Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (HotMobile), 2010.
19. Jon Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (5): 604632, 1999.
20. Nicholas D. Lane, Dimitrios Lymberopoulos, Feng Zhao, and Andrew T. Campbell. Hapori: Context-based local search for mobile phones using community behavioral modeling and similarity. Proceedings of the 12th ACM International Conference on Ubiquitous Computing, 2010.
21. Jialiu Lin, Guang Xiang, Jason I. Hong, and Norman Sadeh. Modeling people's place naming preferences in location sharing. Proceedings of the 12th ACM International Conference on Ubiquitous Computing, 2010.
22. Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. Exploiting social context for review quality prediction. Proceedings of the 19th International World Wide Web Conference (WWW), 2010.

23. Amazon Mechanical Turk. http://aws.amazon.com/mturk/.
24. Emiliano Miluzzo, Nicholas D. Lane, Kristf Fodor, Ronald A. Peterson, Hong Lu, Mirco Musolesi, Shane. B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application. Proc. of ACM Conference on Embedded Networked Sensor Systems (SenSys), 2008.
25. M Motani, V Srinivasan, and P Nuggehalli. Peoplenet: Engineering a wireless virtual social network. Proceedings of ACM MobiCom, 2005.
26. Petteri Nurmi and Sourav Bhattacharya. Identifying meaningful places: The non-parametric way. Proceedings of the 6th International Conference on Pervasive Computing, 2008.
27. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Tech Report: Stanford InfoLab, 1999.
28. The Price of a Four-Star Rating. http://online.wsj.com/article_email/SB119162341176250617-lMyQjAxMDE3OTAxODYwMjgzWj.html.
29. F Rojas, K Kloeckl, and C Ratti. Dynamic city: Investigations into the sensing, analysis, and applications of real-time, location-based data. Proceedings of Harvard GSD Critical Digital Conference, 2008.
30. Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. Proceedings of the 19th International World Wide Web Conference (WWW), 2010.
31. Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Vito Latora. Distance matters: Geo-social metrics for online social networks. Proceedings of the 3rd Workshop on Online Social Networks (WOSN), 2010.
32. Local SEO best practises for Google, Bing and Yahoo. http://www.davidmihm.com/local-search-ranking-factors.shtml.
33. Ian Smith, Sunny Consolvo, Anthony Lamarca, Jeffrey Hightower, James Scott, Timothy Sohn, Giovanni Iachello, and Gregory D. Abowd. Social disclosure of place: From location technology to communication practice. Proceedings of the 3rd International Conference on Pervasive Computing, 2005.
34. Karen P. Tang, Jialiu Lin, Jason I. Hong, Daniel P. Siewiorek, and Norman Sadeh. Rethinking location sharing: exploring the implications of social-driven vs. purpose-driven location sharing. Proceedings of the 12th ACM International Conference on Ubiquitous Computing, 2010.
35. Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Sivan Toledo, Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. Proc. of ACM Conference on Embedded Networked Sensor Systems (SenSys), 2009.
36. Twitter Places: More context for your tweets. http://blog.twitter.com/2010/06/twitter-places-more-context-for-your.html.
37. Luis von Ahn. Games with a purpose. IEEE Computer Magazine, pp 96-98., 2006.
38. Tingxin Yan, Vikas Kumar, and Deepak Ganesan. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. Proceedings of MobiSys, 2010.
39. Yelp - user reviews and recommendations. http://www.yelp.com.
40. Yelp hit with class action lawsuit for running an extortion scheme. http://techcrunch.com/2010/02/24/yelp-class-action-lawsuit/.
41. Yelp user faces lawsuit over negative review. http://news.cnet.com/8301-1023_3-10133466-93.html.
42. Zagat: Ratings and reviews. http://www.zagat.com.

43. Zagat top spots lists. http://www.zagat.com/Verticals/BrowseTopLists.aspx?VID=8&SNP=ZATLR.
44. Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. Proceedings of the 19th International World Wide Web Conference (WWW), 2010.