# Query Processing in Location-Based Social Networks

Ammar Sohail * and David Taniar
Faculty of Information Technology
Monash University, Australia
ammar.sohail@monash.edu, david.taniar@monash.edu

Andreas Züfle
Dept. of Geography and GeoInformation Science
George Mason University, USA
azufle@gmu.edu

Park Jeong-ho
College of Information Technology
Gachon University, South Korea
jong882391@gmail.com

## ABSTRACT

The widespread proliferation of location-acquisition techniques and *GPS*-embedded mobile devices resulted in the generation of geo-tagged data at unprecedented scale and have essentially enhanced the user experience in location-based services associated with social networks. Such location-based social networks allow people to record and share their location and are a rich source of information which can be exploited to study people's various attributes and characteristics and can provide various Geo-Social (GS) services. In this demonstration, we propose a new type of query called *Top-k Geo-Social (TGS)* query, which enriches the semantics of the conventional spatial query by introducing a social relevance attribute. Similarly, users formulate their *TGS* queries and view the results through browser-based client side interface. On the server side, three approaches namely, 1) Social-First 2) Spatial-First and 3) Hybrid are proposed to efficiently process *TGS* queries.

## Keywords

Social Networks; Geosocial Networking; Graph database; Spatio-Social Applications; Geosocial query; Geographic information systems

## 1. INTRODUCTION

The increasing use of mobile devices, location-based services and recent advancements in location-acquisition technologies such as GPS and Web 2.0, have made location information an essential part of social networks. The fusion of social and geographical information has given rise to the notion of online social media known as location-based social networks (LBSNs) such as *Facebook, Foursquare.* Besides, location and social data featured rich sources of information which can be exploited to offer many services e.g., traffic updates, disaster management and finding nearby friends.

*Corresponding Author


*WWW 2017 Companion,* April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3051145

Thesedays, *LBSNs* are not just about friendship, these are way more complex graph-based structures containing various types of entities such as users, places, pages and relationships such as friendship, born-in, likes, visited, and works. Furthermore, some types of entities and relationships are associated with spatial information. To retrieve valuable information from LBSN graphs, various types of queries have been studied [1, 2, 3]. In this work, to the best of our knowledge,we are the first to study *Top-k Geo-Social (TGS)* query that retrieves a list of $k$ point of interests (POIs) ranked according to spatial and social relevance [4]. Specifically, *TGS* query finds the top-k POIs that are popular in a friends' circle in a given region based on their check-ins. To process *TGS* query, three approaches are proposed: 1) Social-First 2) Spatial-First and 3) Hybrid.

This demonstration enables participants to view the actual output of the *TGS* query containing the *top-k* POIs checked-in by friends in given region. In addition to this, other information related to top-k POIs such as name, address, visitors count, friends detail can also be viewed using Google Maps in browser-based interface. On the server side, we process the TGS query to efficiently retrieve the top-k POIs. Further, communication between server and client is handled using standard HTTP post operations.

## 2. FRAMEWORK AND QUERY

Before presenting our framework, we define some terminologies and *TGS* query.

***Location Based Social Network (LBSN):*** A *location-based social network* is an extension of a social network that includes location dimension as well. It can be represented by a graph $G = (U, E_{\text{relation}}, E_{\text{checkIn}}, P)$ where $U$ is a set of users, $E_{\text{relation}} \subseteq U \times U$ is a set of social connections between two users, $P$ is a set of points of interest (POI) and $E_{\text{checkIn}} \subseteq U \times P$ is a set of check-ins made by users to the POIs. Note that any edge $e \in E_{\text{checkIn}}$ is called a "Check-In".

***TGS Query:*** Given an LBSN graph $G = (U, E_{\text{relation}}, E_{\text{checkIn}}, P)$ , a query user $q \in U$, a query range $r$, and integer $k$, a TGS query $TGS(u, k, r)$ retrieves the top-k POIs, according to the number of check-ins made by friends of $u$, having a distance of at most $r$ to $u$.

Intuitively, we propose three approaches to answer *TGS* query: 1) Social-First 2) Spatial-First and 3) Hybrid. The *Social-First* approach first processes the social component (e.g. friendship relations and their check-ins) and then processes the spatial component (e.g. places in given range)

where as, *Spatial-First* initially processes the spatial component followed by processing the social component. In contrast, the *Hybrid approach* is capable of processing both social and spatial components simultaneously to answer such queries. Our framework prototype adopts client-server model in which users submit their queries through a web browser (client), and queries are then sent to the server. For each *TGS* query, the *top-k* POIs are computed and are sent back to the user. Figure 1 illustrates the architecture of our framework.

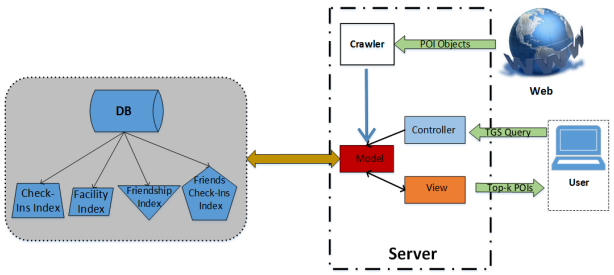



**Figure 1: Framework Architecture**

## 2.1 Client Side

The client side provides users with a mechanism to interact with the server side through a browser-based interface for submitting queries and to view the retrieved POIs. The client side component provides interaction with the map through Google Maps API. When a *TGS* query is submitted, the users specify their *user id*, a location, a range (radius) and number of desired POIs *k*. Consequently, queries are sent to the server by *HTTP* post operation. After a query is processed at the server side, the retrieved top-k POIs are returned and displayed using Google Maps in the client side browser.

## 2.2 Server Side

### *2.2.1 Overview:*

The web server is built using JSP and Apache Tomcat by applying MVC (model, view, controller) architecture. The MVC is divided into business logic and view logic. When a query is received by JSP server (Controller), it is forwarded to *Model* where the query processing algorithm implemented in Java is invoked to retrieved the result set which is then, forwarded to *View* to be sent to the client side browser.

### *2.2.2 Query Processing:*

On server side, the *TGS* query can be answered by employing any of the three proposed approaches. Simply stated, *Social-First* approach first retrieves social information (set of friends) of the query $q$ and then for each friend, the score of all checked-in POIs (spatial component) in given range is computed and updated where as, *Spatial-First* approach starts with the processing of spatial component of the query $q$ by retrieving all POIs in given range regardless of the fact whether these are checked-in by any friend of $q$ or not. Moreover, it then computes the social score of each POI by computing the number of friends checked-in to it by performing an intersection of the set of $q$'s friends and the set of visitors of the POI. In addition, *Hybrid* approach is capable of computing the score of each POI in range by processing social and spatial components of query $q$ together.

In other words, It prunes the friends (which have not visited the top-k POIs) and the POIs in given range (which cannot be the candidate for the top-k) simultaneously. Specifically, the algorithm indexes each user's *check-in summary* using well-known spatial data structure i.e., *R-tree* [5] along with storing each user's friends' *check-in* summary in a separate index structure. Similarly, grid partitioning approach is employed to divide the region formed by given range $r$ into small cells which facilitates in identifying those cells (including the POIs lie inside) that do not overlap with the objects of *friend's check-in summary R-tree* hence are pruned along with the objects (i.e., friends) which do not overlap with cells without further processing.

# 3. DEMONSTRATION DETAILS

In this demonstration, participants will be able to experience how the system can be used for issuing *TGS* queries to retrieve *Top-k* POIs in friends circle. The client-side (browser) interfaces are shown in Figure 2 and 3.

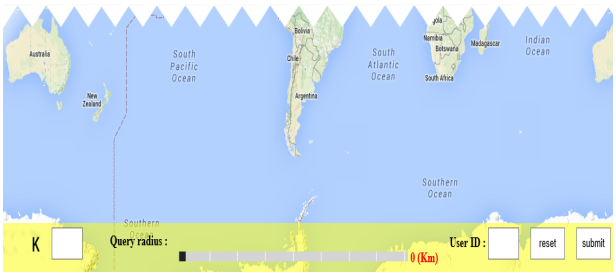
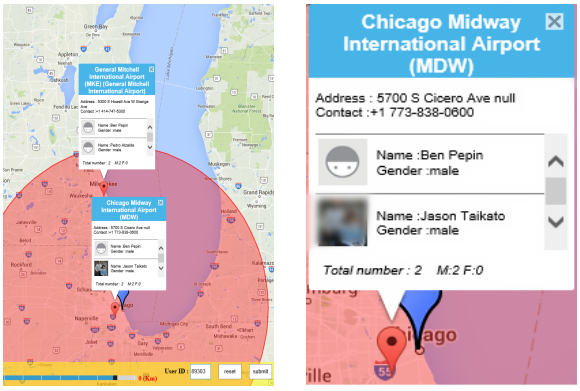



**Figure 2: Main Interface**

## 3.1 Submitting a TGS Query:

Initially, users specify a query User id. Using the *Foursquare API*, the framework focuses on that users hometown. Subsequently, users specify their location by clicking on Google Maps (the latitude and longitude of the selected location is obtained using Google Maps API), radius of the query region using the slide bar and number of desired POIs they are interested in to find.




(a) Query Result (b) POI Detail

**Figure 3: $T_kFP$ Query Result**

## 3.2 Dataset:

We use real-world foursquare [6] data set for the demonstration. The data set contains 3473835 friendship relations, 33,278,683 check-ins, 266,909 users and 3,680,126 POIs. Each POI is represented by a unique id and has longitude and latitude information associated with it. The POI's id is used to retrieved additional information such as POI name and address. Similarly, each check-in entry has user id, longitude and latitude information of checked-in POI and time stamp at which it was issued.

# 4. REFERENCES


[1] Ritesh Ahuja, Nikos Armenatzoglou, Dimitris Papadias, George J. Fakas, *Geo-Social Keyword Search*, Advances in Spatial and Temporal Databases, SSTD 2015, Hong Kong, China.

[2] Nikos Armenatzoglou, Ritesh Ahuja, Dimitris Papadias, *Geo-Social Ranking: functions and query processing*, VLDB J., 2015.

[3] Emrich, Tobias, Franzke, Maximilian, Mamoulis, Nikos, Renz, Matthias, Züfle, Andreas, *Geo-Social Skyline Queries*, Database Systems for Advanced Applications, Springer, Pages 77–91, 2014.

[4] Ammar Sohail, Ghulam Murtaza, David Taniar, *Retrieving Top-k Famous Places in Location-Based Social Networks*, Databases Theory and Applications - 27th Australasian Database Conference, ADC 2016, Sydney, NSW, September 28-29, 2016, Proceedings, 2016.

[5] Guttman, Antonin, *R-trees: a dynamic index structure for spatial searching*, ACM, Volume 14, 1984.

[6] Yang, Dingqi, Zhang, Daqing, Qu, Bingqing, *Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks*, ACM Transactions on Intelligent Systems and Technology (TIST), 2016.