Turkish Journal of Computer and Mathematics Education (TURCOMAT) ISSN: 3048-4855

Vol.8No.1(2017),218-223
DOI: https://doi.org/10.61841/turcomat.v8i1.14974

Identifying the Top-k Local Users in Geo-tagged Social Media Information

[1]AMARAVATHI PENTAGANTI,
RESEARCH SCHOLAR, DEPT OF CSE,
NIILM UNIVERSITY, KAITHAL-HARYANA.

[2]Dr. SONAKSHI KHURANA,
PROFESSOR, DEPT. OF CSE, NIILM UNIVERSITY,
KAITHAL-HARYANA.

Abstract

The use of location-based services and social media platforms is growing in popularity. When put together, they form location-based social media, in which users are linked not only by their internet acquaintances but also by their physical whereabouts. Because of this duality, new methods of querying and using social media data are made feasible. We describe a new and practical issue in the form of geo-tagged tweets: top-k local user search (or TkLUS for short). In order to discover the top-k people who have tweeted about a location within a certain distance from a given location (q), the TkLUS query takes a collection of keywords (W) and a location (q) as inputs. Many application situations may benefit from TkLus enquiries, including spatal decision-making, buddy suggestion, and many more. In order to effectively answer such requests, we develop a set of methods. To begin, we provide two approaches to local user ranking that combine text relevance with geographical proximity in a TkLUS query. After that, we build a hybrid index using scalable tweets. On top of that, we come up with two techniques to handle TkLUS requests. Lastly, in order to test the suggested methods, we do an experiential research using actual twitter datasets. Our ideas are successful, efficient, and scalable, as shown by the experimental findings.

1. Introduction

Twitter, Facebook, and other social media platforms are producing massive volumes of social media data. As an example, the Twitter blog reports that during the 2011 Super Bowl, the most tweets per second (TPS) ever recorded for a sporting event were 4,064 sent by Twitter users in a single second1. The most common reasons individuals use social media are to stay in contact with friends and family and to get information that is socially relevant.

Mobile location-based services (LBS) and GPS-enabled mobile terminals have recently proliferated, allowing social media data to obtain geo-location information. Sharing thoughts and ideas, receiving news, and comprehending conditions around real-world occurrences are all greatly facilitated by geo-tagged microblogs, such as tweets with geo-locations in metadata. In the wake of the 2010 Haitian earthquake, for instance, local organisations were swiftly established on Facebook. Consequently, there is an abundance of geo-related information available via so-called location-based social networks.

The foundation of search engines, traditional information retrieval (IR) methods, place an emphasis on sifting through lengthy, keyword-rich texts to extract important information. Searches using small amounts of social media data defined by few keywords are not well-suited to these. Another feature that Twitter provides is a real-time search service2. This service uses user-inputted keywords to provide highly-ranked tweets. Having said that, the search service does not deal with the geographical issue.

Social media data that has been geotagged opens up new possibilities beyond search engines. A family planning to relocate to Seoul, for instance, would enquire, "Are there any reliable babysitters in Seoul?" This search includes the terms "babysitter" and "Seoul" in its parameters. Such location-dependent and contextualised questions about travel, restaurants, and local services in everyday life are among the most popular categories, according to the social search engine Aardvark [1].

However, tweets and other social media data are intrinsically noisy since most of the time they don't include much important information in a little amount of bytes and don't appeal to a wide audience. Therefore, for location-dependent searches, it may not be enough to directly get tweets. Finding local social media individuals knowledgeable with the relevant concerns in a specific geographical area should be the primary focus of such a location-dependent inquiry, according to our argument. People in need may immediately contact with local Twitter users who are recommended for certain questions, which is a highly valuable feature of Twitter.

Here, we take a look at geo-tagged tweets—those that include location data—to see how we might find the top k local users. As Twitter generates a massive amount of data daily, previous research on microblog indexing and search has concentrated on a real-time architecture. In comparison, the percentage of tweets that include geotags is less than 1%. So, we think it's appropriate to analyse the geo-tagged tweets in batch mode independently. We can now parse the geo-tagged tweets offline after collecting them periodically. Compared to Twitter's current real-time indexing and search capabilities, our study stands out in this context.

Problem Formulation

The issue of TkLUS queries is defined here.

Predicting Results from Social Media

We isolate three distinct but connected ideas within the framework of social media. Specifically, social media is comprised of several postings made by individuals who, by their interactions within these posts, establish a social network.

First, a social media post definition. For every social media post, there is a 4-tuple p = (uid, t, l, W) that contains the following information: the post's date (t), the location (l), and the set of words (w1, w2,..., wn) that represent the post's text.

Depending on whether a user's social media network supports localisation, the location field could not be present for a lot of postings. For instance, if a user's smartphone is GPS-enabled, they have the option to record their location in tweets or not. We use social media postings with non-empty location data to obtain relevant results for user searches in this article.

2. Literature Survey

This research, conducted by Backstrom, Sun, and Marlow (2010) [1], examines the ways in which social networks and physical proximity affect the capacity to forecast users' geographic positions. The authors increase the accuracy of spatial prediction models by integrating data on social connections and geographic distance. The method emphasises the value of geographical and social proximity data in determining users' local presence, which is helpful in determining the top k local users.

In Cheng, Z., Caverlee, J., & Lee, K. (2010) [2], the authors provide a technique that may be used to determine a user's position without the need for explicit location identifiers by analysing the content of their tweets. It makes use of the content's geographic references and linguistic models. It offers a basic content-based approach to user location prediction, which is very pertinent to user activity-based user ranking.

In Hecht, B., & Gergle, D. (2010) [3], the amount of regionally relevant material uploaded online is the main focus of this study that explores the spatial component of user-generated content. The authors demonstrate how users often upload material about their local area. Recognising local users and setting them apart from contributors from across the world requires an understanding of "localness" in user-generated material.

The authors of Liu, Y., Kliman-Silver, C., & Mislove, A. (2014) [4] investigate how Twitter users' behaviour changes over time, focussing on how social and geographic variables affect their interactions and activities. Researchers may adjust their techniques for identifying local users as their social connections and places change, thanks to the insights this study offers into how user behaviour might alter over time.

In this research, Jurgens, D. (2013) [5], the spatial closeness of a user's friends is used to predict the user's position based on their social network. The methodology is applicable to the problem of discovering top local users based on their social circle and emphasises the important role that social connections play in locating users.

In Sadilek, A., Kautz, H., & Bigham, J.P. (2012)[6], the authors provide a real-time system that uses social media platform movement patterns and friend interactions to monitor and anticipate user whereabouts. This approach's real-time nature enables dynamic user identification in the area and real-time mobility monitoring.

Davison, B.D., and Hong, L. (2010)[7] In order to identify the fundamental subjects that people are talking about, this research uses topic modelling on Twitter data. The writers may deduce the interests and possible locations of users by comprehending the themes. To help identify significant local users, topic modelling may be used to cluster people based on their interests and inferred locations.

In their study of the relationship between friendships and mobility in location-based social networks, Cho, Myers, and Leskovec (2011)[8] demonstrate that friends are more likely to live near to one another. The results support the notion that movement patterns and social relationships play a crucial role in identifying local users.

In this research, Scellato, Noulas, Lambiotte, and Mascolo (2011)[9] investigate the socio-spatial characteristics of users in location-based social networks. It draws attention to the link that exists between users' geographical locations and their social networks. Comprehending these socio-spatial patterns facilitates the ranking of local users according to their geographic activity and social connections.

Faloutsos, C., Cui, P., and Jiang, M. (2014)[10], The authors emphasise on how connection patterns might show outliers or non-local users in their model for identifying aberrant behaviours in social networks. When attempting to rank local users, this strategy works well for removing anomalous or irrelevant individuals.

Singer, Y., and Crammer, K. (2003)[11], This seminal study addresses several online ranking algorithms that may be used to user ranking in an ever-expanding data stream. These algorithms may be modified to rank local individuals in real-time according to their behaviour in geotagged social media data.

Hu, Y., Monroy-Hernandez, A., & Farnham, S.D. (2013)[12], Whoo.ly is a system that helps people locate relevant material depending on location by organising and extracting hyperlocal information from social media. Users who are very active in certain geographic areas may be easily identified and ranked because to the system's emphasis on hyperlocal communities.

In this study, Sadilek, A., Krumm, J., & Horvitz, E. (2013)[13] investigate crowdsourcing via social media, in which users participate in geographically-based physical activities. The research provides real-world context-specific insights on user behaviour that may be used to identify nearby people in geotagged datasets.

Gao, H., Tang, J., Hu, X., & Liu, H (2012)[14],. Through the analysis of temporal patterns in user activity, the authors explore how time influences user behaviour in location-based social networks. By taking into account the times when local users are most active, the results aid in the discovery of these people.

Chang, K.C.-C., Wang, S., and Li, R. (2012)[15], This study suggests a technique for creating user profiles that take into account various regions deduced from their interactions and activities on social media. Accurately identifying and evaluating users who could be active in different local settings requires the usage of the multi-location profiling approach.

Zhang, M., and Hossain, M.A. (2013)[16], The privacy issues surrounding location-based social networks are the major topic of this paper, which especially addresses the difficulties in protecting user privacy while analysing location data. Designing systems that recognise local users while protecting their privacy requires an understanding of privacy issues.

In order to identify local events, Lee, R., & Sumiya, K. (2010)[17] examine the regional regularities in social media activity. Clusters of people taking part in the same events may be found using these regularities. Users may be identified and ranked according to their local activity by finding regional regularities in their behaviour.

Hossain, M.D., Cheok, A.D., and Lim, Y.-S. (2013)[18], This paper contributes to a better understanding of user interactions by examining how social and physical closeness affect the strength of linkages in geo-social networks. Based on the strength of their social and geographic ties, the tie strength estimate approach may be used to identify significant local users.

The Livehoods project, led by Cranshaw, J., Schwartz, R., Hong, J.I., & Sadeh, N. (2012)[19], uses geotagged social media data analysis to provide information on the dynamics of urban neighbourhoods and the activities of local users. Using user behaviour in certain neighbourhoods as a cluster, the initiative assists in identifying top-k local users.

In Davis Jr., C.A., Pappa, G.L., de Oliveira, D.R.R., & de L Arcanjo, F. (2011)[20], a model for estimating tweet locations based on a user's social network and affiliations is presented. By using their social ties, the relational inference model enhances local user identification.

The paper, which addresses a frequent problem in social media platforms, suggests techniques to infer user locations from sparse and noisy geo-tagged data (Jurgens, D., 2013[21]). When trying to find local users in big datasets, this method is essential for handling missing data.

According to Liu, J., Yin, H., & Xu, F. (2013)[22], the authors suggest a collaborative filtering framework that helps users choose places to go based on their interests and past activities. It is possible to modify the recommendation algorithm to find users who are active or prominent in certain regions.

Zheng, V.W., Cao, B., Zheng, Y., Xie, X., & Yang, Q (2010)[23],. In this research, we propose a collaborative filtering strategy for mobile recommendations that takes user context—including location—into consideration. Specifically, the system uses user location to generate a model that ranks nearby users according to their mobility patterns.

Ghosh, R., and Lerman, K (2011)[24], Through an analysis of user interactions and engagement levels, this study investigates ways to forecast prominent people in social networks. By examining their local reach and impact, the approach assists in identifying the top k influential users within a given geographic setting.

Sadilek, A., Silenzio, V., and Kautz, H. (2012)[25], The authors demonstrate how location data may provide important insights into user movements and behaviour by using geotagged social media to forecast patterns of disease transmission. Through spatial trajectory analysis, the movement pattern monitoring model may be modified to identify local users.

Cheng, Z., Sui, D.Z., Lee, K., & Caverlee, J. (2011)[26], This study looks at how people utilise location-sharing services and shares and engages with geotagged content. The results provide a broad perspective on user behaviour that may be used to assign people a score according to their local activities.

3. Research Methodologies

Questions like "What are the happenings in New York City?" might be posed in relation to geo-tagged social media data. where can I locate a babysitter in Los Angeles? There may be local individuals among U's social media users with the specified knowledge to answer such questions. In order to keep the questions as wide as possible, we have formalised them as the following issue. Issue Clarification. (Social Media's Top-k Local User Search) In the context of geo-tagged social media data = (P, U, G), a top-k local user search (TkLUS) discovers a set of k users Ek ⊆ D.U that meets the provided requirements given a query q(l, r, W) where q.l is the query location, q.r is the distance value, and q.W is the collection of keywords that capture user demands.

1. $\forall u \in \mathrm{E}_k$, $\exists p \in P_u$ such that $|q.l, p.l| \leq q.r$ ^{4} and $p.W \cap q.W /= \emptyset$.

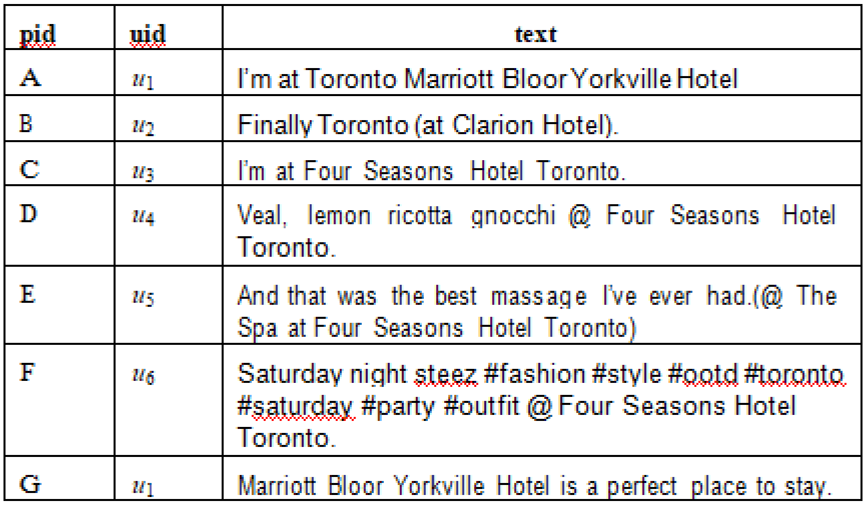
2. $\forall u \in \mathrm{E}_k$ and $\forall u' \in \mathbf{D}.U \setminus \mathrm{E}_k$, either u' does not satisfy condition 1 or $score(u', q) \leq score(u, q)$.

In this case, the relevance of a user to a search query is measured by the score function score(.). Section C.3 will go into depth about our scoring functions, but they take distance and keywords into account.

We provide a simple illustration. In Figure C.1, we can see a map showing the results of a Tk-LUS query at the coordinates (43.6839128037,-79.37356590) using the single keyword "hotel" and a 10-kilometer radius. Table C.1 lists all the tweets that include the word "hotel" along with the individuals that tweeted them. The map shows where these users are located. We will return user u1 if they have more tweets or user u5 if they have more replies/forwards (not visible in the table) based on separate user scoring algorithms.

A non-trivial challenge is to identify the top-k users for a TkLUS query. On many social media sites, such as Twitter, the user set U and the post set P are rather big. Irrelevantly checking the sets is undoubtedly inefficient. Determining how relevant a user is to a certain question is not an easy task. We also need to prioritise the relevant tweets and local users that match a TkLUS query so that we can return the most relevant persons in the query result. Such technological difficulties are methodically addressed in this study.

Table 1: Detailed Information of Example Tweets



pid	uid	text
A	u_1	I'm at Toronto Marriott Bloor Yorkville Hotel
B	u_2	Finally Toronto (at Clarion Hotel).
C	u_3	I'm at Four Seasons Hotel Toronto.
D	u_4	Veal, lemon ricotta gnocchi @ Four Seasons Hotel Toronto.
E	u_5	And that was the best massage I've ever had.(@ The Spa at Four Seasons Hotel Toronto)
F	u_6	Saturday night steez #fashion #style #ootd #toronto #saturday #party #outfit @ Four Seasons Hotel Toronto.
G	u_1	Marriott Bloor Yorkville Hotel is a perfect place to stay.

Tweet Thread and Tweet Popularity

Social media users are more inclined to engage with the tweets of a local user who provides valuable information. Thus, we begin by thinking about how popular a tweet p is in relation to a query q(l, r, W). Without any q.W. keywords, p seems uninteresting at first glance. If a tweet contains a query term or keywords, we will only evaluate it and give it greater weight. Conversely, if a tweet generates noticeable reactions on social media, it is likely to be popular. Take Twitter as an example: when users provide helpful information, their tweets usually get a lot of attention and engagement. Similarly, we rank tweets according to the number of replies they received. We provide the notion of a tweet thread to bolster these ideas.

A tweet thread may be queried using the format q(l, r, W). The tweet tree is denoted by T, where

1. 1. a distinct tweet is associated with every node;
2. A tweet p with keywords in q.W. is the root of T.root;
3. Pj responds to or forwards tweet pi if pi is the parent node of pj.

You may see an example in Figure 1. Query q(l, r, W) includes the terms sought for in tweet p1. Each of the subsequent tweets (p2, p3, and p4) responds to or forwards the tweet (p1) in some way.

Figure 1 shows an example. Tweet p_1 has keywords requested in query $q(l, r, W)$. Tweets p_2, p_3, and p_4 reply to or forward p_1, each having further reply or forward tweets.

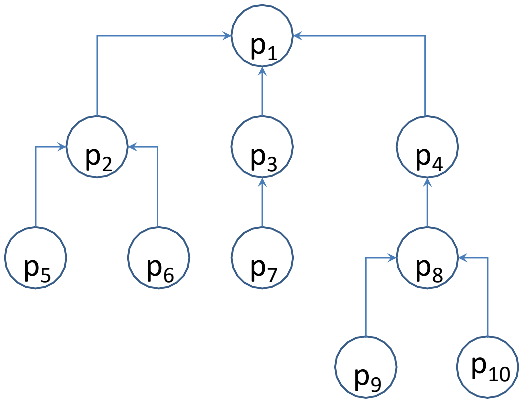


Fig. 1: Example of a Tweet Thread

It seems to reason that a root tweet would be more popular if it were part of a larger thread of tweets. A tweet will stand alone as a thread if it has no connections to other tweets. It seems like we should give this kind of tweet a low score as it isn't popular. In addition, a higher number of tweets in a thread indicates that the root tweet is leading a discourse with many issues, making the thread more meaningful than one with fewer tweets. Consequently, the local user who tweets the root tweet is considered relevant for answering the inquiry. In Section 4, the method for building the thread of tweets and calculating its score will be detailed.

Following the idea of a tweet thread, we say that a tweet p is popular if its popularity score is the same as that of the thread T whose root is p.

Individual The Score of Tweet

In order to get the score of a single tweet p relative to a certain query q(l, r, W), we must consider both the popularity of p relative to q and the distance between p and the query. The distance score δ(p, q) of a tweet is defined in the following way for the former.

Keyword Relevance Score of Tweet

$$\rho(p, q) = \frac{|q.W \cap p.W|}{N} \cdot \varphi(p)$$

The product of the normalized instances of a query term in tweet p, its popularity χ(p), and the product is used as the relevance of p's keywords in this definition. In this case, N is a normalised parameter and additional normalization of χ(p) is not required since ρ(p, q) might be greater than 1. We found that when N is experimentally set around 40, the distance score is equivalent to the keyword relevance score. Furthermore, a bag model of keywords is really used to tally the occurrences. To be more specific, p.W is a bag/multiset and q.W is a set. When a query comprises the words "spicy restaurant" and two words that contain "restaurant," the number of times these words appear in tweet p is 3.

4. Data Organization and Indexing and Architecture and Data Organization

In Figure 2, we can see the system architecture. Crawling Twitter for data in JSON format is a typical use case for the Twitter Rest API. All of the information associated with tweets is saved in a central database after the ETL process. We think it's appropriate to process the geo-tagged tweets independently in batch mode rather of using the current real-time systems that handle general tweets, as they only make up a small fraction of the total tweets dataset. In particular, we might gather the geographical tweets on a regular basis (say, once a day) and then construct an index for them. The Hadoop distributed file system (HDFS) stores a scalable index that includes both geographical and textual information, created using the Hadoop MapReduce algorithm. Additionally, the text and content of tweets are saved in HDFS. Section 4 will provide specifics on the hybrid index. Section 5 will go into depth about how the database and index are used to construct the query processing algorithms.

Figure 2 shows the centralised metadata database where all tweets in our system are saved following the schema of (sid, uid, lat, lon, ruid, rsid). Here is an explanation of what each characteristic means.

The tweet timestamp, or "sid" attribute, is basically just a unique identifier for the tweet. In this tweet, the attribute "uid" shows the user ID. Location data for this tweet is included in the "lat" and "lon" attributes. Both the "ruid" and "rsid" attributes are used to indicate the user IDs of the tweets that are responded to or forwarded by this particular tweet. In addition, a B+-tree is constructed using the attribute "sid" as its principal key. The property "rsid" is the basis of yet another B+-tree. The execution of queries is sped up by making use of these indexes.

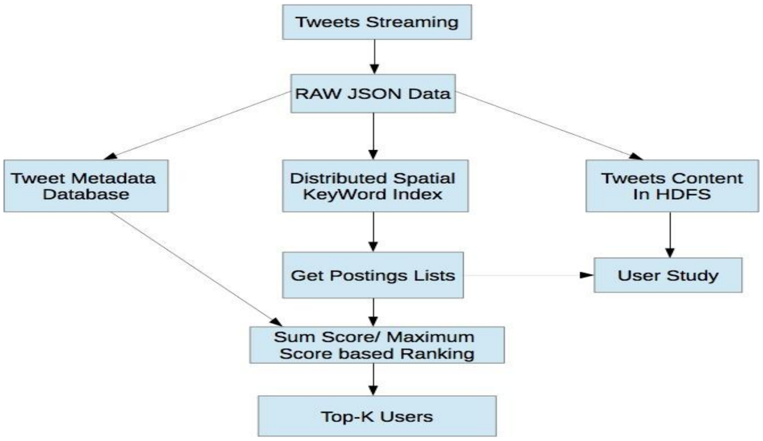
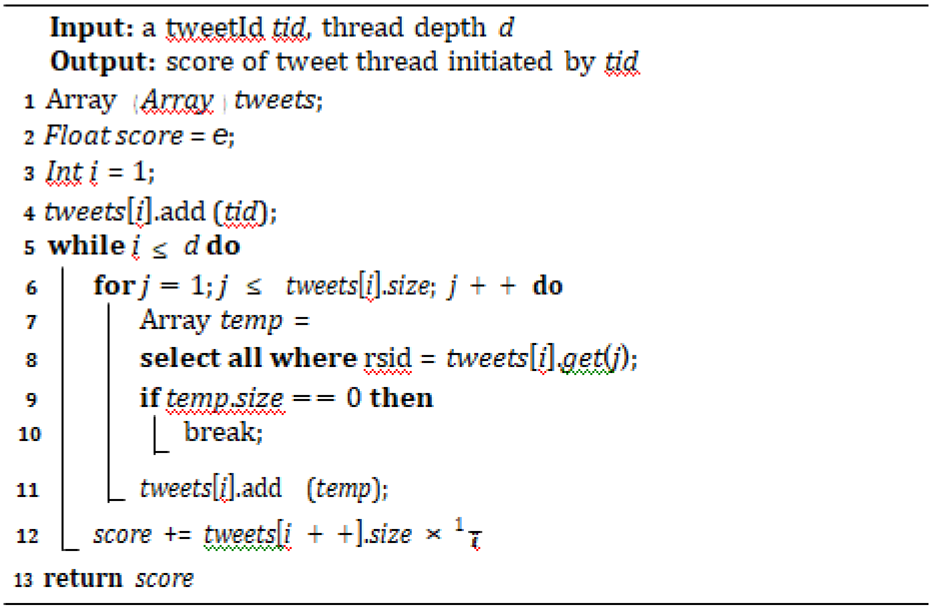


Fig 2: System Architecture

Algorithm 1 lays out the steps for building the tweet thread (discussed in Section 3) and calculating the thread score using this database design and Definition 5. I/Os occur in a SQL query on line 7. Because building a full tweet thread might involve a lot of I/Os, a thread depth d is always provided to restrict the creation process in a realistic implementation.

Algorithm 1: Construct tweet thread and compute its score



```
Input: a tweetId tid, thread depth d
Output: score of tweet thread initiated by tid
Array ⟨Array⟩ tweets;
Float score = e;
Int i = 1;
tweets[i].add (tid);
while $i \leq d$ do
    for $j = 1; j \leq$ tweets[i].size; $j++$ do
        Array temp =
        select all where rsid = tweets[i].get(j);
        if temp.size == 0 then
            break;
        tweets[i].add (temp);
    score += tweets[i++].size $\times \frac{1}{i}$
return score
```

Hybrid Index

The Geohash encoding, which is typically based on the quadtree index, is modified in our hybrid index design. An easily-maintained spatial index structure that uniformly divides the space is quadtree. Every node in a quadtree, unless it's a leaf node, represents a square with four children. In every node split, the parent node is divided in half along the horizontal and vertical axes to produce each quadrant.

Each leaf node in a full-height quadtree is a full geohash if we ignore the tree structure and encode each child by adding two bits to the parent encode. The bits for upper-left, upper-right, bottom-right, and bottom-left are 00, 10, 11, and 01, respectively. For example, if we want to encode a latitude/longitude combination (-23.994140625, -46.23046875) with a 20-bit precision, the encode will be "11001111111011010100," which is determined by the tree's height. The next step is to convert it to Base32 encode, which consists of 22 letters (a–z except a, i, l, and o) and 10 numbers (0–9). To get the final geohash, which is "6gxp," we'll encode a character every five bits. Given the foregoing, it is reasonable to assume that nearby points will share a prefix in order to index the geohash using a trie or prefix tree. An answer to a circle inquiry may be found by building a set of prefixes that minimises the area outside the query zone while completely covering the circle region. A common tool for building such a collection of prefixes is the Z-order curve. The data indexed by geohash will include all points for a particular rectangular region in contiguous slices, which is why we modify their encoding technique. All of the coordinates for a certain rectangle will be stored on a single computer in a distributed system that uses geohash indexing. With this benefit, query evaluation might save on I/O and communication costs.

Figure 3 shows the hybrid index. Part one is the forward index, while part two is the inverted index.

The forward index is structured with each item including a geohash code (gei) and a keyword (kwi). Our solution keeps the forward index in main memory as it is not particularly huge. Each item in the forward index is linked to a posts list (Pi) in the Hadoop HDFS inverted index. In the tweet database, ⟨gei, kwi⟩ pairings are linked to tweet IDs according to the inverted index. To put it otherwise, the inverted index directs each pair ⟨gei, kwi⟩ to the tweets in the database that include kwi.

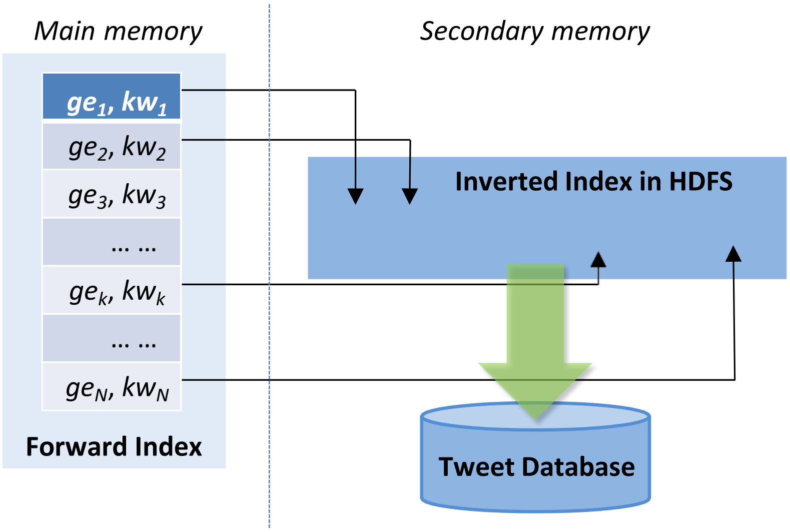


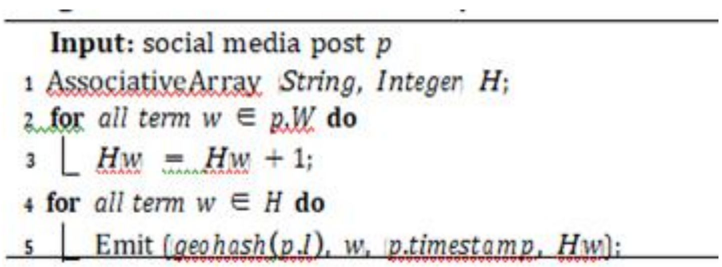
Fig 3. Index Structure

The inverted index key in our solution is a combination of a geohash code and a keyword. Because of their near-identical meaning in the academic literature, we use the terms "term" and "keyword" interchangeably throughout this article. A posts list has pairs TID and TF for each item. In particular, the term frequency (TF) is helpful for counting the occurrences of the query phrase in a tweet while processing a query, and the tweet ID (TID) is effectively the timestamp for the tweet. A prior suggestion [4] outlines the implementation of the forward index for retrieving the posts in query processing.

Hadoop MapReduce is a fault-tolerant and scalable programming model that we've decided to use since we're dealing with massive amounts of geo-tagged tweet data. We find that the geohash encoding approach works well with MapReduce's hybrid index, which takes into account both geographical and textual features.

The Hadoop MapReduce architecture is shown in Index creation Algorithms 2 and 3, which show the Map and Reduce phases of the index creation algorithm, respectively. A social media post, denoted as p in Definition 1, is fed into the mappers. Every post's content is tokenised and every phrase is stemmed in the map function. During the tokenisation process, stop words are filtered out. Each term's frequency is recorded in an associative array H. In the next step, the map function iterates over all terms: for each term, generate a posting by calculating the geohash using the post's geographical information. With the timestamp p.timestamp and the term frequency H w, the posting is a pair. Because every timestamp is distinct, we can see that p.timestamp matches the tweet ID here. In the end, the mapper publishes a key-value pair consisting of the posting and a pair of geohashes and terms. A fresh list P is created and initialised in the reduce phrase. Following that, the reduce function adds to the list all the posts that are linked to each pair of geohashes and terms. The tweet ID (timestamp) and the frequency (f) are the two variables that make up each message. Before they are sent out into the world, the posts are sorted according to the timestamp p.timestamp. In Section C.5, you'll find Algorithms 7 and 8, which describe how to do highly fast intersection operations on the sorted posts during query processing.

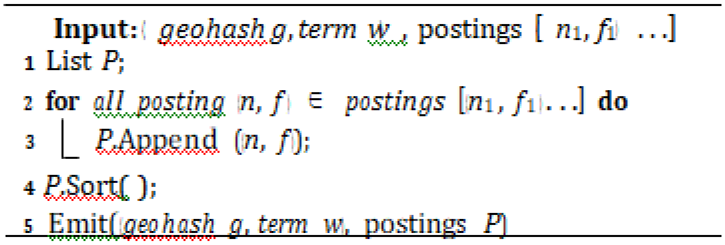
Algorithm 2: Pseudo-code of Map Function



```
Input: social media post p
AssociativeArray ⟨String, Integer⟩ H;
for all term w ∈ p.W do
    $H_w$ = $H_w$ + 1;
for all term w ∈ H do
    Emit(⟨geohash(p.l), w⟩, ⟨p.timestamp, $H_w$⟩);
```

A posting forward index is established to monitor the location of each posts list in HDFS, and another MapReduce job is performed over the inverted index files in HDFS to produce the forward index discussed before. Take note that the inverted index key may be guaranteed to be sorted using the Hadoop MapReduce architecture. This signifies that the word for the composite key geohash has been sorted. In order to ensure that nearby locations linked with the same keyword are kept in contiguous disc pages, it is likely that they will all have the same geohash prefix. You will save a lot of time by organising the posting lists that belong to nearby locations that use the same keyword. Also, we can simply expand our index building to petabyte- or terabyte-sized data sets using MapReduce.

Algorithm3 : Pseudo –code of the Reduce function



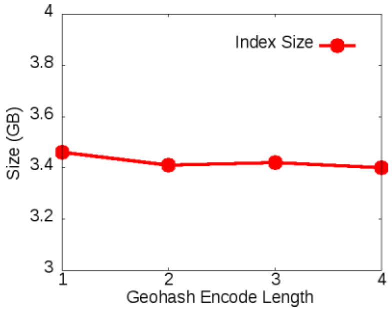
```
Input: ⟨geohash g, term w⟩, postings [⟨$n_1$, $f_1$⟩ ...]
List P;
for all posting ⟨n, f⟩ ∈ postings [⟨$n_1$, $f_1$⟩...] do
    P.Append (⟨n, f⟩);
P.Sort( );
Emit(⟨geohash g, term w⟩, postings P)
```

4. Experimental Study

We test the suggested methods for processing TkLUS questions in a comprehensive experimental research. We use geographical coordinates retrieved via Twitter's REST API to sample a real-world dataset. Nearly 514 million tweets with geotags were included in the 20.3GB dataset, which spans the months of September 2012 through February 2013. The pertinent experimental findings are detailed in this section.Time and Money Spent on Construction and Storage We begin by conducting tests to assess the index building process. We take measurements of the duration and size of the index during building while varying the Geohash settings. For further information, see Table 1. In a Hadoop MapReduce cluster consisting of three PCs, we build the suggested hybrid index. Intel(R) Core(TM) i7 Quad Core processors power all three of the computers. The setup calls for one computer to act as master and the others to play the role of slave.

As shown in Figure 4, the efficiency of index creation is unaffected by the Geohash setup. Around 850 minutes is the constant duration of index creation. In around 1,500 minutes, the state-of-the-art spatial keyword index I3 can analyse 15 million geo-tagged tweets using a single system with a Quad-Core AMD Opteron (tm) Processor 8356 and 64GB of RAM. Whereas the I3 index takes an order of magnitude longer to process tweets, our MapReduce index generation approach gets the job done in an order of magnitude less time. It should be noted that our index is constructed and maintained in a distributed manner, in contrast to others such as IR-tree versions and I3, which are centralised and incapable of handling large-scale data or solving TkLUS queries.

Figure 5 displays the outcomes of the hybrid index sizes. With a consistent 3.5 GB in HDFS, the index size remains constant regardless of the Geohash setup. Our indexes process an order of magnitude more tweets than I3, yet the size of the index is almost same.



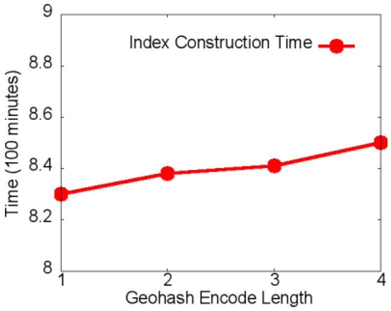


Fig. 4: Index Construction Time **Fig. 5:** Index Size

Assessment of the Processing of Queries

Here in the experiments, we still use HDFS to store the index and tweets, and we retain all of the tweet information in one place. Table 2 contains the cluster parameters.

Table 2: Cluster Summary

Node Type	**Memory**	**Hard Disk**	**CPU**
Master	8GB	220GB	Quad Core Intel(TM) i7
Slave	8GB	500GB	8 Core Intel(TM) i7
Slave	8GB	500GB	8Core Intel(TM) i7

Query Settings

We choose 30 relevant terms, including the 10 most common ones in Table C.3, using data set statistics. In the trials, one out of thirty results is assigned at random to a query with one keyword. The AOL query logs that include the single keyword from Table C.3 are used to generate queries with 2 and 3 keywords. We utilize AOL query records to see, for instance, the frequency with which the terms "restaurant seafood" and "morroccan restau- rants houston" appear. We use the geographical distribution of our data collection to randomly assign locations to each query. Lastly, a 90-query set was used in our studies, which consisted of randomly combined keywords and places. Thirty enquiries correspond to each question type inside the query collection, based on the amount of keywords: single, two, and three.

To ensure that the two variables are treated as having an equal influence, we set α to 0.5 in accordance with Equation 11. For objective test results, we enable both the HDFS cache and the database cache. Due to its short size (less than 12MB in our trials), the posts forward index is loaded into memory before query processing begins. In HDFS, disk-based random access to inverted index is used.

Table 3: Top-10 Frequent Keywords

Freq. Rank	**Keyword**		**Freq. Rank**	**Keyword**
1	restaurant		2	game

3	cafe		4	shop
5	hotel		6	club
7	coffee		8	film
9	pizza		10	mall

The experimental queries range from 1 to 3 with varying numbers of query keywords (q.W) and k, the number of local people to return, from 5 to 10. From 5 km to 20 km, we change the query radius for each setup. We fixed the value of e in Definition 4 to 0.1 in our implementation.

Effect of Geohash Encoding Length

The impact on the query processing is examined when we first change the Geohash setting from 1-length encoding to 4-length encoding. Table C.4 displays the geohash at various lengths; for instance, consider the coordinates (-23.994140625, -46.23046875).

Table 4: Geohash Encoding Length Example

length	**Geohash**	**length**	**geohash**
1	6	3	6gx
2	6g	4	6gxp

From 5 km to 20 km, we change the query radius for each setup. We use a random selection of 10 enquiries from the query set to generate each query radius. The average time it takes to execute a query is shown in Figure 6.

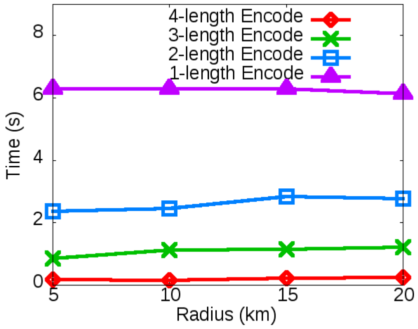


Fig. 6: Effect of Geohash Encoding Lengths

There will be fewer but bigger grid cells in a coarser grid if the Geohash encoding is of a shorter length. Since the close points linked with the same keyword are likely stored in contiguous disc pages, more I/Os are not necessarily caused by a longer encoding, even though it complicates the construction of a set of prefixes fully covering the query range, increasing the number of candidate grid cells in search. Using a lower length encoding results in bigger grid cells and requires processing more points per cell, but the I/O cost remains almost the same. Consequently, TkLUS query processing benefits from encodings of larger duration.

Findings from Searches Using a Single Keyword

Then, sticking to the one-word searches outlined in Section C.6.2, we change the query radius from 5 km to 100 km.

Figure C.8 displays the outcomes of our comparison of the two ranking methods—sum score based and maximum score based—with respect to the efficiency of query processing. Increasing the query range causes both techniques to take longer to process the query. Both approaches

work similarly, with the maximum score based ranking technique only marginally superior, for query ranges less than 20 km. The maxi- mum score based ranking approach obviously beats the sum score based ranking method for bigger query ranges. The pruning power of the maximum score based ranking approach becomes more apparent when dealing with big query ranges including more candidates, which is why it is preferable for these types of queries. Pruning has less impact on smaller query ranges since there are fewer tweets in them. We look at the query's link with the Kendall tau rank:

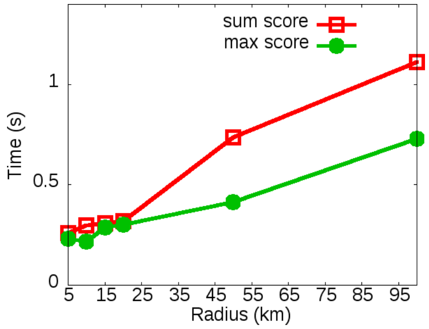


Fig. 7: Single Keyword Efficiency

As a consequence, we find the top-5 and top-10 results for a single keyword query and display the variation Kendall tau coefficient in Figure C.9. The Kendall tau coefficient is more than 0.863 in all of the tested situations. That the two rating systems are so compatible with one another is clear from the data.

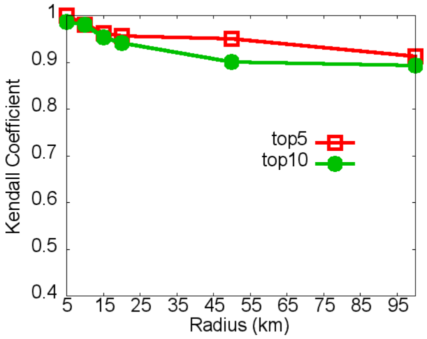


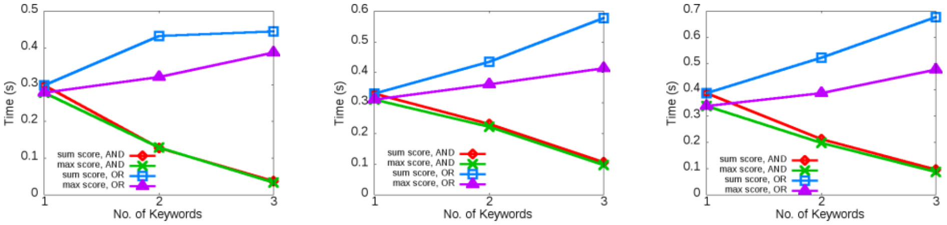
Fig.8 : Kendall Tau for Single Query Keyword

Findings from Searches that Involve Multiple Keywords

In this series of tests, we examine how query processing changes when we change the amount of keywords from one to three.

When processing a query that has numerous keywords, the OR or AND semantic might be utilized. The findings about the effectiveness of query processing are shown in Figure 8. The general rule is that in the OR semantic, the processing time of a query increases as the number of keywords increases, whereas in the AND semantic, the reverse is true. Since AND semantic eliminates more possibilities, the rationale is obvious.

When it comes to query ranges of 20 km and 50 km in particular, the maximum score based user ranking usually outperforms the total score based ranking. Due to the high number of candidates pruned by the intersection operation used to analyse the AND semantic, the pruning power for maximum score based ranking is severely limited. On the other hand, more candidates are generated and more space is available for pruning when the OR semantic is processed by the union operation.

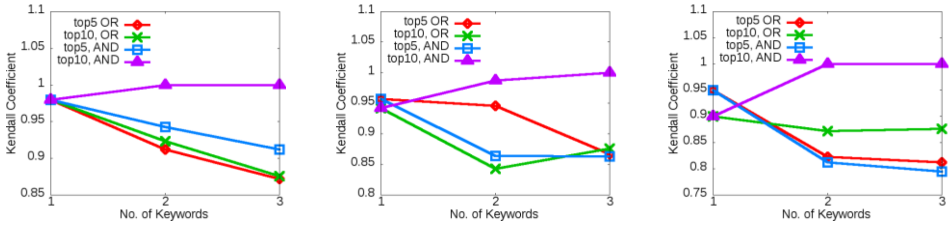


(a) 10 km query range (b) 20 km query range

(c) 50 km query range

Fig. 9: Efficiency of Queries Using Multiple Keywords

We also use the same procedure to estimate the Kendall tau coefficient in the same set of studies. Figure 8 displays the outcomes of the experiments. Regardless of the range of queries, the Kendall tau coefficient for the AND semantic is consistently more than 0.95. What this means is that the results returned by two user rating techniques are always somewhat similar. A Kendall tau coefficient little around 0.8 is considered minimal for the OR semantic. Both ways of rating remain unchanged.

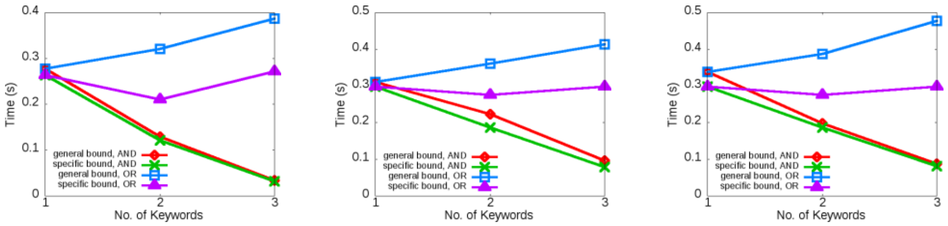


(a) 10 km query range (b) 20 km query range (c) 50
km query range

Fig. 10: Multi-Query Keyword Kendall Tau

The Influence of a Targeted Popularity Level on Maximum Score-Based User Ranking

Explains how the maximum score based query processing method uses hot keyword popularity to determine specified upper bounds for tweets. It is believed that searches containing those popular keywords would perform better with these pre-computed boundaries. Table 4 and Figure C indicate that we uphold the top limit of popularity for ten trending keywords. In Section, Definition 12, you can see the findings on the query improvements compared to using the generic upper bound. When dealing with several keywords, the "AND" semantic takes the upper limit that is the least among all of the query keywords, while the "OR" semantic takes the greatest. If you have a query that includes the word "Mexican restaurant" but the upper bound popularity of the word "restaurant" is higher, the "AND" semantic will use the "Mexican" upper bound popularity and the "OR" semantic will use the "restaurant" upper bound popularity.



(a) 10 km query range (b) 20 km query range (c)

50 km query range

Fig. 11: Effect of Specific Tweet Popularity Bound

Using such a precise popularity bound of trending keywords clearly speeds up the semantic and quantitative aspects of query processing. The speed improvement is most noticeable for larger query ranges. When the query processing algorithm calculates tweet threads, the particular popularity bound is useful for excluding irrelevant tweets due to those popular terms.
User Research

To test how well our scoring mechanisms work for the top k local users, we do a user research. Thirty queries including one to three keywords are randomly provided. Using five, ten, fifteen, and twenty km as our parameters, we get the top ten results for each query. Every line in the top ten results of a query is structured as a pair (userId, tweet content), with userId identifying a user and tweet content being the matching set of keywords discovered in the query. In order to ensure that the query results are relevant, we are inviting six people who are acquainted with Twitter to provide comments. If a participant believes a query result line is relevant to the inquiry, she or he will give it a 1; otherwise, a 0 In order to analyze each query result four times, we provide each participant a set of 20 top-10 results. If a user's tweets are deemed relevant twice or more, they will be considered relevant for that relevant inquiry on Twitter.

In order to measure efficacy in the user research, we use accuracy. Here, "precision" means how many local users were considered relevant by the user research out of all the returning users. As shown in Figure 12, we quantify the accuracy for the top-5 and top-10 results of the query.

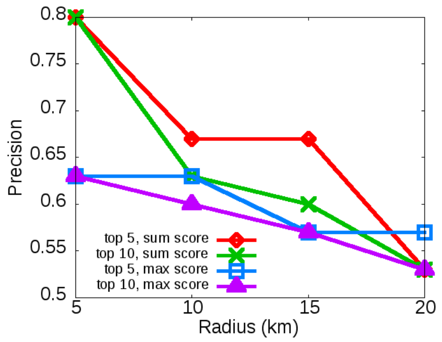


Fig.12 User Study Results

When it comes to moderate and short query ranges, both user ranking methods—sum score based and maximum score based—are quite successful. For query ranges less than 10 km, the accuracy is consistently between 60% and 80%. As the query range becomes larger, the accuracy drops down a little. These results provide credence to our distance score, which gives more weight to tweets sent from places that are geographically closer to the query location.

Conversely, the top-5 query results consistently outperform the top-10 results in terms of accuracy, reaching up to 80%. That our user ranking algorithms are able to give preference to local individuals who are more relevant to the inquiry is evident from this. Using Social Media Locations to Your Advantage

There is a growing interest in studying how to explore and analyze social media with spatial awareness. To find out about events as they happen in real time and where they are located, use algorithms to sift through twitter streams. Using geotagged tweets as a starting point, create and display topical spatiotemporal patterns.

The term "location-based social network" (LBSN) refers to a kind of online community where users are linked by the mutual reliance on one another's physical locations and the geotagged assets they provide, including images, videos, and text.

One example of a social search With Aardvark, users may pose questions and the algorithm will choose the persons best suited to answer them. Instead of finding the correct document, the goal of this social search engine is to locate the appropriate person to answer one's information needs. Since many of the questions are sensitive to their vicinity, geography plays a crucial role in this social engine's question routing process.

When applied to geo-tagged tweets, TkLUS presents a novel challenge, distinct from previous research. In contrast to previous works, this one introduces a hybrid index for tweets that is enabled by MapReduce, scores people and tweets using separate functions, and uses TkLUS query techniques.

5. Conclusion and Future Work

In geo-tagged social media, this study focuses on top-k local user search (TkLUS) queries. When a user inputs a location q, a distance r, and a set of needs-describing keywords W, a TkLUS query discovers the top-k local users who have tweeted about the set of keywords at a location within r from q. In order to handle these TkLUS enquiries, a collection of applicable methods is developed. To measure how popular tweets are, we offer the idea of a "tweet thread," which would allow us to rank people and tweets according to factors like geographical closeness and the relevance of their keywords. Distributed processing of large volumes of geo-tagged tweets is the goal of a hybrid index. Efficient algorithms and pruning limits are created to handle TkLUS requests. Experimental results on a large dataset of actual tweets with geo-tags are used to assess the suggested methods. Scalability, efficiency, and efficacy are highlighted by the experimental outcomes of our solutions.

Future study might go in a number of different areas. An expansion of the TkLUS query definition that takes time into account when ranking tweets and local users is possible. For instance, we can limit our search to tweets published during a certain time frame by defining a query for that window. While searching through all tweets is still an option, we can now prioritize the most recent tweets (and the persons who sent them) in the results.
There are tweets that reference a location or places but don't include longitude or latitude in the metadata. Research on how to harness the implicit location data in tweets like these to fulfil user requests like TkLUS enquiries should be prioritized.

Within the scope of this article, geo-tagged tweets are the predominant emphasis. To get more informative query results by integrating various social networks, it's also fascinating to make the search for local individuals beyond platform boundaries.

References

1. L. Backstrom, E. Sun, and C. Marlow, “Find me if you can: Improving geographical prediction with social and spatial proximity,” in Proc. 19th Int. Conf. World Wide Web (WWW), 2010, pp. 61–70.
2. Z. Cheng, J. Caverlee, and K. Lee, “You are where you tweet: A content-based approach to geo-locating Twitter users,” in Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2010, pp. 759–768.
3. B. Hecht and D. Gergle, “On the ‘localness’ of user-generated content,” in Proc. 2010 ACM Conf. Comput. Supported Coop. Work (CSCW), 2010, pp. 229–232.
4. Y. Liu, C. Kliman-Silver, and A. Mislove, “The tweets they are a-changin': Evolution of Twitter users and behavior,” in Proc. 8th Int. AAAI Conf. Weblogs Social Media (ICWSM), 2014, pp. 305–314.
5. D. Jurgens, “That’s what friends are for: Inferring location in online social media platforms based on social relationships,” in Proc. 7th Int. Conf. Weblogs Social Media (ICWSM), 2013, pp. 273–282.
6. A. Sadilek, H. Kautz, and J. P. Bigham, “Finding your friends and following them to where you are,” in Proc. 5th ACM Int. Conf. Web Search Data Mining (WSDM), 2012, pp. 723–732.
7. L. Hong and B. D. Davison, “Empirical study of topic modeling in Twitter,” in Proc. First Workshop Social Media Analytics, 2010, pp. 80–88.

8. E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD), 2011, pp. 1082–1090.
9. S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-spatial properties of online location-based social networks," in Proc. 5th Int. AAAI Conf. Weblogs Social Media (ICWSM), 2011, pp. 329–336.
10. M. Jiang, P. Cui, and C. Faloutsos, "Inferring strange behavior from connectivity pattern in social networks," in Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining (PAKDD), 2014, pp. 126–138.
11. K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," J. Mach. Learn. Res., vol. 3, no. 6, pp. 1025–1058, 2003.
12. Y. Hu, S. D. Farnham, and A. Monroy-Hernandez, "Whoo.ly: Facilitating information seeking for hyperlocal communities using social media," in Proc. SIGCHI Conf. Human Factors Comput. Syst. (CHI), 2013, pp. 3481–3490.
13. A. Sadilek, J. Krumm, and E. Horvitz, "Crowd physics: Planned and opportunistic crowdsourcing for physical tasks," in Proc. ACM Conf. Ubiquitous Comput. (Ubicomp), 2013, pp. 1007–1014.
14. H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in Proc. 7th ACM Int. Conf. Web Search Data Mining (WSDM), 2014, pp. 373–382.
15. R. Li, S. Wang, and K. C.-C. Chang, "Multiple location profiling for users and relationships from social network and content," in Proc. 20th Int. Conf. World Wide Web (WWW), 2012, pp. 537–546.
16. M. A. Hossain and M. Zhang, "Differential privacy in location-based social networks: Are we there yet?" in Proc. IEEE 10th Int. Conf. Trust Security Privacy Comput. Commun. (TrustCom), 2013, pp. 889–898.
17. R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection," in Proc. 2nd ACM SIGSPATIAL Int. Workshop Location-Based Social Networks (LBSN), 2010, pp. 1–10.
18. Y.-S. Lim, M. D. Hossain, and A. D. Cheok, "Using spatial and social proximity for tie strength estimation in geo-social networks," in Proc. 4th Int. Conf. Internet Multimedia Comput. Service (ICIMCS), 2013, pp. 63–68.
19. J. Cranshaw, R. Schwartz, J. I. Hong, and N. Sadeh, "The livehoods project: Utilizing social media to understand the dynamics of a city," in Proc. 6th Int. AAAI Conf. Weblogs Social Media (ICWSM), 2012, pp. 58–65.
20. C. A. Davis Jr., G. L. Pappa, D. R. R. de Oliveira, and F. de L. Arcanjo, "Inferring the location of Twitter messages based on user relationships," Trans. GIS, vol. 15, no. 6, pp. 735–751, 2011.

21. D. Jurgens, “Location inference from sparse and noisy data,” in Proc. 21st Int. Conf. World Wide Web (WWW), 2013, pp. 691–702.
22. J. Liu, H. Yin, and F. Xu, “An efficient collaborative filtering framework for location-based services,” in Proc. 2013 IEEE 29th Int. Conf. Data Eng. (ICDE), 2013, pp. 430–441.
23. V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, “Collaborative filtering meets mobile recommendation: A user-centered approach,” in Proc. 24th AAAI Conf. Artif. Intell. (AAAI), 2010, pp. 236–241.
24. R. Ghosh and K. Lerman, “Predicting influential users in online social networks,” in Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM), 2011, pp. 317–326.
25. A. Sadilek, H. Kautz, and V. Silenzio, “Predicting disease transmission from geo-tagged social media,” in Proc. 25th AAAI Conf. Artif. Intell. (AAAI), 2012, pp. 136–141.
26. Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, “Exploring millions of footprints in location sharing services,” in Proc. 5th Int. AAAI Conf. Weblogs Social Media (ICWSM), 2011, pp. 81–88.