Spatial Textual Top-k Search in Mobile Peer-to-Peer Networks

Thao P. Nghiem[1], Cong Ma[1], J. Shane Culpepper[1], and Timos Sellis[2]

[1] RMIT University, Melbourne, Australia
jessie.nghiem@rmit.edu.au
s3496858@student.rmit.edu.au
shane.culpepper@rmit.edu.au
[2] Swinburne University, Melbourne, Australia
tsellis@swin.edu.au

Abstract. Mobile hardware and software is quickly becoming the dominant computing model for technologically savvy people around the world. Nowadays, mobile devices are commonly equipped with GPS and wireless connections. Users have also developed the habit of regularly checking into a location, and adding comments or ratings for restaurants or any place of interest visited. This work explores new approaches to make data available from a local network, and to build a collaborative search application that can suggest locations of interest based on distance, user reviews and ratings. The proposed system includes light-weight indexing to support distributed search over spatio-textual data on mobile devices, and a ranking function to score objects of interest with relevant user review content. From our experimental study using a Yelp dataset, we found that our proposed system provides substantial efficiency gains when compared with a centralised system, with little loss in overall effectiveness. We also present a methodology to quantify efficiency and effectiveness trade-offs in decentralized search systems using the Rank-based overlap (RBO) measure.

1 Introduction

Location-aware services are becoming increasing popular in advanced database applications. One of the most important fields in location-aware services is local business search using associated user reviews and ratings. For example, a person moves to a new suburb, and wishes to find an affordable Chinese restaurant. They can go to Zomato to search for local restaurants, and read user reviews and ratings [3]. This type of search involves both spatial and textual search. The expected results of this type of search are a list of the k highest ranking objects according to some spatial and textual similarity metric. To rank the result, many different scoring functions could be applied. One example is a linear combination of the spatial relevance of the location of the objects and the query point, along with the user-rating and the textual relevance between query keywords and user review contents.

In the vast majority of previous research, search systems combining spatial and keyword queries are centralised, which requires a single server to store data and process

[3] https://www.zomato.com/

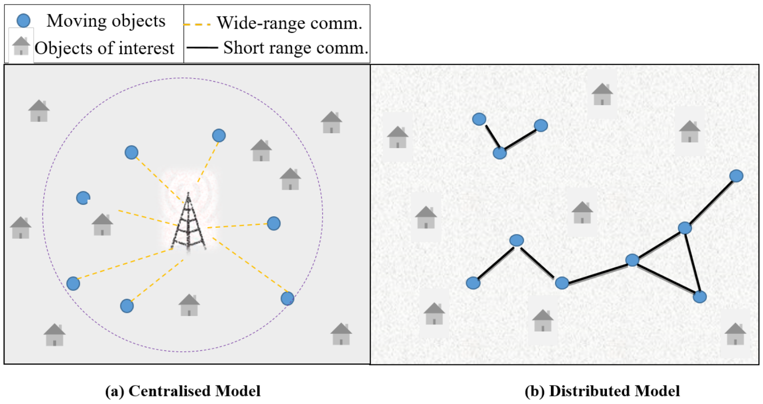

Fig. 1: Centralised Systems versus P2P Systems. Circles represent moving objects; house symbol: objects of interest; dashed lines: wide-range communication; continuous lines: short-range communication; dots: the base station network range.

queries [7, 13, 14, 19]. This model is shown in Fig. 1(a). However, a single point of failure makes the system susceptible to too much traffic, natural disasters, and/or denial of service attacks, which can lead to widespread disruptions [11]. Moreover, with the continual growth of data, update costs and storage is a persistent issue.

This work proposes a collaborative P2P search framework with spatial and textual indexing. It does not rely on a centralised server to store data or process queries; instead the queries are processed by each mobile device. Specifically, the overall contributions of this research are as the following:

1. We introduce a new direction in mobile distributed location-aware search for spatial and textual data.
2. We propose an indexing structure for textual data (user reviews) that can be searched for cached locations of interest associated with reviews from other mobile users.
3. A ranking function is also developed to score objects of interest with relevant user review content.
4. From our experimental study using a real-dataset, we found that our proposed system is substantially more efficient than a centralised system. We also show that the effectiveness of the proposed distributed workload model is comparable to the centralised approach using the RBO measure [18].

2 Background

2.1 Top-k kNN queries

Top-k search in an important application in geographical information retrieval. This type of query returns a ranked list of the top-k documents, ranked by spatial and textual similarity. These queries are supported through a variety of spatial and textual indexing data structures.

There have several recent studies on new approaches to combining spatial and textual indexes [3, 6, 7, 14, 19]. In general, they can be classified into two categories. The first approach maintains two independent indexing structures; one for text (for example, inverted files) and another for spatial data (such as: R-tree and variants [9]). For example, the work in [20] loosely combined R*-tree indexing. Other spatial indexes like grid-based and space filling based structures are also possible [6, 17]. The queries are answered by using a spatial index as an initial filter, and reranking the remaining items with an inverted index, or vice versa. The second approach focuses on a more tightly combined scoring reqime by combining both data representations into a single, hybrid index. The hybrid indexes simultaneously handle spatial and textual pruning to produce the final top-k result set. The IR-tree [7, 12] is the most widely used indexing structure in this group. Conceptually, an IR-tree is an R-tree, where each node is augmented with an inverted file.

To estimate the relevance between documents and user queries, a *scoring function* must be defined. In particular for geographical textual search, the scoring function can be composed of two main components: textual relevance and spatial relevance [2, 12, 14]. Textual relevance can be measured using variants of the TF-IDF model such as BM25, or the language model. Spatial relevance is often measured using a distance metric such as Euclidean or network distance.

2.2 Mobile collaborative caching and local distributed query processing

With the development of the state-of-the-art wireless communication technologies, such as IEEE 802.11 and Bluetooth, mobile collaborative caching has increasingly drawn attention as an alternative for information sharing among mobile hosts over standard centralized models. In general, the technique can improve data retrieval performance by allowing moving objects to access local caches on peers [4]. The first on-demand distributed data sharing algorithm for kNN queries was introduced by Ku and Zimmermann [11]. The scenario is shown as follows. The query node collects and verifies information from peers. If results cannot be verified, they are sent to the server or base station (BS). The BS will complete the task, and send the result back to the query node. This approach is efficient in reducing server workload, and alleviating traffic congestion in the BS.

A distributed multi-dimensional index structure, called P2PRdNN was introduced by Chen et al. [1] to efficiently support reverse nearest neighbour queries. Other related work [5, 16] proposed a framework to find an approximate answer for spatial-only range and kNN queries. Another solution for nearest neighbour queries in static sensor networks called a *peer-tree* was proposed by Demirbas and Ferhatosmanoglu [8] The approach is not amenable to mobile P2P environments due to the fixed communication infrastructure.

The novelty of our approach is that query processing and indexing is accomplished using a purely distributed spatial and textual search model. Top-k range queries are answered only by harnessing the power of peer collaboration without any central supervision.

3 Proposed Model

3.1 System model and assumptions

We assume a mobile network with no central supervision, where query objects and peers are dynamic as is commonly found in a mesh network. The environment is a symmetric system where each moving object, such as a smart mobile phone or tablet, can be both a query node and a peer of other nodes. Moving objects are also self-aware of their current location through an equipped GPS. The location of moving objects and objects of interest mentioned in this paper are physical locations.

Moving objects are equipped to support ad-hoc communication with other moving neighbours via Bluetooth, Wireless Local Networks (WLANs), Wireless Local Personal Networks (WPANs), or WiFi Direct – an emerging form of P2P communication. In addition, points of interest are randomly distributed in the network. To enhance the P2P query processing, a memory cache is assigned to store spatial data for points of interest from previous queries. A priority queue manages requests based on the distance from the cached point of interest to the moving object. When the cache (priority queue) is full, new points of interest will be cached only if their ranking score is high enough to displace the k-th ranked object in the cache. This deletion strategy assures that cached data is the most useful answer for future queries, and increases the accuracy of the query results.

3.2 Query models and message types

The proposed system is designed to answer top-k kNN queries based on the available information from peers. The query point is always at the same location as the moving object issuing the query. The information from peers is the result from previous queries stored in memory cache of peers.

There are four distinct message types between the query node q and a peer p_j:

1. A beacon message (*beacon_msg*) is broadcast from the query node to detect peers within communication range.
2. An acknowledgement message (*ack_msg*) from peers to the query node to assign a location to the responding peers.
3. A query message (*query_msg*) from the query node to the selected peers asking for points of interest cached locally by those peers.
4. A query reply (*reply_msg*) from peers to the query node with a possible answer from the cache of the peer. The answer consists of the location, and the type of the point of interest.

Here the system is working with cached data from moving objects; therefore, it is expected that the number of points of interest stored in each cache is relatively small. Hence, a spatial index is not necessary. Instead, indexing text data (user reviews) associated to the objects of interest is most important.

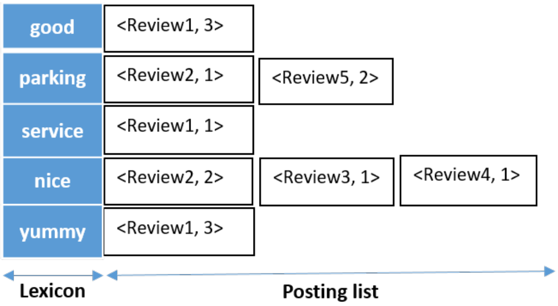

Fig. 2: Indexing structure for user reviews.

3.3 Indexing user reviews and ranking function

Indexing structure To support text search, an indexing structure for cached user reviews at each moving object is required. As the storage and computation for the moving objects is limited, a simple inverted index is used. First, the reviews are read from the dataset file. Then all the terms minus stop words are extracted, formatted into lower case and registered in the indexing file [15]. The indexing structure consists of a lexicon and a posting list, as shown in Figure 2.

Ranking function For a query q with location l_q, a set of keywords $t \in q$, and a candidate user review r with location l_r, the combined score of q and r, $s(q, r)$ is computed as:

$$s(q, r) = w_1 \times \mathit{ls}(l_q, l_r) + w_2 \times \mathit{ts}(q, r) + w_3 \times \mathit{rs} \tag{1}$$

where $w_1, w2, w3 \in (0, 1)$ are the parameters used to weight the importance of the spatial, textual or rating components, $w_1 + w_2 + w_3 = 1$, *ls* is the spatial relevance component, and *ts* is textual relevance component. The normalised user rating *rs* is also associated with the object for each user (node).

$$\mathit{rs} = \frac{\mathit{user\_rating}}{\mathit{rating}_{\mathit{max}}} \tag{2}$$

where $\mathit{rating}_{\mathit{max}}$ is the maximum rating allowed.

Spatial relevance component In this model, Euclidean distance is used to measure the distance between the objects of interest and the query point. The spatial relevance score *ls* is computed as below:

$$\mathit{ls}(l_q, l_r) = 1 - \frac{\mathit{distance}_E(l_q, l_r)}{\mathit{distance}_{\mathit{max}}} \tag{3}$$

where $\mathit{distance}_{\mathit{max}}$ is the maximum distance from two unique points in the geographical space.

```
Data: Query node $q$, transmission range $R$
Result: At node $q$, a priority queue of peers $P$
begin
    $P \leftarrow \varnothing$
    Node $q$ broadcasts a one-hop beacon_msg to every peer.
    foreach $p_i$ in range $R$ do
        if a peer $p_i$ receives beacon_msg then
            $p_i$ sends ack_msg with a location and ID to $q$.
        end
    end
    if $q$ receives an ack_msg from $p_i$ then
        $P \leftarrow P \cup \{p_i\}$.
    end
end
```

Algorithm 1: Initialisation and Peer Discovery

Textual Relevance Component Variants of TF-IDF are the most commonly used textual similarity metric, and are used in this work. Specifically, the calculation of this score is as the following:

$$ts(q,r) = \frac{\sum_{t \in q} tf_{r,t} * log \frac{N_r}{df_{r,t}}}{ts_{max}} \tag{4}$$

where $tf_{r,t}$ is the number of times the term occurs in each review, N_r is the total number of reviews, and $df_{r,t}$ is the total count of the term in the collection.

3.4 System details

Overview Our ultimate goal is to harness the collaborative power of mobile devices to process spatial and keyword queries locally. Overall, the proposed system is divided into two primary phases: (1) Initialisation and Peer Discovery Phase; and (2) Query Processing. Each phase is described in detail below.

Initialisation and Peer Discovery Phase Each moving object maintains a default map of the associated objects with user reviews and ratings in a cache. This can be loaded during the initialisation phase, or downloaded from a local provider. Since mobile users move frequently, the associated peers also change. As a result, before starting to send queries, a query node q needs to discover which moving objects are in communication range by sending a one-hop broadcast message. Moving objects receiving the broadcast message send an acknowledgement message which contains their ID and location information. More specifically, this phase is described in Algorithm 1. The query node q collects all acknowledgement messages from the surrounding nodes to construct a peer list. Note that q is assigned an acknowledgement time-out period. Therefore, q waits to receive acknowledgement messages from peers for a fixed period of time.

```
Data: A query node $q$ initialised with the number of ranked results required $k$, a range
      $range$, a set of keywords $QK$. On each peer $p$, a set of cached objects of interest
      $IO_p$, and an indexing structure $I$ of user reviews $R$
Result: To node $p$, a set of sorted objects of interests $Result_p$ with relevant user reviews
        and ratings
begin
    foreach $IO_i$ in $C$ do
        if $distance(l_{IO_i}, l_q) > range$ then
            $C \leftarrow C - \{IO\}_i$
        end
        else if $IO_i$ has no review containing any $k_i \in QK$ then
            $C \leftarrow C - \{IO\}_i$
        end
    end
    foreach $IO_i$ in $C$ do
        $score \leftarrow 0$
        $review_{no} \leftarrow 0$
        foreach $r_i$ in $R$ do
            Calculate score $s(r_i)$ using the ranking function in Section 3.3.
            Increment $review_{no}$.
            $score += s(r_i)$
        end
        $score = score / review_{no}$
    end
    Return $k$ objects in $C$ with the highest score.
end
```

Algorithm 2: Query Processing Algorithm

Query Processing Phase After the first stage, q is aware of all peers close enough to query. When a peer receives the query, data is retrieved from the local cache, followed by pruning and ranking which is computed as the follows.

Pruning objects and user reviews at peers. For the user reviews associated with the candidate set, if the reviews contain the keywords, they will be ranked using the ranking function described in equation (1). The ranked lists of reviews and objects of interest from the peers are sent to the query object. Each query object then collects objects of interest sorted by similarity, and return the top k objects of interest along with the relevant user reviews. Overall, the query processing phase is summarised in Algorithm 3.

4 Performance Evaluation

4.1 Simulation Setup and Configuration

All results are computed using the MiXiM simulation environment, which is derived from an OMNeT++-based framework to model and analyse Mobile P2P Query Processing Systems [10]. Each moving object contains 8 modules as shown in Figure 3.

```
Data: Query node $q$, on $q$, a set of peers $P$, $k$ value, range $range$, set of keywords $QK$,
      and expiry_time
Result: To node $q$, a set of sorted objects of interest $IO_q$ with relevant user reviews and
        ratings
begin
    Node $q$ sends a query set of keywords $QK$ to every peer in $P$.
    Node $q$ starts a timer waiting_time.
    foreach $p_i$ in $P$ do
        Node $p$ calls PeerRankingFunction (Algorithm 2) for local cached data.
        Node $p$ return the top $k$ results to $q$.
    end
    if waiting_time $>$ expiry_time then
        Node $q$ keeps only the top $k$ $IO_i$ from peer with highest score.
        Return $IO_q$ with relevant user reviews and ratings.
        Node $q$ updates cached data.
    end
end
```

Algorithm 3: Query Processing Algorithm

Table 1: Simulation parameters

Parameter	Value
Playground	5km × 5km
Number of reviews	10000
Number of moving objects	600
k	10
Expected number of queries generated	1000
Cache Size	1000 reviews
Simulation time	600 sec

Here we use Nic80211 for the Wi-Fi connection. According to the configuration for the network interface cards in MiXiM, a transmission current *txCurrent* $= 153$mA and a receiving current *rxCurrent* $= 200$mA are used. The communication to the server is conducted via 3G (WCDMA), band I-2100 which is used by Vodaphone and Optus in Australia. Data rate for high speed moving objects in this network is 128 kbps [4] and current consumption in connected state is 365.6 mW [5]. All queries are generated using a Poisson arrival model. A universal λ is assigned to all moving objects to represent the average number of queries arriving per unit time; or the expected number of queries generated by each moving object is $E(N) = \lambda T$ where T is the simulation time. Initially, each moving object is assigned random objects of interest and the corresponding user review and added to the local cache. The query expiry time is set to 30 sec. After this time, even if there are still peers to query, the query object aborts communications, and processes the current results.

[4] http://www.silicon-press.com/briefs/brief.3g/index.html

[5] http://www.option.com/en/newsroom/media-center/white-papers/

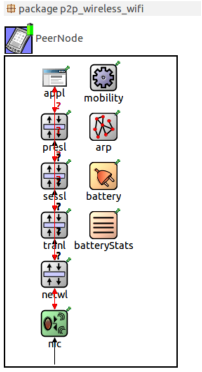

Fig. 3: Moving objects' modules

The simulation was ran using both real and synthetic datasets for a large-scale network. Empirically, we set w_1 (distance weight) to 0.8, w_2 (text relevance weight) to 0.15 and w_3 (rating weight) to 0.05 in the ranking function as locality is the most important feature. Data is sampled from Yelp dataset [6]. This dataset includes data for businesses in America, including location, attributes, user ratings and reviews. In this simulation, we consider a subset of restaurants in Las Vegas with total $10{,}000$ user reviews. To prevent reviews from popular restaurants dominating the dataset, a maximum of 20 reviews for each restaurant is used. In the initial stage, each moving object caches a number of restaurants in the neighbourhood area with the associated user reviews, which is defined by the cache size parameter. Other parameters are described in Table 1.

4.2 Simulation Results and Discussions

In this section, we evaluate the performance of the proposed model in term of efficiency and effectiveness when varying the parameter values. The evaluation is based three different measures: processing time (from the time a query object discovers peers to the time the query is answered), energy consumption (energy spent at each moving object) and RBO (a similarity measure between incomplete rankings that handles non-conjointness, and gives higher weight to higher ranking objects). It is noted that to calculate RBO, the query results from the proposed method are in comparison with that of the central method.

Figure 4 compares the efficiency of the proposed P2P Model to the centralised model. In particular, energy consumption and query processing time are measured at the node level. Figure 4(a) clearly shows that the total energy consumption of the centralised model is higher than in the P2P model. In this simulation, mobile nodes do not go into sleep mode. Therefore, the receiving energy consumption at each node is stable, and much greater than the transmission energy consumption. That is why the transmission energy consumption has little fluctuation in Figure 4(b). As expected, increasing

[6] https://www.yelp.com/academic_dataset/

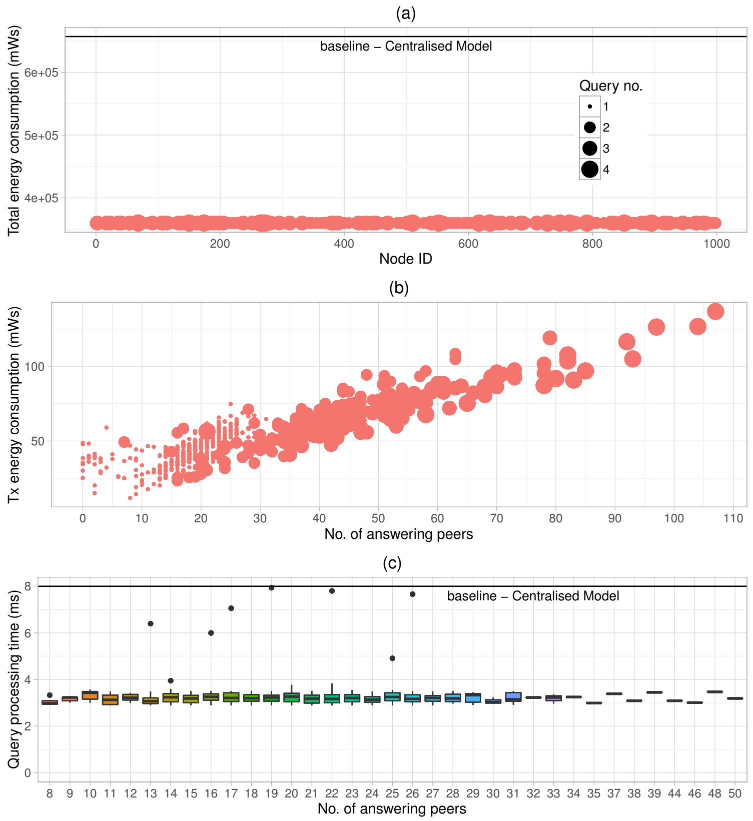

Fig. 4: Efficiency

the number of peers queried or queries result in higher energy consumption. In addition to the energy efficiency, Figure 4(c) indicates that the processing time of the P2P model can save up to 25% over the centralised model. This is due to the difference costs in wide-range communications between the server and mobile objects, versus short-range P2P communications.

Figure 5 shows the accuracy of top-k results in the P2P model. The centralised model is the ground truth as this system can exhaustively process the entire dataset, while moving objects in the P2P model only cache a subset of review lists. This trade-off in query processing time and power consumption is the core idea exploited in our model. Increasing the size of the dataset subset cached in moving objects, the RBO increases as expected. There is a three fold increase in RBO when the initial number of reviews in the memory cache changes from $1{,}000$ to $5{,}000$. Another possible way to improve the accuracy is to select the most relevant objects, and load them into the initial cache. In this simulation at the initialisation stage, reviews related to the nearest restaurants to the moving objects are randomly chosen during cache initialisation. Figure 5 shows the

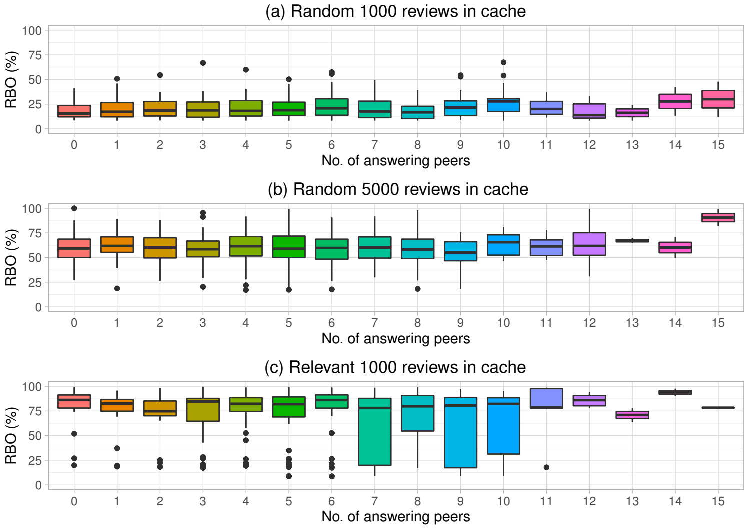

Fig. 5: Effectiveness

effects on RBO when relevance caching is used, achieving an RBO of 75% when the cache size is $1{,}000$ reviews (10% of the collection).

5 Conclusions and future work

We present a peer-to-peer solution to solve the problem of retrieving the top-k spatial-textual objects of interest that are associated with a list of relevant user reviews and ratings. The proposed model harnesses the power of collaboration between moving objects, and requires no central supervision. We also apply a simple indexing structure that is suitable for shared data in mobile networks, and develop a ranking function that considers several different factors including distance, rating, and user review relevance. The simulation results demonstrate the feasibility of our model.

The work has a number of promising extensions in the future. First, the model can be applied in a trusted social network. In particular, a query object can ask not only the surrounding objects, but also friends that are not spatially close through a social network. Second, the search model can be personalised, depending on the user preferences. User profiles could become a valuable criteria for improving search result. Finally, it would be interesting to consider an incentive model to encourage data sharing in distributed mobile query processing environments.

Acknowledgment. This work was supported by the Australian Research Council's *Discovery Projects* Scheme (DP140101587). Shane Culpepper is the recipient of an Australian Research Council DECRA Research Fellowship (DE140100275).

Bibliography

[1] D. Chen, J. Zhou, and J. Le. Reverse nearest neighbor search in peer-to-peer systems. In *Flexible Query Answering Systems*, volume 4027, pages 87–96. 2006.

[2] L. Chen, G. Cong, C. S. Jensen, and D. Wu. Spatial keyword query processing: an experimental evaluation. *PVLDB*, 6(3):217–228, 2013.

[3] F. M. Choudhury, J. S. Culpepper, T. Sellis, and X. Cao. Maximizing bichromatic reverse spatial and textual k nearest neighbor queries. *PVLDB*, 9(6):456–467, 2016.

[4] C. Chow, H. V. Leong, and A. T. S. Chan. GroCoca: group-based peer-to-peer cooperative caching in mobile environment. *J. on Sel. Areas in Comm.*, 25(1):179–191, January 2007.

[5] C. Chow, M. Mokbel, and H. Leong. On efficient and scalable support of continuous queries in mobile peer-to-peer environments. *IEEE Trans. on Mob. Comp.*, 10:1473–1487, 2011.

[6] M. Christoforaki, J. He, C. Dimopoulos, A. Markowetz, and T. Suel. Text vs. space: Efficient geo-search query processing. In *Proc. CIKM*, pages 423–432, 2011.

[7] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, Aug. 2009.

[8] M. Demirbas and H. Ferhatosmanoglu. Peer-to-peer spatial queries in sensor networks. In *Proc. P2P*, pages 32–39, 2003.

[9] A. Guttman. R-trees: a dynamic index structure for spatial searching. *SIGMOD Record*, 14:47–57, June 1984.

[10] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In *Proc. STTCNS*, 2008.

[11] W. Ku and R. Zimmermann. Nearest neighbor queries with peer-to-peer data sharing in mobile environments. *Pervasive and Mobile Computing*, 4(5):775 – 788, 2008.

[12] Y. Li, H. Chen, R. Xie, and J. Z. Wang. Bgn: A novel scatternet formation algorithm for bluetooth-based sensor networks. *Mobile Information Systems*, 7:93–106, 2011.

[13] Z. Li, K. C. K. Lee, B. Zheng, W.-C. Lee, D. Lee, and X. Wang. IR-Tree: An efficient index for geographic document search. *TKDE*, 23(4):585–599, Apr. 2011.

[14] J. Mackenzie, F. M. Choudhury, and J. S. Culpepper. Efficient location-aware web search. In *Proc. ADCS*, pages 4:1–4:8, 2015.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[16] T. P. Nghiem, K. Maulana, D. Green, A. B. Waluyo, and D. Taniar. Peer-to-peer bichromatic reverse nearest neighbors in mobile ad-hoc networks. *JPDC*, 74(11):3128 – 3140, 2013.

[17] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson. Spatio-textual indexing for geographical search on the web. In *Proc. SSTD*, 2005.

[18] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4):20:1–20:38, Nov. 2010.

[19] D. Zhang, C.-Y. Chan, and K.-L. Tan. Processing spatial keyword query as a top-k aggregation query. In *Proc. SIGIR*, 2014.

[20] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for location-based web search. In *Proc. CIKM*, 2005.