Spatial Semantic Search in Location-Based Web Services

Jeong-Hoon Park [1]
Supervised by Chin-Wan Chung [2]
Department of Computer Science, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
jhpark@islab.kaist.ac.kr[1], chungcw@kaist.edu[2]

ABSTRACT

As GPS-enabled mobile devices have advanced, the location-based service(LBS) became one of the most active subjects in the Web-based services. Major Web-based services such as Google Picasa, Twitter, Facebook, and Flicker employ LBS as one of their main features. Consequently, a large number of geotagged documents are generated by users in the Web-based services. Recently, there have been studies on the spatial keyword search which aims to find a set of documents in the Web-based services by evaluating the spatial relevance and keyword relevance. It is a combination of the spatial search and keyword search, each of which has been studied for a long time.

In this paper, we address the spatial semantic search problem which is to find top k relevant sets of documents with spatial constraints and semantic constraints. For devising an effective solution of the spatial semantic search, we propose a hybrid index strategy, a ranking model and an efficient search algorithm. In addition, we present the current status of our research progress, and discuss remaining challenges and future works.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Spatial Semantic Search, LBS, Web Services

1. INTRODUCTION

Recently, the location-based service(LBS) became one of the most popular trends in Web-based services. LBS is a service which is accessed by mobile devices and mobile networks, and utilizes geographical positions of the mobile devices. The demands on LBS increased as the capability of GPS-enabled mobile devices such as smart phones and tablet PCs has rapidly advanced. In addition, diverse map-based Web services are developed and open APIs for accessing and utilizing the services are provided (e.g., Google Maps API, Microsoft Bing Maps API and Yahoo Flickr API), so that the barriers to developing and launching the location-based Web service is lower than before. Furthermore, social network services such as Twitter, Facebook, and Foursqaure have absorbed the LBS for facilitating the mobile users to share the information of their private activities and experiences in any place in real time. As the location-based services can make more tight-relations among the Web users by providing an easy way for sharing the experiences in their lives, it becomes one of the most important features of current Web-based services. Consequently, a large amount of geotagged documents that are generated by users have been collected. It provides a great opportunity for providing more advanced Web-based services. As an advanced Web-based service, the spatial keyword search has been proposed in recent years. Given geotagged Web documents and a user query, the spatial keyword search is finding the relevant documents which satisfy the geographical constraints such as the range intersection and distance proximity, and semantic constraints such as the keyword relevancy.

There have been studies on the spatial keyword search. [15, 5, 4, 2, 8] proposed methods for the spatial keyword search. It helps users to find appropriate Web documents located in places near to the query point. However, the previous researches only focus on finding the Web documents, not on finding geographical areas. For the mobile users, it can be very useful to find the areas related to a particular semantic or topic. For example, the query, 'In the New York city, find the areas with less than 50 meters of diameter in which an Apple store and a subway station are located in', can be useful for the visitors to the New York city. The methods proposed in the previous studies are not appropriate for searching such areas since a single document cannot describe all the information about the area. As a solution for searching areas, [1] introduces the collective spatial keyword search. The collective spatial keyword search is, given a query consisting of a spatial point and a set of keywords, to find the set of documents such that each query keyword is matched to one of the documents in the set, and the ranking score considering the diameter of the set and the distance from query point is minimum. However, the keyword matching for the selection of the set is not accurate. For instance, assume that the set of query keywords contains 'vehicle'. In this case, the documents containing the keyword 'car' cannot be considered in the query processing. Also, assume that 'JAVA' is included in the set of query keywords. This keyword is matched to two different documents such that one contains 'JAVA coffee' and the other contains 'JAVA Programming language'. Consequently, the simple keyword matching cannot guarantee the quality of the search result.

In this paper, we propose an effective method to search top k interesting sets of documents in the LBS-based Web service, each of which satisfies geographical constraints and semantic constraints in the user query. In order to effectively evaluate the relevance of the

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2567953.

semantics, instead of using keywords of documents, we use semantic annotations of documents. Semantic annotation [6] is the set of semantic concepts of the ontology which represents the semantics of the Web contents in a machine-understandable format. By using the relationships among the semantic concepts, we can improve the accuracy of the search results. In addition, the proposed method partitions the documents and constructs an R*-tree for each partition in the indexing time. Then, our method uses only necessary parts of the R*-tree in the search time, so that the efficiency of the query processing can be improved.

This paper is organized as follows. Section 2 reviews related works. The problem to solve is formally defined in Section 3. Section 4 presents the proposed approach for the solution of our problem. Section 5 describes the methodology to be used for evaluating the proposed method. In Section 6, the current status of the PhD research is described. Finally, we provide conclusions and present the future works in Section 7.

2. RELATED WORKS

In recent years, there have been studies on the spatial keyword search.

[13] and [14] are the researches for the m-CK search that is to find the most relevant set of multiple geotagged documents with the m keywords constraints.

In [13], authors firstly introduce the m-closest keywords query (m-CK query). Given a user query and a spatial database in which each tuple corresponds to a geotagged document, the m-CK query aims to find the spatially closest tuples which match the m keywords in the user query. The bR*-tree is proposed for indexing the geotagged documents. Each node of the bR*-tree has a node MBR, a bitmap file which is equal to the signature file of the previous researches, and keyword MBRs each of which has a corresponding keyword. A keyword MBR is the minimum bounded rectangle of the documents having the corresponding keyword that are reachable from the node.

In contrast to [13], the authors in [14] propose a comparably lightweight index for an m-CK query by labeling each node in the bR*-tree, and constructing an inverted index for the labeled nodes. In the search time, a virtual bR*-tree is constructed using only the nodes related to the keywords in the user query. By reducing the number of nodes that are loaded into memory, the efficiency is improved. Even though the authors propose an improvement of the bR*-tree, the size of the nodes that are loaded into memory during the search time is still large because of the bitmap file.

Along with the studies [13, 14], the multiway spatial join method [11, 10] can be a solution to the m-CK query. However, the efficiency of the multi-way spatial join is worse than [13, 14].

[1] and [9] studied the collective spatial keyword searches which are similar to the problem of [13] and [14]. The main difference is that the collective spatial keyword searches considers the distance between the set of geotagged documents and the query point.

In [1], the authors address two types of collective spatial keyword searches and provide the proofs that the problems are NP-complete. In the cost function, the type1 problem considers only the diameter of the set while the type2 problem considers both the diameter and the distance from the query. Then, the authors propose exact algorithms and approximated algorithms for solving the problems.

The problems of [9] are almost equal to the type1 and type2 problems of [1]. The authors in [9] also address two types of collective spatial keyword searches, $MaxSum - CoSKQ$, and $DIa-CoSKQ$. $MaxSum-CoSKQ$ is equal to the type2 problem. $DIa - CoSKQ$ considers only the diameter of the set as the type1 problem for the cost function. However, $DIa - CoSK$ employs a different const function for measuring the diameter. More efficient exact algorithm and approximated algorithm are proposed in [9]. The improvements are achieved by the distance owner driven approach. All the feasible set can be categorized by the distance owner consisting of 3 objects in the set. Then, by using the pruning methods, the search space can be dramatically reduced so that the efficiency is improved.

The existing solutions of m-CK search and the collective spatial keyword searches cannot be directly adopted for solving our problem. It is because the existing problems utilize the keywords only for the selection of the set of documents while our problem utilizes the semantic concepts for not only the selection of the set but also the evaluation of the rank score. In addition, we consider the semantic relationships such as the $subClassOf$ and $instanceOf$ relationships in the selection of the set of documents.

3. PROBLEM DEFINITION

In this section, we formally define the problem to be solved. Before defining the problem, the assumptions and definitions used in this paper are introduced.

We assume that there exists a spatial database D, in which each tuple of the database represents a geotagged and semantically annotated document.

We denote the user query as q = (k, loc, l, dis, SEM). k is the number of results that will be returned. loc is the geographical position of the query point. l is the maximum limit to the geographical size of a set of documents. dis is the maximum limit to the distance of a set of documents from the query point. SEM = {c_1, c_2, .. , c_m} is the set of semantic concepts. Each semantic concept $c_i \in SEM$ is a class or instance of the ontology.

l and dis can be directly input by the user issuing a query. However, as a practical way, l and dis can be systematically provided. For example, by referring to the current zoom level of the GUI interface, l and dis is automatically derived and set.

We define 'complete closure' and use it in the later part of this paper.

DEFINITION 1 (COMPLETE CLOSURE). *We call a set of documents a 'complete closure' if the set satisfies all the following conditions:*
***(Condition. 1)** The geographical size of the set of documents is less than $q.l$.*
***(Condition. 2)** The distance from $q.loc$ to the set is less than $q.dis$.*
***(Condition. 3)** Each semantic concept in $q.SEM$ is matched to only one document in the set, which has the concept in its semantic annotation.*

The geographical size of a set of spatial objects (e.g. MBRs of a tree nodes, or location points of documents) is computed as the follow:

$$GSize(S_k) = \max_{(o_i,o_j)\in S_k \times S_k, i\neq j} minDist(o_i.loc, o_j.loc) \quad (1)$$

where S_k is a set of objects, o_i is an object in S_k and $minDist(A, B)$ is the minimum Euclidean distance between A and B. In addition, the distance between $q.loc$ and a set of objects is computed as the follow:

$$GDist(q.loc, S_k) = \min_{o_i\in S_k} minDist(q.loc, o_i.loc) \quad (2)$$

where S_k is a set of objects, o_i is a document in the closure, $o_i.loc$ is the location of o_i and $minDist(A, B)$ is the minimum Euclidean

distance between A and B. The geographical size of a set of documents is defined as the maximum of the distances of pairs of documents. With the maximum distance, the proposed method becomes compatible with existing systems.

The spatial semantic search is, given a user query q, to efficiently find the top k complete closures which are physically and semantically relevant. The physical relevancy of a closure clo_x is measured by using the geographical size of clo_x, and the distance between $q.loc$ and clo_x. The degree of semantic relevancy of clo_x is measured by using the semantic similarity between $q.SEM$ and clo_x.

4. PROPOSED APPROACH

4.1 Indexing Strategy: SR*-Tree

In this section, we propose a hybrid index, Semantic R*-tree (SR*-Tree) for the geotagged and semantically annotated documents.

4.1.1 Documents Setting

Before constructing the SR*-tree, it is necessary to generate the semantic annotations of documents. If the semantic annotation of a document contains two concepts c_i and c_j such that c_i is an ancestor concept of c_j, we remove c_i in order to avoid the redundant indexing of the document. For example, if a document has both the 'Automobile' and 'Sedan' concepts in the semantic annotation, we remove the concept 'Automobile' from the document. Even when the user query contains only 'Automobile', our method can find the documents which have the concept 'Sedan' in their semantic annotation.

4.1.2 SR*-tree Construction

To construct the SR*-tree, we refer to the concept hierarchy of the ontology. The ontology consists of semantic concepts and their relationships. In the ontology, a semantic concept is a class or an instance. Additionally, the relationships among semantic concepts are represented by 'property' in the ontology. The concept hierarchy of the ontology consists of the 'subClassOf' property and the 'instanceOf' property. The 'subClassOf' property denotes the parent-child relationship between the classes. For example, the classes 'Device' and 'Mobile_Phone' are connected by the 'subClassOf' property in the ontology. The 'instanceOf' property represents the instance-class relationship. For example, the 'Barack_Obama' instance and the 'USA_President' are connected by the 'instanceOf' property in the ontology.

We construct a tree in which each node corresponds to a semantic concept in the concept hierarchy. Each node have pointers to its sub nodes that correspond to the sub concepts of the concept of the node. Then, for each node, the proposed method constructs an R*-tree for the location points of the documents having the corresponding concept. A node in the SR*-tree refers to the root node of the corresponding R*-tree.

4.2 Top k Closure Search

In this part, we present an efficient method to search for the top k complete closures whose ranks are determined based on the proposed ranking model. Because a set of documents becomes a complete closure when all of the concepts in $q.SEM$ are related to the documents in the set and the geographical constraints are satisfied, the algorithm synchronously traverses multiple R*-trees for the concepts in $q.SEM$. During the search time, the R*-trees which are not related to the concepts in $q.SEM$ are not considered. Thus, we can reduce the size of the search space at the beginning of the algorithm. In addition, the results are returned in an incremental manner for efficient calculation of the overall ranks of the complete closures.

We propose a ranking model, then we describe an efficient search algorithm using the proposed ranking model.

4.2.1 Ranking Model

In this section, we propose an effective model for ranking the closures. In order to measure the relevancy of a closure, the ranking model consists of the physical factor and the semantic factor. The physical factor is based on the geographical size and distance. We define the physical score as follows:

$$PS(clo_i) = \frac{1}{1 + ln(1 + GDist(q.loc, clo_i) \times GSize(clo_i))} \quad (3)$$

where clo_i is the input closure, $GDist(q.loc, clo_i)$ is the geographical distance of clo_i from the query point $q.loc$, and $GSize(clo_i)$ is the geographical size of clo_i.

The semantic factor is the semantic relevancy of the complete closure to the user query.

We devise an extended version of TF-IDF called a CF-IDF, which considers the relationships among the semantic concepts. CF-IDF is the acronym for Concept Frequency and Inverse Document Frequency. The following equation is the model for computing the semantic relevancy.

$$SS(clo_i) = \sum_{c_k \in q.SEM} CF(d(clo_i, c_k), c_k) \times IDF(c_k) \quad (4)$$

where clo_i is the input closure, c_k is a semantic concept in $q.SEM$, and $d(clo_i, c_k)$ is the document in clo_i which is matched to c_k.

The CF score is computed as follows:

$$CF(d_j, c_k) = \ln(1 + \frac{\sum_{c_q \in Subs(c_k)} Freq(c_q)}{|SemAnn(d_j)|}) \quad (5)$$

where $Subs(c_k)$ is the set of descendant concepts of c_k in the concept hierarchy, $SemAnn(d_j)$ is the set of concepts in the semantic annotation of d_j.

CF considers not only the concepts in $q.SEM$, but also their descendant concepts in the concept hierarchy. For example, if a document d_y has the 'starbucks' concept and there is the 'coffeeShop' concept in $q.SEM$, the ranking model takes the frequency of 'starbucks' into account when calculating $CF(d_y$,'coffeeShop').

The IDF score is computed as follows:

$$IDF(c_k) = \frac{|D|}{|\bigcup_{c_x \in Subs(c_k)} \{d_A | d_A \in D, c_x \in SemAnn(d_A)\}|} \quad (6)$$

where D is the set of all the document in the spatial database.

The total ranking score is computed as the follow:

$$RankScore(clo_i) = \alpha \times PS(clo_i) + \beta \times (1 - e^{-SS(clo_i)}) \quad (7)$$

For efficiency, the CF and IDF values of all of the concepts are computed in the indexing time.

4.2.2 Basic Search Algorithm

Before describing the algorithm, we define 'candidate closure' and 'sub closure'.

DEFINITION 2 (CANDIDATE CLOSURE). *The candidate closure is the set of elements each of which can be a document or a tree node of the R*-tree such that the geographical constraints $q.l$ and $q.dis$ are satisfied, each semantic constraint in $q.SEM$ is*

matched to only one element in the set, and at least one element of the set is a tree node.

DEFINITION 3 (SUB CLOSURE). *The closure C_A is a sub-closure of the closure C_B when, for each element e_l of C_A, e_l satisfies one of the following conditions:*
***(Condition. 1)** If e_l is a document, C_B has a data node referring to the document, or an ancestor node of the data node in its R*-tree.*
***(Condition. 2)** If e_l is a tree node, C_B has an ancestor node of e_l in its R*-tree.*
***(Condition. 3)** $e_l \in C_B$*

The basic idea of the proposed search algorithm is based on the incremental search. In our method, a priority queue(p-queue) is used to incrementally output the complete closures. The priority is based on the physical score of the closures. By replacing a candidate closure with its sub closures in the p-queue, we can get complete closures eventually. The incremental search enables the pipelined process between the tasks when selecting closures and computing ranking scores of the closures. As the first step, for each c_i, the algorithm gets root nodes from the R*-trees that correspond to c_i and its descendant concepts. For each c_i, the algorithm generates a new special node pointing to the root nodes of the corresponding R*-trees. The algorithm then checks if the set of the special nodes can be a candidate closure or not. If the set can be a candidate closure, the algorithm initializes a candidate closure with the special nodes and puts the candidate closure into the p-queue. The candidate closure is the starting point of the best-first search. The algorithm gets a closure at the head of the p-queue. If the closure is a complete closure, it is returned to evaluate the total ranking score. Otherwise, the algorithm generates the sub closures of the closure and enqueue them in the p-queue. This is repeated until, the algorithm is sure that there is no more closure to return or the overall algorithm is terminated. Among the complete closures whose total ranking scores are estimated, we choose the top k closures as the results. In order to show that the algorithm is correct, we have the following lemma.

LEMMA 4.1. *If the closure clo_i is a sub-closure of clo_j, clo_i cannot have a higher physical score than that of clo_j.*

PROOF. Assume that clo_i has greater physical score than clo_j. This implies that 1) there is a pair of elements P_k in clo_j such that the distance between the elements of P_k is larger than $GSize(clo_i)$, or 2) there exists an element e_x in clo_i such that $minDist(q.loc, e_x)$ is less than $GDist(q.loc, clo_j)$. We will show that 1) and 2) are contradictory conditions. We denote that an element e_{sub} is a child element of e when e has a pointer to e_{sub} in the R*-tree. Each element in clo_i has the parent-child relationship with an element in clo_j. For 1), there must be a pair in clo_i, which consists of the same or child elements of the elements in clo_j. The child elements have MBRs which the MBRs of the parent elements enclose. Therefore, the distance between the MBRs of the two elements in clo_i is equal to or greater than that of the counterpart elements in clo_j. This is a contradiction. For 2), there must be an element e_y in clo_j which is equal to e_x or is the parent element of e_x. If e_y is equal to e_x, this is of course a contradiction since $minDist(q.loc, e_x)$ is equal to $minDist(q.loc, e_y)$. If e_y is the parent element of e_x, the MBR of e_y encloses the MBR of e_x, $minDist(q.loc, e_x)$ is always equal to or greater than $minDist(q.loc, e_y)$. Therefore, this is also a contradiction. □

The remaining task for the top k closure search is developing pruning techniques for reducing the irrelevant search space. By the pruning techniques, it is possible to terminate the priority queue based algorithm in the early stage.

Table 1: Semantic Concepts used in Queries

Centralpark	Donut	Bridge	Asian_restaurent
Subway	Tatoo	Movie_KingKong	Parade
BroadWay_NYC	Art_Meseum	CoffeeShop	Maxican_Restraunt
Church	Police	City_Hall	Newyork_Film_Festival
zoo	Taxi	Skyscraper	Dance
Hamburger	Pizza	Bus_stop	HBO
Yoga	Parking	nationalmall	disco

5. METHODOLOGY

5.1 Evaluation Methodology

The methodology used in evaluating the proposed method is, based on a real dataset, to examine how effective the proposed method is in terms of the efficiency of the query processing and the accuracy of the top k result sets. For the evaluation, we will implement two systems for a naive method using the keyword search instead of the semantic search, and the proposed method, respectively. Then, we will compare the two systems in terms of the efficiency and accuracy.

For the evaluation of the efficiency, randomly generated 20 queries are generated. To the best of our knowledge, there have not been a system processing the spatial semantic search. Therefore, it is difficult to collect the real query logs. Table 1 shows the semantic concepts used in the user query. For the generation of queries, we randomly choose the combination from the concepts.

The generated queries are issued to the two systems and the systems will process the queries one by one. All the execution time and the amount of memory size will be the average value from the query processings over 20 queries.

For the evaluation of the accuracy, we will compare the precision of the returned top k sets of documents from the two systems. In order to evaluate the accuracy of the proposed ranking model, we will not use the synthetic dataset since it is difficult to determine the right answers. Also, $P@50$ is employed for this comparison, which represents the ratio of correct answers among the top fifty results of the method. Along with the precision, the recall is an important measurement of the accuracy. Computing the recall requires finding the complete set of the correct answers which is called a ground truth. However, it is difficult to find the ground truths for all of the queries. Instead of finding ground truth for all the real dataset, we randomly choose 5 areas whose width and height are $1km$ in the New York city. Then, we set $q.loc$ as the center of the area. We will take the average value of recall in the 5 areas as the result.

5.2 Data Collection Methodology

In order to use the real dataset, a crawler system is developed for collecting the geotagged documents from a Web-based service. By using a $HTTP$ request, the crawler gets a geotagged document from the service. Then, the crawler filters out stop words which are not meaningful such as 'a', 'the', 'of'. From the remaining keywords, we generate the semantic annotation for the document using a semantic annotation system [7] and the Yago ontology. In here, the semantic annotation is a set of concepts in the Yago ontology which corresponds to the concepts appearing in the document.

6. RESULTS

6.1 Developement of an Efficient Search Method

We proposed the SR*-tree index, a ranking model, and the baseline of the top k closure search algorithm. Since the baseline algorithm uses only load the necessary parts of SR*-tree, which are related to the semantic concepts in SEM of the user query, we expect that the search space can be dramatically reduced from the beginning. Also, the algorithm progressively returns the complete closure for the physical score. Therefore, the method can be a pipelined approach. The current status of the research is developing pruning methods associated to the proposed baseline algorithm. Our goal is improving the efficiency of the search algorithm by filtering out the unnecessary search space. The pruning methods will utilize the spatial constraints and semantic constraints driven from the status of the priority queue.

6.2 Data Collection for Evaluation

For using the real data in the evaluation, we crawled the geotagged Web documents from a real location-based Web service, Flickr. In order to use English as the main language of the semantic concepts, we crawl the photos uploaded in the Newyork city. The number of crawled documents is about 120,000. In addition, for the crawled documents, we generate semantic annotations. The total number of semantic concepts in the semantic annotation of all the photos is about 1,250,000. The distinct number of semantic concepts in the data is about 9,000. We use the ontology concept hierarchy from Open Directory RDF Dump [3] and the YAGO ontology [12].

7. CONCLUSION AND FUTURE WORKS

In this paper, we address the spatial semantic search for improving the effectiveness of the Web-based service, and present an efficient method for the proposed search. Given a user query with geographical constraints and semantical constraints, and geotagged semantic annotated documents, the spatial semantic search finds the top k relevant sets of documents. In order for the effective search, we propose a hybrid index SR*-tree and an efficient search algorithm associated with the SR*-tree. The future works are as follows:

1. **(Improvement of Efficiency)** In order to improve the efficiency of our method, we aim to devise effective pruning techniques for the search method by using the spatial and semantic constraints. For the spatial constraints, the limitations of the maximum query distance and the diameter size described in the query are utilized. In addition, during updates of the priority queue, the current top k closures provide the upper-bounds of the physical score. For the semantic constraints, we can assign a range label for each concept. By using the query concepts, we can avoid irrelevant visits to tree or tree nodes. Furthermore, as the spatial constraints, the current top k closures in the priority queue also provide the upper-bounds of the sematic score.

2. **(Implementation)** For the evaluation of the proposed method, we will implement a naive method and the proposed method.

3. **(Comparison and Analysis)** By using the implemented systems and a real dataset, we will compare the two methods in terms of the efficiency and the accuracy and analyze experimental results.

8. ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea grant funded by the Korean government (MSIP) (No. NRF-2009-0081365).

9. REFERENCES

[1] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 373–384, New York, NY, USA, 2011. ACM.

[2] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *Proc. VLDB Endow.*, 2:337–348, August 2009.

[3] N. C. Corporation. Open directory rdf dump.

[4] I. De Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 656 –665, april 2008.

[5] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatial-keyword (sk) queries in geographic information retrieval (gir) systems. In *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*, SSDBM '07, pages 16–, Washington, DC, USA, 2007. IEEE Computer Society.

[6] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semant.*, 2:49–79, December 2004.

[7] Köhler,, Jacob and Philippi,, Stephan and Specht,, Michael and Rüegg,, Alexander. Ontology based text indexing and querying for the semantic web. *Knowledge Based System*, 19(8):744–754, December 2006.

[8] Z. Li, K. Lee, B. Zheng, W.-C. Lee, D. L. Lee, and X. Wang. Ir-tree: An efficient index for geographic document search. *IEEE Transactions on Knowledge and Data Engineering(TKDE)*, 23(4):585 –599, april 2011.

[9] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu. Collective spatial keyword queries: a distance owner-driven approach. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 689–700, New York, NY, USA, 2013. ACM.

[10] N. Mamoulis and D. Papadias. Multiway spatial joins. *ACM Trans. Database Syst.*, 26:424–475, December 2001.

[11] D. Papadias and D. Arkoumanis. Approximate processing of multiway spatial joins in very large databases. In *Proceedings of the 8th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '02, pages 179–196, London, UK, 2002. Springer-Verlag.

[12] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago - A Core of Semantic Knowledge. In *Proc. of the 16th international Conference on World Wide Web (WWW2007)*, pages 697–706, 2007.

[13] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, pages 688–699, Washington, DC, USA, 2009. IEEE Computer Society.

[14] D. Zhang, B. C. Ooi, and A. Tung. Locating mapped resources in web 2.0. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 521 –532, march 2010.

[15] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for location-based web search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 155–162, New York, NY, USA, 2005. ACM.