ISSN (Online): 2349-7084
GLOBAL IMPACT FACTOR 0.238
DIIF 0.876

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 640-645

A Novel Framework to Measure the Degree of Difficulty on Keyword Query Routing

[1] Kallem Rajender Reddy , [2]Y.Sunitha

[1]M.Tech (CS),Department of Computer Science & Engineering , Sri Indu Institute of Engineering & Technology, Sheriguda (v)Ibrahimpatnam(M), RR District,501510.
[2]Asst-professor,Department of Computer Science & Engineering , Sri Indu Institute of Engineering & Technology, Sheriguda (v)Ibrahimpatnam(M), RR District,501510
Email:kallemrajenderreddy@gmail.com,yadasunitha@gmail.com

Abstract:- Spatial queries, such as range search and nearest neighbor retrieval, involve only conditions on objects geometric properties. A spatial database manages multidimensional objects(such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria.Keyword queries on databases provide easy access to data, but often suffer from low ranking quality, i.e., low precision and/or recall, as shown in recent benchmarks. It would be useful to identify queries that are likely to have low ranking quality to improve the user satisfaction. For instance, the system may suggest to the user alternative queries for such hard queries. In this paper, we analyze the characteristics of hard queries and propose a novel framework to measure the degree of difficulty for a keyword query over a database, considering both the structure and the content of the database and the query results. We evaluate our query difficulty prediction model against two effectiveness benchmarks for popular keyword search ranking methods. we present a suite of optimizations to minimize the incurred time overhead.
.

Keywords: Keyword queries, KEYWORD query interfaces, INEX Queries, Semantic Search.

– – – – – – – – – – ◆ – – – – – – – – – –

I.INTRODUCTION

Spatial data mining is a special kind of data mining. The main difference between data mining and spatial data mining is that in spatial data mining tasks we use not only non-spatial attributes (as it is usual in data mining in non-spatial data), but also spatial attributes. Spatial data mining is the application of data mining methods to spatial data. The objective of spatial data mining is to find patterns in data with respect to geography. So far, data mining and Geographic Information Systems (GIS) have existed as two separate technologies, each with its own methods, traditions, and approaches to visualization and data analysis. The immense explosion in geographically referenced data occasioned by developments in IT, digital mapping, remote sensing, and the global diffusion of GIS emphasize the importance of developing data-driven inductive approaches to geographical analysis and modeling. Today, widely used of search engines has made it realistic to write spatial queries in a new way. Traditionally, queries focus on objects only geometric properties, for example whether a point is in rectangle or how two points are close from each other. Some new application allows users to browse objects based on both of their geometric coordinates and their associated texts. Such type of queries called as spatial keyword query. For example, if a search engine can be used to find nearest hotel that offer facilities such as pool and internet at the same time. From this query, we could first obtain the entire hotel whose services contain the set of keywords, and then find the nearest one from the retrieved restaurant. The major drawback of this approach is that, on the difficult input they do not provide real time answer. For example, from the query point the real neighbor lies quite far away, while all the closer neighbors are missing at least one of the query keywords. Spatial keyword queries have not been widely explored. In the past years, the group of people has showed interest in studying keyword search in to multidimensional data [5][6]. The best method for nearest neighbor search with keywords is because of Felipe et al. [5]. They combine the spatial index R-tree [7] and signature file [8]. So they developed a structure called IR2-tree. This tree has the ability of both R-tree and signature files. Like R-tree it stores the spatial proximity of object and like signature file it filters those objects that do not include all query keywords.

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 640-645

KEYWORD query interfaces (KQIs) for databases have attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Since any entity in a data set that contains the query keywords is a potential answer, keyword queries typically have many possible answers. KQIs must identify the information needs behind keyword queries and rank the answers so that the desired answers appear at the top of the list. Unless otherwise noted, we refer to keyword query as query in the remainder of this paper. Databases contain entities, and entities contain attributes that take attribute values. Some of the difficulties of answering a query are as follows: First, unlike queries in languages like SQL, users do not normally specify the desired schema element(s) for each query term.

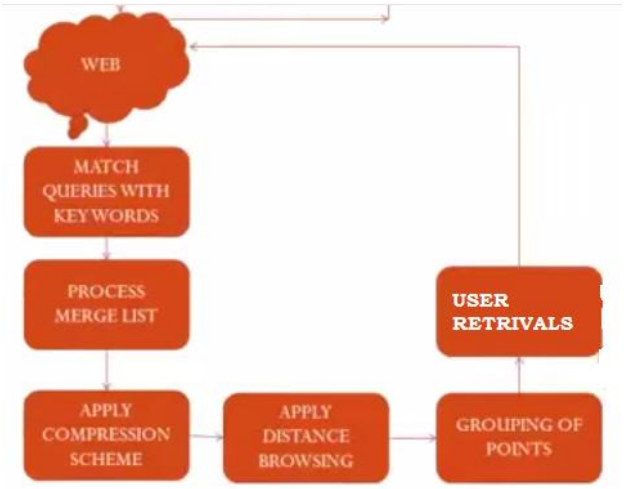

Fig 1. Query Keyword Processing Scenario

II.LITERATURE SURVEY

Cao et al. [1] proposed collective spatial keyword query, they presented the new problem of retrieving a group of spatial objects, and each associated with a set of keywords. They develop approximation algorithms with provable approximation bounds and exact algorithms to solve the two problems.

Lu et al. [2], combined the notion of keyword search with reverse nearest neighbor queries. They propose a hybrid index tree called IUR-tree Intersection-Union RTree) to answer the Reverse Spatial Textual k Nearest Neighbor (RSTkNN) query that effectively combines location proximity with textual similarity. They design a branch-and-bound search algorithm which is based on the IUR-tree. To further increase the query processing, they proposed an improved variant of the IUR-tree called cluster IUR-tree and two corresponding optimization algorithm

Zhang and Chee[3] introduced hybrid indexing structure bR*-tree, that combines the R*-tree and bitmap indexing to process the m-closest keyword query that returns the spatially closest objects matching m keywords. They utilized a priori based search strategy that successfully reduce the search space and also proposed two monotone constraints, distance mutex and keyword mutex to help effective pruning.

G. Cong, C.S. Jensen, and D. Wu [5] proposed an approach that computes the relevance between the documents of an object and a query. This relevance is then incorporated with the Euclidean distance between object and query to calculate an overall similarity of object to query.

Yufie Tao and Cheng Sheng [6], developed a new access method which is called as spatial inverted index. It extends the conventional inverted index to lay hold on multidimensional data, and uses the algorithms that can answer nearest neighbor queries with keywords in real time. They designed a variant of inverted index called spatial inverted index that is optimized for multidimensional points. This access method successfully includes point coordinates into a conventional inverted index with small space.

III.EXISTING SYSTEM:

There have been collaborative efforts to provide standard benchmarks and evaluation platforms for Fast Nearest Neighbor Search with Keywords over databases. One effort is the data-centric track of INEX Queries where each node has to match the whole querying keywords .It does not consider the density of data objects in the spatial space. Another effort is the series of Semantic Search Challenges (SemRetrival).The results indicate that even with structured data, finding the desired answers to keyword queries is still a hard task. More interestingly, looking closer to the ranking quality of the best performing methods on both the efforts.

WORKING MODEL OF PRESENTED SYSTEM:

KEYWORD query interfaces (KQIs) for databases have attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Since any entity in a data set that contains the query keywords is a potential answer, keyword queries typically have many possible answers. KQIs must identify the information needs behind keyword queries and rank the answers so that the desired answers appear at the top of the list. Unless otherwise noted, we refer to keyword query as query in the remainder of this paper. Databases contain entities, and entities contain attributes that take attribute values. Some of the difficulties of

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 640-645

answering a query are as follows: First, unlike queries in languages like SQL, users do not normally specify the desired schema element(s) for each query term. For instance, query Q1: Godfather on the IMDB database (http://www.imdb.com) does not specify if the user is interested in movies whose title is Godfather or movies distributed by the Godfather company. Thus, a KQI must find the desired attributes associated with each term in the query. Second, the schema of the output is not specified, i.e., users do not give enough information to single out exactly their desired entities. For example, Q1 may return movies or actors or producers[9]. Recently, there have been collaborative efforts to provide standard benchmarks and evaluation platforms for keyword search methods over databases. One effort is the data-centric track of INEX Workshop where KQIs are evaluated over the well-known IMDB data set that contains structured information about movies and people in show business. Queries were provided by participants of the workshop. Another effort is the series of Semantic Search Challenges (SemSearch) at Semantic Search Workshop, where the data set is the Billion Triple Challenge data set at http://vmlion25.deri.de. It is extracted from different structured data sources over the Web such as Wikipedia. The queries are taken from Yahoo! keyword query log. Users have provided relevance judgments for both benchmarks. The Mean Average Precision (MAP) of the best performing method(s) in the last data-centric track in INEX Workshop and Semantic Search Challenge for queries are about 0.36 and 0.2, respectively[10]. These results indicate that even with structured data, finding the desired answers to keyword queries is still a hard task. More interestingly, looking closer to the ranking quality of the best performing methods on both workshops, we notice that they all have been performing very poorly on a subset of queries. For instance, consider the query ancient Rome era over the IMDB data set. Users would like to see information about movies that talk about ancient Rome. For this query, the state-of the- art XML search methods which we implemented return rankings of considerably lower quality than their average ranking quality over all queries. Hence, some queries are more difficult than others. Moreover, no matter which ranking method is used, we cannot deliver a reasonable ranking for these queries. Such a trend has been also observed for keyword queries over text document collection[11].

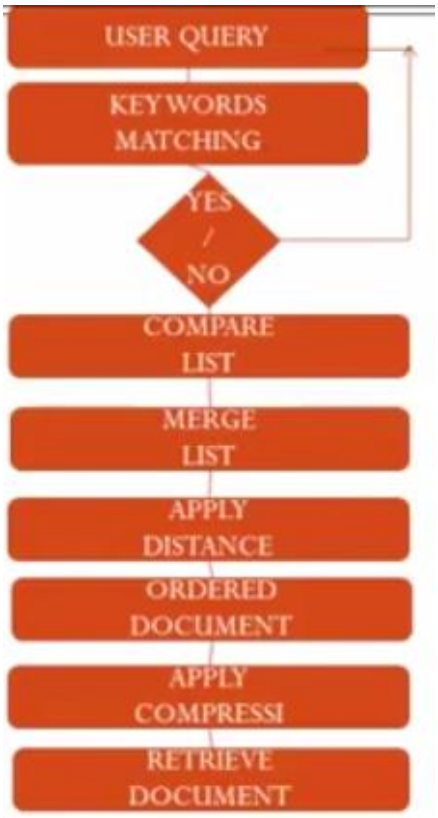

Fig 1. Flowcahart of Keyword Search

DISADVANTAGES OF EXISTING SYSTEM:

- IR2-tree is popular technique for indexing data but it having some drawbacks, which impacted on its efficiency. The disadvantage called as false hit affecting it seriously. The number of false positive ratio is large when the aim of the final result is far away from the query point and also when the result is simply empty. In these cases, the query algorithm will load the documents of many objects; as each loading necessitates a random access, it acquires costly overhead.
- Suffer from low ranking quality.
- Performing very poorly on a subset of queries.
- Failure of finding objects in space.

PROPOSED SYSTEM

- To overcome the problem of existing system we are implementing clustered based approach , Clustering is the process of making a group of abstract objects into classes of similar objects. We are using one of the clustered method is hierarchical method .
- This method creates a hierarchical decomposition of the given set of data objects. We can

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 640-645

classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here

- Agglomerative Approach
- Divisive Approach

AGGLOMERATIVE APPROACH: This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.In agglomerative clustering, the search for the nearest neighbor is repeated several times per iteration and every search requires O(N) merge cost calculations. The graph is utilized so that the search is limited only to the clusters that are directly connected by the graph structure. This reduces the time complexity of every search from O(N) to O(K). The parameter k affects the quality of the solution and the running time. If the number of neighbor's ðkÞ is small, significant speedup can be obtained.e set forth a principled framework and proposed novel algorithms to measure the degree of the difficulty of a query over a DB, using the ranking robustness principle. Based on our framework, we propose novel algorithms that efficiently predict the effectiveness of a keyword query.

ADVANTAGES:

- It can produce an ordering of the objects, which may be informative for data display.
- Smaller clusters are generated, which may be helpful for discovery.
- Easily mapped to both XML and relational data.
- Higher prediction accuracy and minimize the incurred time overhead.

PROPOSED SYSTEM FUNCTIONING

In this paper, we analyze the characteristics of difficult queries over databases and propose a novel method to detect such queries. We take advantage of the structure of the data to gain insight about the degree of the difficulty of a query given the database. We have implemented some of the most popular and representative algorithms for keyword search on databases and used them to evaluate our techniques on both the INEX and Research benchmarks. The results show that our method predicts the degree of the difficulty of a query efficiently and effectively. We make the following contributions:

• We introduce the problem of predicting the degree of the difficulty for queries over databases. We also analyze the reasons that make a query difficult to answer by KQIs .

• We propose the Structured Robustness (SR) score, which measures the difficulty of a query based on the differences between the rankings of the same query over the original and noisy (corrupted) versions of the same database, where the noise spans on both the content and the structure of the result entities [14].

• We present an algorithm to compute the SR score, and parameters to tune its performance.

• We introduce efficient approximate algorithms to estimate the SR score, given that such a measure is only useful when it can be computed with a small time overhead compared to the query execution time.

• We show the results of extensive experiments using two standard data sets and query workloads: INEX and SemSearch [15]. Our results show that the SR score effectively predicts the ranking quality of representative ranking algorithms, and outperforms non-trivial baselines, introduced in this paper.

IV.SYSTEM IMPLEMENTATIONS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

SYSTEM ARCHITECTURE

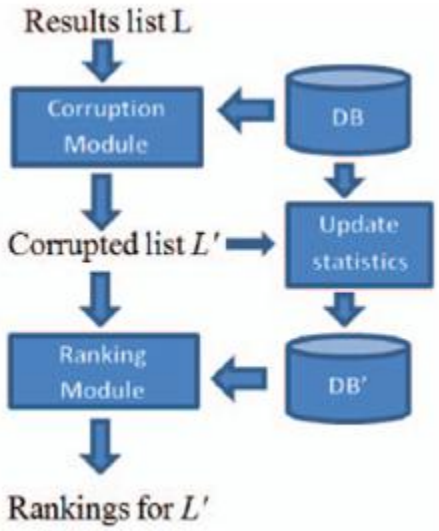

Fig 3.Praposed System Architecture

- Data and Query Modeling
- Ranking for Structured Data
- Corruption Module
- Ranking Module

DATA AND QUERY MODELING

In this Phase, first we develop a System Model for our proposed System. We model a database as a set of entity sets. Each entity set S is a collection of entities E. For instance, movies and people are two entity sets in IMDB.We ignore the physical representation of data in this paper. That is, an entity could be stored in an XML file or a set of normalized relational tables. The above model has been widely used in works on entity search and data-centric XML retrieval [8], and has the advantage that it can be easily mapped to both XML and relational data.

RANKING FOR STRUCTURED DATA

In this Phase we present the Ranking Robustness Principle, which argues that there is a (negative) correlation between the difficulty of a query and its ranking robustness in the presence of noise in the data.The degree of the difficulty of a query is positively correlated with the robustness of its ranking over the original and the corrupted versions of the collection. We call this observation the Ranking Robustness Principle.

CORRUPTION PHASE

The first challenge in using the Ranking Robustness Principle for databases is to define data corruption for structured data. For that, we model a database DB using a generative probabilistic model based on its building blocks, which are terms, attribute values, attributes, and entity sets.A corrupted version of DB can be seen as a random sample of such a probabilistic model.

RANKING PHASE

Each ranking algorithm uses some statistics about query terms or attributes values over the whole content of DB. Some examples of such statistics are the number of occurrences of a query term in all attributes values of the DB or total number of attribute values in each attribute and entity set. These global statistics are stored in M (metadata) and I (inverted indexes) in the SR Algorithm pseudocode.SR Algorithm generates the noise in the DB on-the-fly during query processing. Since it corrupts only the top K entities, which are anyways returned by the ranking module, it does not perform any extra I/O access to the DB, except to lookup some statistics. Moreover, it uses the information which is already computed and stored in inverted indexes and does not require any extra index.

V.CONCLUSSION:

In this paper, we analyze the characteristics of difficult queries over databases and propose a novel method to detect such queries. We take advantage of the structure of the data to gain insight about the degree of the difficulty of a query given the database. We have implemented some of the most popular and representative algorithms for keyword search on databases and used them to evaluate our techniques on both the INEX and SemSearch benchmarks. We introduced the novel problem of predicting the effectiveness of keyword queries over DBs. We showed that the current prediction methods for queries over unstructured data sources cannot be effectively used to solve this problem. We set forth a principled framework and proposed novel algorithms to measure the degree of the difficulty of a query over a DB, using the ranking robustness principle. Based on our framework, we propose novel algorithms that efficiently predict the effectiveness of a keyword query. Our extensive experiments show that the algorithms predict the difficulty of a query with relatively low errors and negligible time overheads.

REFERENCES:

[1] X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi, "Collective Spatial Keyword Querying," Proc. ACM SIGMOD Int'l Conf.
Management of Data, pp. 373-384, 2011.
[2] J. Lu, Y. Lu, and G. Cong, "Reverse Spatial and Textual k Nearest Neighbor Search," Proc. ACM SIGMOD Int'l Conf. Management
of Data, pp. 349-360, 2011.
[3] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa, "Keyword Search in Spatial Databases: Towards Searching by Document," Proc. Int'l Conf. Data Eng. (ICDE), pp.688-699, 2009.
[4] G. Cong, C.S. Jensen, and D. Wu, "Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects," PVLDB, vol. 2, no. 1, pp.
337- 348, 2009.
[5] I.D. Felipe, V. Hristidis, and N. Rishe, "Keyword Search on Spatial Databases," Proc. Int'l Conf. Data Eng. (ICDE), pp. 656-665, 2008.
[6] Yufei Tao and Cheng Sheng, "Fast Nearest Neighbor Search with Keywords", IEEE transactions on knowledge and data engineering,
VOL. 26, NO. 4, APRIL 2014.
[7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Da-

vies, "The case for vm-based cloudlets in mobile computing," IEEE Pervasive Computing, vol. 8, pp. 14–23, 2009.
[8] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in Proc. of IEEE INFOCOM, 2012.
[9] Z. Huang, C. Mei, L. E. Li, and T. Woo, "Cloudstream: Delivering high-quality streaming videos through a cloud-based svc proxy," in INFOCOM'11, 2011, pp. 201 205.
[10] T. Coppens, L. Trappeniners, and M. Godon, "AmigoTV: towards a social TV experience," in Proc. of EuroITV, 2004.
[11] N. Ducheneaut, R. J. Moore, L. Oehlberg, J. D. Thornton, and E. Nickell, "Social TV: Designing for Distributed, Sociable Television Viewing," International Journal of Human-Computer Interaction, vol. 24, no. 2, pp. 136–154, 2008.
[12] A. Carroll and G. Heiser, "An analysis of power consumption in as smartphone," in Proc. of USENIX-ATC, 2010.
[13] What is 100% Pure Java, http://www.javacoffeebreak.com/faq/faq0006.html.
[14] J. Santos, D. Gomes, S. Sargento, R. L. Aguiar, N. Baker, M. Zafar, and A. Ikram, "Multicast/broadcast network convergence in next generation mobile networks," Comput. Netw., vol. 52, pp. 228–247, January 2008.
[15] DVB-H, http://www.dvb-h.org/.
[16] K. Chorianopoulos and G. Lekakos, "Introduction to social tv: Enhancing the shared experience with interactive tv," International Journal of Human- Computer Interaction, vol. 24, no. 2, pp. 113–120, 2008.