International Journal of Science Engineering and Advance Technology, IJSEAT, Vol 3, Issue 10
ISSN 2321-6905
October 2015

IJSEAT International Journal of Science Engineering and Advance Technology

Explicit Uncertainty word directing through Data Mining

Anguluri Anilkumar[1], Prashant Byravarapu[2]

#1. Student of M.Tech(CSE), Department of Computer Science and Engineering, Eluru College of Engineering & Technology, West Godavari dist., Eluru, AP

#2. Assosc.Professor, Department of Computer Science and Engineering, Eluru College of Engineering & Technology, West Godavari dist., Eluru, AP, INDIA.

Abstract:

Searching a keyword on an enormous a colossal is somewhat easier, however the search over a enlarge range of structured and connected information creates a problem. Routing keywords solely to applicable sources will scale back the high value of looking of queries over all sources. It's tough for net user to use this net information by means that of SQL or SPARQL. we tend to rent a keyword component relationship outline that succinctly represents relationships between keywords and also the information parts referred to as the set-level keyword-element relationship graph (KERG).A structure rating mechanism is recommended for computing the relevant of routing plans supported the extent of keyword, information parts , component sets and sub graphs that connect these parts. The web may be a not operation it's solely provides a link for looking the online document supported the keyword. The question may be shaped from keywords that square measure wont to retrieve the document. It's tough for the standard net users to take advantage of this net information by means that of structured queries exploitation languages like SQL or SPARQL. In info analysis, most of the approaches use solely the only supply solutions. The most issue here is computing the foremost relevant mixtures of sources. To route keywords solely to relevant sources, a completely unique methodology is projected for computing top-k routing plans supported their keyword question. The keyword-element relationship outline is employed to represents the relationships between keywords and also the information parts. Structure rating mechanism is projected for computing the connection of routing plans supported scores at the extent of keywords and information parts. It's no data regarding the command language and it as hostile structured queries. That the schema or the underlying information is required.

Index Terms—Keyword search, keyword query, keyword query routing, graph-structured data, RDF

I.INTRODUCTION:

A search question may be a question that a user enters into a program to satisfy their info desires. These queries square measure distinctive. There square measure 3 broad classes like Informational queries, steering queries and Transactional queries. There square measure totally different styles of links may be established for various queries. The foremost relevant queries square measure retrieved supported the keyword query; i.e., selects the only most relevant databases. The most issue here is to reason the foremost relevant mixtures of sources from the info. The goal is to provide routing plans, which might be wont to reason results from multiple sources. We tend to square measure focusing to the matter of keyword question routing over an outsized range of knowledge sources. Routing keywords solely to relevant sources will scale back the high value of looking for structured results that extent multiple sources. Relationships square measure delineate between keywords and/or information parts. They are created for the whole assortment of connected sources, and so classified as parts referred to as the set-level keyword-element relationship graph (KERG). To include connection at the extent of keywords, the IR-style ranking methodology has been projected. The second family of unstructured p2p networks contains Gnutella-like networks that don't impose any structure on the overlay network [6]. The default search mechanism in Gnutella is to blindly forward queries to all or any neighbors at intervals a precise range of hops. Though this mechanism handles network dynamics o.k., search through blind flooding is sort of inefficient. This has driven a number of studies proposing varied enhancements to look in unstructured networks. Major enhancements embody replacement the blind flooding with a random-walk [7] or associate degree increasing ring search, craft the network construction to attain properties of little world graphs [8], reflective the capacities of heterogeneous nodes in topology-construction, and caching tips to content settled one hop away. All of those proposals (except for caching) retain the "blind" nature of question forwarding in Gnutella. In alternative words, the forwarding of queries is freelance of the question string and doesn't exploit the knowledge contained within the question itself. The keywords within the question square measure used just for looking the native content index. The target of this work is to style associate degree economical query routing mechanism for unstructured peer-to-peer networks. We tend to propose to create probabilistic routing tables at nodes, created associate degreed maintained through an exchange of updates among immediate neighbors within the overlay. These routing tables use a completely unique arrangement — the exponential return Bloom Filter (EDBF) — to with efficiency store and propagate probabilistic info regarding content hosted within the neighborhood of a node. The number of data in associate degree EDBF (and the quantity of bits wont to store this information) decreases exponentially with distance. Such exponential decrease in info with distance restricts the impact of network dynamics to the neighborhood of any outward or fresh inward node. The ascendable question Routing (SQR) mechanism we tend to style uses hints obtained from these probabilistic routing tables to forward queries. The employment of probabilistic hints provides a major advantage over the fully blind nature of existing mechanisms, translating into giant reductions within the average range of hops over that a question is forwarded before it's answered.

II. RELATED WORK

There are two directions of work

1. Keyword search approaches compute the most relevant structured results.

2. Solutions for source selection compute the most relevant sources.

2.1 Keyword Search There are two main categories:

2.1.1 Schema Based Approaches There are schema-based approaches implemented on top of off-the-shelf databases. A keyword query is processed by mapping keywords to elements of the database (called keyword elements). Then, using the schema, valid join sequences are derived, which are then employed to join (“connect”) the computed keyword elements to form so-called candidate networks representing possible results to the keyword query. » Effective Keyword Search in Relational Databases [10] In relational databases, we have three key steps for processing a given keyword query. (1) Generate all candidate answers, each of which is a tuple tree by joining tuples from multiple tables. (2) Then compute a single score for each answer. The scores should be defined in such a way so that the most relevant answers are ranked as high as possible. (3) And finally return answers with semantics. DBXplorer [16], DISCOVER [15], BANKS [18], and Hristidis et al. are systems that support keyword search on relational databases. For the first step, they generate tuple trees from multiple tables as answers. The first three systems (DBXplorer [16], DISCOVER [15], BANKS [18]) require an answer containing all keywords in a query, while the last one only requires an answer containing some but not necessarily all keywords in the query. Efficiency has been the focus for the first step: rules are designed to avoid generation of unnecessary tuple trees, and more efficient algorithms are proposed to improve the time and space complexities. For the second step, the first two systems use a very simple ranking strategy: the answers are ranked in ascending order of the number of joins involved in the tuple trees. When two tuple trees have the same number of joins, their ranks are determined arbitrarily. Thus, all tuple trees consisting of a single tuple are ranked ahead of all tuples trees with joins. The ranking strategy of the BANKS system is to combine two types of information in a tuple tree to compute a score for ranking: a weight (similar to PageRank for web pages) of each tuple, and a weight of each edge in the tuple tree that measures how related the two tuples are. The strategy of DBXplorer and DISCOVER and the strategy of BANKS for the second step do not utilize any state-of-the-art IR ranking methods, which have been tremendously successful. A state-of-the-art IR ranking method is used to compute a score between a given query and each text column value in the tuple tree. A final score is obtained by dividing the sum of all these scores by the number of tuples (i.e. the number of joins plus 1) in the tree. However, they only concentrate on the efficiency issue of the implementation of the ranking strategy and do not conduct any experiments on the effectiveness issue. This paper focuses on search effectiveness. 2.1.2 Schema-Agnostic Approaches Systems for “schema-agnostic” keyword search on databases, such as DBXplorer [16], BANKS [18] and Discover [15], model a response as a tree connecting nodes (tuples) that contain the different keywords in a query (or more generally, nodes that satisfy specified conditions). Here “schema-agnostic” means that the queries need not use any schema information (although the evaluation system can exploit schema information). For example, the query “Gray transaction” on a graph derived from DBLP may find Gray matching an author node, transaction matching a paper node, and an answer would be the connecting path; with more than two keywords, the answer would be a, connecting tree. The tree model has also been used to find connected Web pages that together contain the keywords in a query. Schema-agnostic approaches [3],

[4], [8], [13] operate directly on the data. Structured results are computed by exploring the underlying data graph. The goal is to find structures in the data called Steiner trees (Steiner graphs in general), which connect keyword elements [8]. » EASE: An Effective 3-In-1 Keyword Search Method For Unstructured, Semi-Structured And Structured Data[8] In this paper, they propose an efficient and adaptive keyword search method, called EASE, for indexing and querying large collections of heterogeneous data. To achieve high efficiency in processing keyword queries, we first model unstructured, semi structured and structured data as graphs, and then summarize the graphs and construct graph indices instead of using traditional inverted indices. They propose an extended inverted index to facilitate keyword-based search, and present a novel ranking mechanism for enhancing search effectiveness. They have conducted an extensive experimental study using real datasets, and the results show that EASE achieves both high search efficiency and high accuracy, and outperforms the existing approaches significantly. » BLINKS: Ranked Keyword Searches on Graphs[4] A top-k keyword search query on a graph finds the top k answers according to some ranking criteria, where each answer is a substructure of the graph containing all query keywords. BLINKS, a bi-level indexing and query processing scheme for top-k keyword search on graphs. BLINKS follow a search strategy with provable performance bounds, while additionally exploiting a bi-level index for pruning and accelerating the search. To reduce the index space, BLINKS partitions a data graph into blocks: The bi-level index stores summary information at the block level to initiate and guide search among blocks, and more detailed information for each block to accelerate search within blocks. Their main contributions are the following: 1. Better search strategy. 2. Combining indexing with search. 3. Partitioning-based indexing.

2.2 Database Selection The wide popularity of free-and-easy keyword based searches over World Wide Web has fueled the demand for incorporating keyword-based search over structured databases. However, most of the research work focuses on keyword-based (2.1) searching over a single structured data source. With the growing interest in distributed databases and service oriented architecture over the Internet, it is important to extend such a capability over multiple structured data sources. One of the most important problems for enabling such a query facility is to be able to select the most useful data sources relevant to the keyword query. More closely related to this work existing solutions to database selection, where the goal is to identify the most relevant databases. The main idea is based on modeling databases using keyword relationships. A keyword relationship is a pair of keywords that can be connected via a sequence of join operations. For instance, is a keyword relationship as there is a path between uni1 and prize in Fig. 1. A database is relevant if its keyword relationship model covers all pairs of query keywords. MKS [1] captures relationships using a matrix. Since M-KS considers only binary relationships between keywords, it incurs a large number of false positives for queries with more than two keywords. This is the case when all query keywords are pairwise related but there is no combined join sequence which connects all of them. G-KS [2] addresses this problem by considering more complex relationships between keywords using a keyword relationship graph (KRG). Each node in the graph corresponds to a keyword. Each edge between two nodes corresponding to the keywords indicates that there exists at least two connected tuples ti →tj that match ki and kj. Moreover, the distance between ti and tj are marked on the edges.

III. PROBLEM DEFINITION

Until now, keyword searching is done only in certain graph database but in real application, there is uncertain graph data. However, so far, there is no work on keyword search in uncertain graph data. For keyword searching in uncertain graph database, two phases were used which are filtering and verification. For filtering purpose, there were also sub phases which are existence probabilistic prune, path based probabilistic prune and tee based probabilistic phase which consumed more time for filtering and finally verification is applied. This procedure consumed much more time so it is necessary to reduce processing time for that a new approach can be used which will also reduce the high cost of processing keyword search query over uncertain graph data. This approach greatly helps to improve the performance of keyword search, without compromising its result quality.

IV. Block diagram of the proposed system:

This paper propose to route keywords only to relevant sources to reduce the high cost of processing keyword search queries over all sources. It propose a novel method for computing top-k routing plans based on their potentials to contain results for a given keyword query. It employs a keyword-element relationship summary that compactly represents relationships between keywords and the data elements mentioning them. A multilevel scoring mechanism is proposed for computing the relevance of routing plans based on scores at the level of keywords, data elements, element sets, and sub graphs that connect these elements. Based on modeling the search space as a multilevel inter-relationship graph, it proposed a summary model that groups keyword and element relationships at the level of sets, and developed a multilevel ranking scheme to incorporate relevance at different dimensions.

It reduce the high cost of processing keyword search queries over all sources. It improves the performance of keyword search.

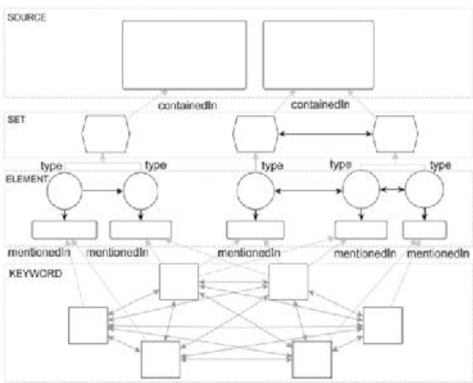

Figure: Block diagram of the proposed system

V.QUERY EXPANSION USING LINGUISTIC AND SEMANTIC FEATURES:

In document retrieval, many query expansion techniques are based on information contained in the top-ranked retrieved documents. The linguistic features are extracted from Word Net.

The features are:

Synonyms: words having similar meaningsto the input keyword k.

Hyponyms: words representing a specialization of the input keyword k.

Hyponyms: words representing ageneralization of the input keyword k.

These semantic features are defined as the following semantic relations:

sameAs: deriving resources having the same identity as the input resource using owl:sameAs.

seeAlso: deriving resources that provide more information about the input resource using rdfs:seeAlso.

Class/property equivalence: deriving classesor properties providing related descriptions for the input resource using owl:equivalentClass and owl:equivalentProperty.

superclass/-property: deriving all super classes/properties of the input resource by following the rdfs:subClassOf or rdfs:subPropertyOf property paths originating from the input resource.

subclass/-property: deriving all sub resources of the input resource ri by following the rdfs:subClassOf or rdfs:subPropertyOf property paths ending with the input resource. broader

concepts: deriving broader concepts related to the input resource ri using the SKOS vocabulary properties skos:broader and skos:broadMatch. narrower concepts: deriving narrowerconcepts related to the input resource ri using skos:narrower and skos:narrowMatch. related concepts: deriving related conceptsto the input resource ri using skos:closeMatch, skos:mappingRelation and skos:exactMatch.

The following preprocessing methods are involved here:

1) Tokenization: extraction of individual words, ignoring punctuation and case.

2) Stop word removal: removal of common words such as articles and prepositions.

3) Word lemmatization: determining the lemma of the word. Based on the elements and sets of elements in which they occur, the keyword-element relationships are created. Pre-computing relationships between data elements are typically performed for keyword search to improve the performance. These relationships are stored in specialized indexes and retrieved at the time of keyword query processing to accelerate the search for Steiner graphs. They are represented as keywordelement relationships.

VI. COMPUTING ROUTING PLANS:

Routing plans are computed by searching for Steiner graphs a routing graph contains a set of data sources and it contains information that enables the user to assess whether it is relevant: i.e., a plan is relevant only if the nodes mentioning the keywords and relationships between them correspond to the intended information need. This additional information will be used in the evaluation to assess the effectiveness of ranking.

Basically, the computation can be divided into three stages:

1. Computation of routing graphs,

2. Aggregation of routing graphs, and

3. Ranking query routing plans. The procedure for computing routing plans is described in the given Algorithm:

Algorithm :PPRJ: ComputeRoutingPlan(K, Wk)Input: The query K, the summary Wk(Nk, Ek)

Output: Set of routing plans [RP]

JP <- a join plan that contains all (ki, kj) 2k;

T <- a table where every tuple captures a joinsequence of KERG relationships e'k, and thecombined score of the join sequence; it is initiallyempty;

While – JP.empty() do

(ki ,kj) – JP.pop() ;

ἐ (ki . kj) retrieve(ἐk , (ki , kj));

if T , empty() then

T ἐ (ki ,kj));

else

T ἐ (ki ,kj) ∞ T ;

Compute scores of tuples in T via

SCORE(k, W'ks);

[RP] Group T by sources to identify unique

Combination of sources;

Compute score of routing plans in [RP] via

SCORE(K, RP);

Sort [RP] by score;

VII. CONCLUSION:

The keyword query routing is developed for a solution to the novel problem. The summary model is proposed based on modeling the search space as a multilevel inter-relationship graph, which groups keyword and element relationships at the level of sets. And the multilevel ranking scheme is developed to incorporate relevance at different dimensions. Keyword query search is a widely used approach for retrieving linked data in an efficient manner. In order to reduce the high cost of searching the keywords are redirected to the relevant data sources. When routing is applied to an existing keyword search system, the performance gain can be achieved. In this paper we have given different keyword search techniques and database selection techniques. Keyword search categorized into schema-based approaches and schema-agnostic approaches. Keyword search approaches computes the most relevant structured results.

REFERENCES

[1] V. Hristidis, L. Gravano, and Y. apakonstantinou, "Efficient IR-Style Keyword Search over Relational Databases," Proc. 29th Int'l Conf. Very Large Data Bases (VLDB), pp. 850-861, 2003.

[2] F. Liu, C.T. Yu, W. Meng, and A. Chowdhury, "Effective Keyword Search in Relational Databases," Proc. ACM SIGMOD Conf., pp. 563-574, 2006.

[3] Y. Luo, X. Lin, W. Wang, and X. Zhou, "Spark: Top-K Keyword Query in Relational Databases," Proc. ACM SIGMOD Conf., pp. 115-126, 2007.

[4] M. Sayyadian, H. LeKhac, A. Doan, and L. Gravano, "Efficient Keyword Search Across Heterogeneous Relational Databases," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 346-355, 2007.

[5] B. Ding, J.X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin, "Finding Top-K MinCost Connected Trees in Databases," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 836-845, 2007.

[6] B. Yu, G. Li, K.R. Sollins, and A.K.H. Tung, "Effective Keyword- Based Selection of Relational Databases," Proc. ACM SIGMOD Conf., pp. 139-150, 2007.

[7] Q.H. Vu, B.C. Ooi, D. Papadias, and A.K.H. Tung, "A Graph Method for KeywordBased Selection of the Top-K Databases," Proc. ACM SIGMOD Conf., pp. 915-926, 2008.

[8] V. Hristidis and Y. Papakonstantinou, "Discover: Keyword Search in Relational Databases," Proc. 28th Int'l Conf. Very Large Data Bases (VLDB), pp. 670-681, 2002.

[9] L. Qin, J.X. Yu, and L. Chang, "Keyword Search in Databases: The Power of RDBMS," Proc. ACM SIGMOD Conf., pp. 681-694, 2009.

[10] G. Li, S. Ji, C. Li, and J. Feng, "Efficient Type-Ahead Search on Relational Data: A Tastier Approach," Proc. ACM SIGMOD Conf., pp. 695-706, 2009.