ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A SURVEY ON KEYWORD QUERY ROUTING IN DATABASES

N.Saranya*, R.Rajeshkumar, S.Saranya
Department of Computer Science and Engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India

ABSTRACT

The web is a no longer operation it is only provides a link for searching the web document based on the keyword. The query can be formed from keywords which are used to retrieve the document. It is difficult for the typical web users to exploit this web data by means of structured queries using languages like SQL or SPARQL. In database research, most of the approaches use only the single source solutions. The main issue here is computing the most relevant combinations of sources. To route keywords only to relevant sources, a novel method is proposed for computing top-k routing plans based on their keyword query. The keyword-element relationship summary is used to represents the relationships between keywords and the data elements. Multilevel scoring mechanism is proposed for computing the relevance of routing plans based on scores at the level of keywords and data elements. It has no knowledge about the query language and it as opposed to structured queries. So the schema or the underlying data is needed.

KEYWORDS: Keyword search, keyword query, keyword query routing, graph-structured data, RDF.

INTRODUCTION

A Web search query is a query that a user enters into a web search engine to satisfy their information needs. These queries are distinctive. There are three broad categories such as Informational queries, Navigational queries and Transactional queries. There are different kinds of links can be established for different queries. The most relevant queries are retrieved based on the keyword query; i.e., selects the single most relevant databases. The main issue here is to compute the most relevant combinations of sources from the database. The goal is to produce routing plans, which can be used to compute results from multiple sources. We are focusing to the problem of keyword query routing over a large number of data sources. Routing keywords only to relevant sources can reduce the high cost of searching for structured results that extent multiple sources. Relationships are represented between keywords and/or data elements.They are constructed for the entire collection of linked sources, and then grouped as elements called the set-level keyword-element relationship graph (KERG).

To incorporate relevance at the level of keywords, the IR-style ranking method has been proposed. A multilevel relevance model is employed in this method, where elements are considered as key words, entities mentioning these keywords, corresponding sets of entities, relationships between elements of the same level, and inter-relationships between elements of different levels.

Keyword Query Search can be divided into two directions of work. They are: 1) keyword search approaches compute the most relevant structured results and 2) Solutions for source selection compute the most relevant sources.

KEYWORD SEARCH

There are two approaches can be used for keyword searching.

1. Schema based approaches
2. Schema-agnostic approaches

The schema based approaches are top of off-the-shelf databases. In this approach the keyword query is processed by mapping keywords to the elements of database which is referred to as keyword elements. The valid join sequences are derived, to compute keyword elements to form the keyword query.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

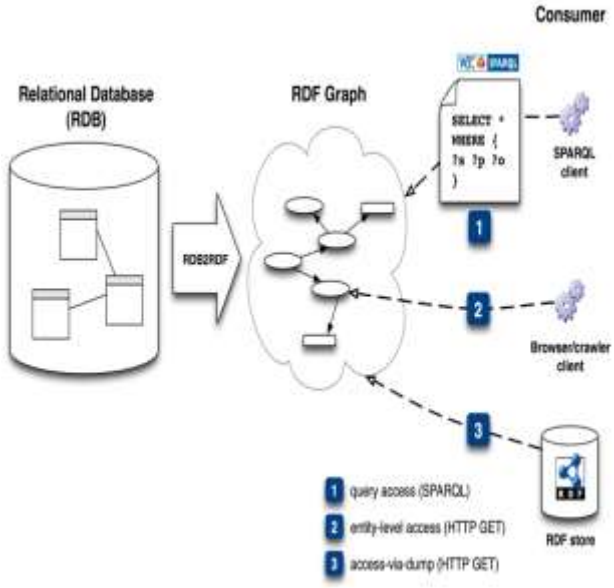

Fig. 1. Extract of the web data graph

Schema-agnostic approaches operate directly on the data. The main goal of this approach is to find structures in the data called Steiner trees which connect keyword elements. A Steiner graph is the path between uni1 and prize1 in Fig. 1. Various kinds of algorithms have been proposed for the efficient exploration of keyword search results over data graphs, which might be very large.

The schema-based techniques are used to find candidate networks in the multisource setting. It employs schema matching techniques to discover links between sources and uses structure discovery techniques to find foreign-key joins across sources.

DATABASE SELECTION

The main goal of the database selection is to identify the most relevant databases. For this the main idea is based on modelling databases with keyword relationships. A database is relevant if its keyword relationship model covers all pairs of query keywords.

An element-level data graph consists of:

1. The set of nodes N, which is the disjoint union of NE]NV, where the nodes NE represent entities and the nodes NV capture entities' attribute values, and .
2. The set of edges E, subdivided by E ¼ ER] EA, where ER represents interentity relations, EA stands for entity-attribute assignments. We have eðn1; n2Þ 2 ER iff n1; n2 2 NE and eðn1; n2Þ 2 EA iff n1 2 NE and n2 2 NV.

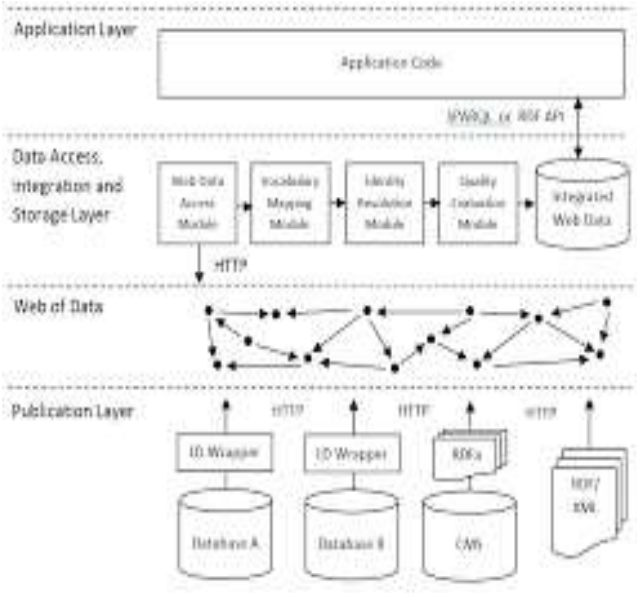

Fig. 2. Set-level web data graph

The set-level graph essentially captures a part of the Linked Data schema on the web that are represented as relations between classes. Often, a schema might be incomplete or simply does not exist for RDF data on the web. A pseudoschema can be obtained by computing a structural summary. A set-level data graph can be derived from a given schema or a generated pseudoschema.

KEYWORD QUERY ROUTING

The keyword query is used to find the result from data sources. The results may contain data from several sources.

All keyword search approaches is the pragmatic assumption that users are only interested in compact results. The problem of keyword query routing is to find the top-k keyword routing plans based on their relevance to a query. A relevant plan should correspond to the information need as intended by the user.

The search space of keyword query routing using a multilevel inter-relationship graph. At the lowest level, it models relationships between keywords. The inter-relationships between elements at different levels are shown in Fig. 3. A keyword is mentioned in some entity descriptions at the element level. Entities at the element level are associated with a set-level element based on the type. A set-level element is contained in a source. There is an edge between

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

two keywords iff two elements at the element level mentioning these keywords are connected based on a path. The ranking schema is proposed here based on the graph.

The keyword search relies on an element-level model to compute keyword query results. Elements mentioning keywords are retrieved and paths between them are explored to compute graphs. To deal with the keyword routing problem, the elements can be stored along with the sources. So the information can be retrieved to derive the routing plans from the computed keyword query results. And the data graph is expensive for exploring the paths between the keywords.

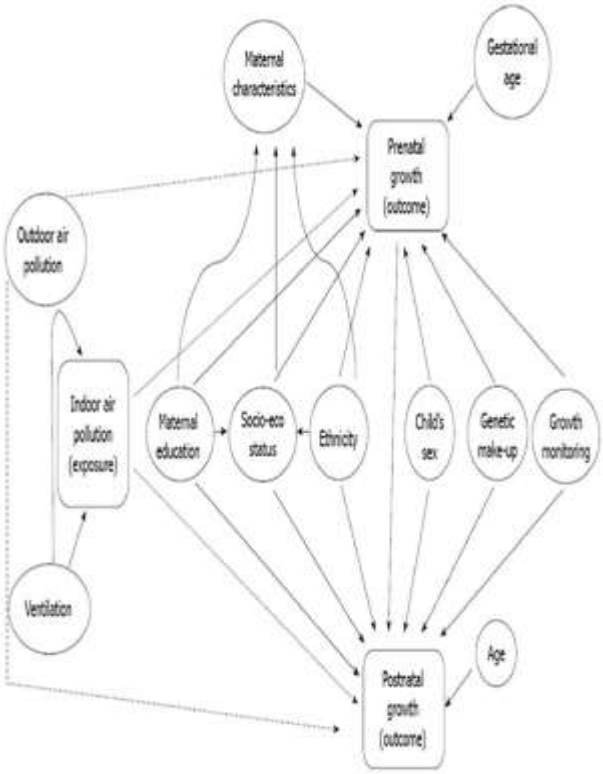

Fig. 3. Multilevel Inter-relationship graph

KRG is used to captures relationships at the level of keyword. The relationships between keywords are captured by KRG are not direct edges between tuples. KRG relationships are retrieved for all pairs of query keywords in database selection. But it is not possible to guarantee that it is not a Steiner graph, so the sub graphs should be validated. This is called as the filtering process which makes use of all the information about the keywords. The main goal here is to ensure that not only keywords but also tuples mentioning them are connected. This approach relies on information at the element level.

KRG focuses on database selection; it only needs to know whether two keywords are connected by some join sequences. This information is stored as relationships in the KRG and can be retrieved directly. The paths between data elements are retrieved and explored in keyword search. Retrieving and exploring paths that might be composed of several edges are clearly more expensive than retrieving relationships between keywords. A multisource KRG models both relationships between sources. Keyword relationships are stored together with the elements are associated with source information. An element-level keyword-element relationship graph (E-KERG).

APPROACHES FOR KEYWORD QUERY ROUTING

There are four different approaches for keyword query routing.

1. Keyword level model
2. Element level model
3. Set level model and
4. Query expansion using linguistic and semantic features.

KEYWORD LEVEL MODEL

In keyword level, the relationship can be represented using *Keyword Relationship Graph* (KRG). It captures relationships at the keyword level. The relationships captured by a KRG are not direct edges between tuples but stand for paths between keywords. Keyword search over relational databases finds the answers of tuples in the databases which are connected through primary/foreign keys and contain query keywords. A *tuple unit* is a set of highly relevant tuples which contain query keywords.

ELEMENT LEVEL MODEL

An element-level model is used to compute keyword query results. Elements mentioning keywords are retrieved from this model and paths between them are explored to compute Steiner graphs. To characterize the individual data models the graph-based data models. A tuple in a relational database can be modeled as an entity, and foreign key relationships can be represented as inter entity relations. The data graph and the number of keyword elements are possibly very large in our scenario, and thus,

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

exploring all paths between them in the data graphs is expensive. This is the main drawback of this model.

SET LEVEL MODEL

The set-level graph essentially captures a part of the Linked Data schema on the web that is represented in RDFS, i.e., relations between classes. A schema might be incomplete or simply does not exist for RDF data on the web. A set-level data graph can be derived from a given schema or a generated pseudo schema.

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 WordNet. The features are:

- *Synonyms:* words having similar meanings to the input keyword *k*.
- *Hyponyms:* words representing a specialization of the input keyword *k*.
- *Hyponyms:* words representing a generalization 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 classes or 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 narrower concepts related to the input resource *ri* using skos:narrower and skos:narrowMatch.
- *related concepts:* deriving related concepts to 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 lemmatisation:* 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 keyword-element relationships.

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 1: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 join sequence of KERG relationships e'_k, and the combined score of the join sequence; it is initially empty;
While – JP.empty() do
(k_i, k_j) – JP.pop() ;

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

$\dot{\varepsilon}_{(ki\,.\,kj)}$ retrieve($\dot{\varepsilon}_k$, (k_i , k_j));
if T , empty() then
T $\dot{\varepsilon}_{(ki\,,\,kj))}$;
else
T $\dot{\varepsilon}_{(ki\,,\,kj)} \infty$ T ;
Compute scores of tuples in T via
SCORE(k, $W'_k{}^s$);
[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;

CONCLUSION

The keyword query routing is developed for a solution to the novel problem. The summary model is proposed based on modelling 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.

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 Min-Cost 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 Keyword-Based 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.