A Study on Specified Keyword Mining

A.Sheeba
Research scholar
Department of Computer Science
Vels University, Chennai, TamilNadu

K.Rohini
Assistant Professor
Department of Computer Science
Vels University, Chennai, TamilNadu

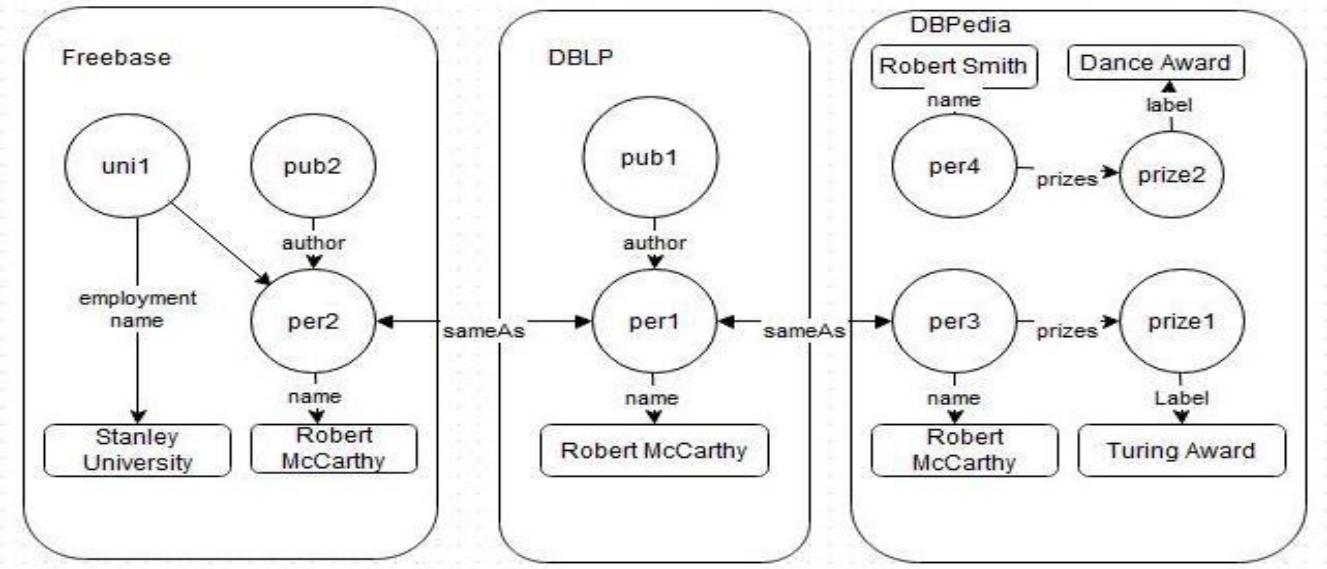
Abstract- Keyword search is an intuitive paradigm for searching linked data sources on the web. We propose to route keywords only to relevant sources to reduce the high cost of processing keyword search queries over all sources. We propose a novel method for computing top-k routing plans based on their potentials to contain results for a given keyword query. We employ 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. Experiments carried out using 150 publicly available sources on the web showed that valid plan (precision@1 of 0.92) that are highly relevant (mean reciprocal rank of 0.89) can be computed in 1 second on average on a single PC.

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

I. INTRODUCTION

The web is no longer only a collection of textual documents but also a web of interlinked data source (e.g., Linked Data). One prominent project that largely contributes to this development is Linking Open Data. Through this project, a large amount of legacy data have been transformed to RDF, linked with other sources, and published as Linked Data. Collectively, Linked Data comprise hundreds of sources containing billions of RDF triples, which are connected by millions of links. While different kinds of links can be established, the ones frequently published are same as links, which denote that two RDF resources represent the same real-world object. It is difficult for the typical web users to exploit this web data by means of structured queries using languages like SQL or SPARQL. To this end, keyword search has proven to be intuitive. As opposed to structured queries, no knowledge of the query language, the schema or the underlying data are needed. In database research, solutions have been proposed, which given a keyword query, retrieve the most relevant structured results or simply, select the single most relevant databases. However, these approaches are single-source solutions. They are not directly applicable to the web of Linked Data, where results are not bounded by a single source but might encompass several Linked Data sources. As opposed to the source selection problem, which is focusing on computing the most relevant sources, the problem here is to compute the most relevant combinations of sources. The goal is to produce routing plans, which can be used to compute results from multiple sources.

We propose to investigate the problem of keyword Mining for keyword search over a large number of structured and Linked Data sources. Routing keywords only to relevant sources can reduce the high cost of searching for structured results that span multiple sources. To the best of our knowledge, the work presented in this paper represents the first attempt to address this problem. Existing work uses keyword relationships (KR) collected individually for single databases. We represent relationships between keywords as well as those between data elements. They are constructed for the entire collection of linked sources, and then grouped as elements of a compact summary called the set-level keyword-element relationship graph (KERG). Summarizing relationships is essential for addressing the scalability requirement of the Linked Data web scenario.



.

Figure 1: Example of Linked data on web

For selecting the correct routing plan, we use graphs that are developed based on the relationships between the keywords present in the keyword query. This relationship is considered at the various levels such as keyword level, element level, set level e.t.c.,

II.PROPOSED ALGORITHM

There are two directions of work:

a) keyword search- approaches compute the most relevant structured results.
b) Database selection- solutions for source selection compute the most relevant sources.

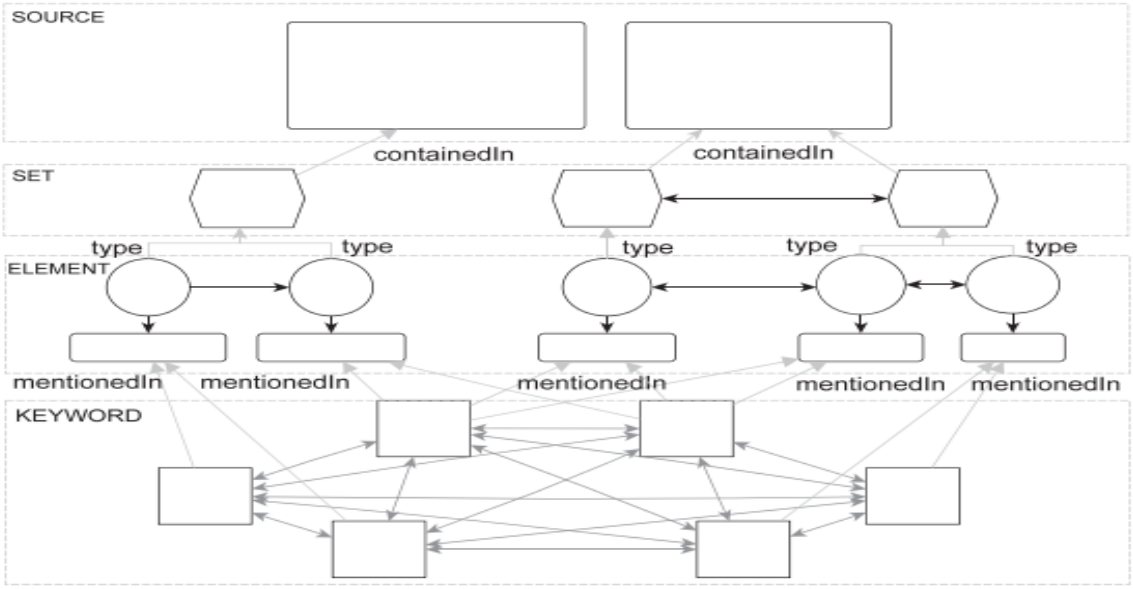


Fig.2.1 Multilevel Inter-relationship graph.

a)Keyword search

In the keyword searching, we mainly follow two approaches. They are *schema-based approaches* and *schema-agnostic approaches*.

Schema-based approaches are implemented on top of off-the-shelf databases. A keyword is processed by mapping keywords to the elements of the databases, called *keyword elements*. Then, using the schema, valid join sequences are derived and are employed to join the computed keyword elements to form the candidate-networks that represent the possible results to the keyword query.

Schema-agnostic approaches operate directly on the data. By exploring the underlying graphs the structured results are computed in these approaches. Keywords and elements which are connected are represented using Steiner trees/graphs. The goal of this approach is to find structures in the Steiner trees. For the query "Stanley Robert Award" for instance, 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. Recently, a system called Kite extends schema-based techniques to find candidate networks in the multi source setting. It employs schema matching techniques to discover links between sources and uses structure discovery techniques to find foreign-key joins across sources. Also based on pre computed links, Hermes translates keywords to structured queries.

b)Database Selection

In order to get the efficient results for keyword search, the selection of the relevant data sources plays a major role. 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, (Stanley, Award) is a keyword relationship as there is a path between uni1 and prize1 in Fig. 1. A database is considered relevant if its keyword relationship model covers all pairs of query keywords.

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 pair wise related but there is no combined join sequence which connects all of them.

G-KS 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 (ki, kj) 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.

2.1 APPROACH AND DEFINITION OF SPECIFIED KEYWORD

The keywords to the relevant data sources and searching the given keyword query, we propose four different approaches. They are:

a) Keyword level model
b) Element level model
c) Set level model.

a) Keyword Level Model

In keyword level, we mainly consider the relationship between the keywords in the keyword query. This relationship can be represented using *Keyword Relationship Graph* (KRG) [7]. It captures relationships at the keyword level. As opposed to keyword search solutions, relationships captured by a KRG are not direct edges between tuples but stand for paths between keywords. For database selection, KRG relationships are retrieved for all pairs of query keywords to construct a sub graph. Based on these keyword relationships alone, it is not possible to guarantee that such a sub graph is also a Steiner graph (i.e., to guarantee that the database is relevant). To address this, sub graphs are validated by finding those that contain Steiner graphs. However, since KRG focuses on database selection, it only needs to know whether two keywords are connected by some join sequences or not. This information is stored as relationships in the KRG and can be retrieved directly. For keyword search, paths between data elements have to be retrieved and explored.

Keyword search over relational databases finds the answers of tuples in the databases which are connected through primary/foreign keys and contain query keywords. As there are usually large numbers of tuples in the

databases, these methods are rather expensive to find answers by on-the-fly enumerating the connections. A *tuple unit* is a set of highly relevant tuples which contain query keywords.

Definition (Tuple Units): Given a database *D* with *m* connected tables, *R1*, *R2*, . . . ,*Rm*, for each tuple *ti* in table *Ri*, let *Rti* denote the table with the same primary/foreign keys as *Ri*, having a single tuple *ti*. The joined result of table *Rti* and other tables *Rj*($j \neq i$) based on foreign keys, denoted by R =j≠i Rj Rti, is called a tuple set. Given two tuple sets *t1* and *t2*, if any tuple in *t2* is contained in *t1*, we call that *t1* covers *t2* (*t2* is covered by *t1*). A tuple set is called a tuple unit if it is not covered by any tuple set.

b) Element Level Model

Keyword search relies on an element-level model (i.e., data graphs) to compute keyword query results. Elements mentioning keywords are retrieved from this model and paths between them are explored to compute Steiner graphs. To deal with the keyword routing problem, elements can be stored along with the sources they are contained in so that this information can be retrieved to derive the routing plans from the computed keyword query results.

In this model, we mainly concentrate on IR technique of data retrieval. This technique allow users to search unstructured information using keyword based on scoring and ranking, and do not need users to understand any database schemas. We use graph-based data models to characterize individual data models.

Definition (Element-level Data Graph): An element-level data graph g (*N*, *ε*) consists of The set of nodes *N*, which is the disjoint union of *Nε NV*, where the nodes *Nε* represent entities and the nodes *NV* capture entities' attribute values, and α The set of edges *ε*, subdivided by *ε* = *εR*] *εA*, where *εR* represents inter entity relations, *εA* stands for entity-attribute assignments. We have *e* (*n1*, *n2*) 2 *εR* iff *n1*; *n2* 2 *Nε* and *e* (*n1*, *n2*) 2 *εA* iff *n1* 2 *Nε* and *n2* 2 *NV*. The set of attribute edges *εA* (*n*) = {*e* (*n*, *m*) 2 *εA*} is referred to as the description of the entity *n*.

Note that this model resembles RDF data where entities stand for some RDF resources, data values stand for RDF literals, and relations and attribute correspond to RDF triples. While it is primarily used to model RDF Linked Data on the web, such a graph model is sufficiently general to capture XML and relational data.

c) Set Level Model

In this model we derive the summary at the level of set of elements.

Definition (Set-level Data Graph): A set-level data graph of an element-level graph *g* (*Nε NV*; *εR εA*) is a tuple *g′* = (*N′*, *ε′*). Every node *n′ N′* stands for a set of element level entities *Nn′Nε*, i.e., there is mapping type: *Nε* → *N′* that associates every element-level entity *n Nε* with a set-level element *n′ N′*. Every edge *e′* (*n′i*, *n′j*) *ε′* represents a relation between the two sets of element-level entities *n′i* and *n′j*. We have *ε′* = {*e′* (*n′i*, *n′j*) | *e* (*ni*, *nj*) *εR*, type (*ni*) = *n′i*; type (*nj*) = *n′j*}.

This 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. Often, a schema might be incomplete or simply does not exist for RDF data on the web. In such a case, a pseudo schema can be obtained by computing a structural summary such as a data guide. A set-level data graph can be derived from a given schema or a generated pseudo schema. Thus, we assume a membership mapping type: *Nε* → *N′* exists and use *n n′* to denote that *n* belongs to the set *n′*.

2.2 ALGORITHM FOR SPECIFIED KEYWORD SEARCHING

Algorithm:

PPRJ ComputeRoutingPlan(*K*, *W′K*)
Input: The query *K*, the summary *W′K* (*N′K*, *ε′K*)
Output: The set of routing plans [*RP*]
JP ← a join plan that contains all {*ki*, *kj*} 2*K*

$T \leftarrow$ a table where every tuple captures a join sequence of KERG relationships $e'K\ \varepsilon'K$, the score of each $e'K$, and the combined score of the join sequence; it is initially empty;
While *JP* .empty()
do {*ki*, *kj*}← *JP* .pop();
ε'{*ki*, *kj*}← retrieve($\varepsilon'K$, {*ki*, *kj*});
if *T*.empty() **then** $T \leftarrow \varepsilon'${*ki*, *kj*};
else $T \leftarrow \varepsilon'${*ki*, *kj*} *T*;
Compute score of tuples in *T* via SCORE (*K*, *W'SK*);
[*RP*] ← Group *T* by sources to identify unique combinations of sources;
Compute scores of routing plans in [RP] via SCORE (*K*, *RP*);
Sort [*RP*] by score;

Such a way they have offered a solution to the novel problem of Specified Keyword Mining. Based space as a multilevel inter-relationship graph, they proposed a summary model that groups keyword and elements relationships at the level of sets, and developed a multilevel ranking scheme to incorporate relevance at different dimensions. This paper showed that when routing is applied to an existing keyword search system to prune sources, considerable performance gain can be achieved.

III. Experiment and Result

a) Keyword Search
Result

3.1Searching for the keyword "edison"

Fig shows the displayed results for the queried keyword 'edison'. There are total 6 records matching for the keyword. Out of the 6 results displayed, 5 records are having information about Thomas Alva Edison and one record has information about Dr. Edison Rodrigues. We now show the triples generated for the Thomas Alva Edison.

3.2Search results for the keyword Edison
b) With Enhanced Search Result

The subject elements from the triples generated in the proposed search are given as input to the enhanced search. The enhanced search applies the concepts of Maximum Likelihood Algorithm on the resultant linked resources of the proposed search to get the estimations of the keyword in result sources. With obtained estimation values we get the knowledge of how relevant are the resulted sources to the user query. Fig3.3 shows the querying of the enhanced search. In enhanced search the subject part of the triples generated are al extracted and given as input to the enhanced search. The enhanced search then searches different datasets to retrieve matching results for the queried keyword.

3.3Querying the Enhanced Search

The enhanced search then mines the the linked resources to find the number of times the queried keyword is present in the sources. It mines the entire result document to find the count of the occurance of the keyword in the result document which is displayed along with the results. With the count we can determine how relevant is the document for the user query and can filter the documents that are less relevant by setting a count threshold for he result to be displayed. Fig 3.4 shows the results displayed in the enhanced search with the occcurance count value of the queried keyword.

3.4 Results generated with count values

IV.CONCLUSION

We have presented a solution to the novel problem of Specified Keyword Mining. Based on modeling the search space as a multilevel inter-relationship graph, we 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. The experiments showed that the summary model compactly preserves relevant information. In combination with the proposed ranking, valid plans (precision@1 ¼ 0:92) that are highly relevant (mean reciprocal rank ¼ 0:86) could be computed in 1 s on average.

REFERENCES

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

[2] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou, "Ease: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-Structured and Structured Data," Proc. ACM SIGMOD Conf., pp. 903-914, 2008.

[3] H. He, H. Wang, J. Yang, and P.S. Yu, "Blinks: Ranked Keyword Searches on Graphs," Proc. ACM SIGMOD Conf., pp. 305-316, 2007.

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

[5] Y. Luo, X. Lin, W. Wang, and X. Zhou, "Spark: Top-K Keyword Query in Relational Databases, " Pr oc. AC M SIG MOD Conf., pp. 115-126, 2007.

[6] 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.

[7] Pratiksha Nikam and Prof. Srinu Dharavath, "Review of Approximate String Search in Spatial Dataset," International Journal of Multidisciplinary and Current Research Vol.2 (March/April 2014)

[8] 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.

[9] 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.