Efficient Keyword-Based Searching Strategies for Linked Databases

SRINIVAS VNVSR **SONY KRISHNA R**
Department of Computer Science Engineering, Andhra Loyola Institute of Engineering and Technology, India.

Abstract:
Increasing the linked data sets in the web, web user's addresses the problem of querying is complex. Instead of browsing along links through results and querying the complex queries, and querying the linked data using structured languages is complex. Searching with a keyword is better because users might not know the query languages and also no need of remembering the schema of the database. In this paper, we implement different techniques based keyword query routing which addresses the problems of linked data querying. And find the keyword search, keyword element relationship, set level relationship, multilevel interrelationships, routing plan. We employ a keyword element relationship that represents the relation between keywords and data elements mentioning them, based on this we find the exact data source.

Index terms:
keyword search, keyword element relationship, set level relationship, multilevel interrelationships, routing plan.

1. Introduction:

Most of the information needs are based on web, web is not only a collection of single source data it also specifies a data which is linked with other sources, querying that large amount of data is challenging task. Linked data comprise of hundreds of sources Contain billions of RDF tuples, which are connected by many of links.

Example 1: Person X is a new computer science student at Y College. She(X) is interested to learn more about this vast research field and decided to find information about research work of researchers at Z University. This example mainly specifies that users face the burden of accessing web of data is hard,

Because knowing the schema is must by using any query language means. And it is not correct way to think that by querying and accessing the web of data is typical task, because the web contains not only a collection of textual documents but also an interlinked data i.e., remembering the schema of linked data is not easy task so in this scenario keyword search is solution. By using keyword searching no need of remembering the query languages and database schema. Concerning these problems the question we deal with this is user requirement is expressed in terms of keywords, that query is either single or multiple keyword sets. Before a period of time keyword search is supported by some of semantic web search engines such as SWSE and SIGMA, they are limited to processing a simple list of keywords, in that user can refer. Whereas this approach deals with finding relationships among the keywords and display the results.

Before while existing approaches select single databases, in this approach we deal with linked databases, in this results are combined from multiple databases. In this we can find the database sources for individual keywords and make relationships between to get accurate results. The goal of this approach is generating the routing plans which capture the combination of sources that contain non-empty sources. And queried input keywords cover several linked sources. In database research, solutions can be find based on a given query is of two types, first finding the most exact results, and second finding the single most exact databases, so these are the approaches to single source solutions. They are not exactly exact to the web of linked data, here we have to find the an exact sources from exact combination of sources because it is linked data sources. The aim is to find the routing plan from the multiple sources. At last we specify the techniques:

- Here we investigate the problem of keyword search over structured and linked data sources, routing keyword to linked data sources. This approach can reduce the high cost of searching.
- In existing system keyword relationships can be used for single databases, here we present relationships between keywords as well as those between data elements.

Ad-numb	Job-type	Job-id	Sex	salary
2051	self-employed	77516	Male	30k
2052	Private	83311	Male	30k
2053	Private	215646	Male	30k
2054	Private	234721	Female	30k
2055	Private	338409	Female	30k
2056	Private	284582	Female	30k
2057	self-employed	160187	Male	60k

Fig: 1 Adult Dataset stored in database source 1

- Finding the keyword relationships for keywords which the user can query, and find the keyword relationship which is most exact to the query.

Manuscript received July 5, 2016
Manuscript revised July 20, 2016

2. Related Work

In existing system, the source is single and it is structured databases, structured database contains data can be stored in rows and columns with a particular format, or can be stored in a file with specified columns.
In this type of databases the user entered keyword query can be taken and searching the record by record and retrieve the exact results which matched exactly the keyword, by using this type of approach redundant data retrieval is possible.

Ad-numb	No.of persons	Service-numb	safety
2050	2	5000	low
2051	3	5001	avg
2052	4	5002	high
2053	2	5003	low
2054	3	5004	avg
2055	4	5005	high
2056	4	5006	low
2057	2	5007	avg

Fig: 2 Car Details Data Set Store in Data Source 2

Example 2: the user entered ***input query*** Q= {2051, self-employed, female}
Based on the user input first 2051 can be taken and search db1 retrieve results which matches
keyword1 i.e. k1= 2051➔
Result of (k1) =
{2051, self-employed, 77516, male, 30k}
In the same way for
keyword2 i.e. K2 = self-employed ➔
Result of (K2) =
{{2051, **self-employed**, 77516, male, 30k},
{2057, **self-employed**, 160187, male, 60k}}
For **keyword3** i.e.k3 = female ➔
R (k3) = {{2055, Private, 338409, **Female**, 30k},
{2056, Private, 284582, **Female**, 30k}, {2054, private, 234721, **Female**, 60k}}.
So results can be redundant and limited to single database. This type of searching not possible in linked databases.
Researchers have been developed several different types of keyword searching algorithms over a period of time with differentiation in performance and various types of data. Some of they are stated below. The algorithms used in the unstructured database belongs to schema agnostic algorithms, these are mainly used in a situation like we don't know the schema of a database.

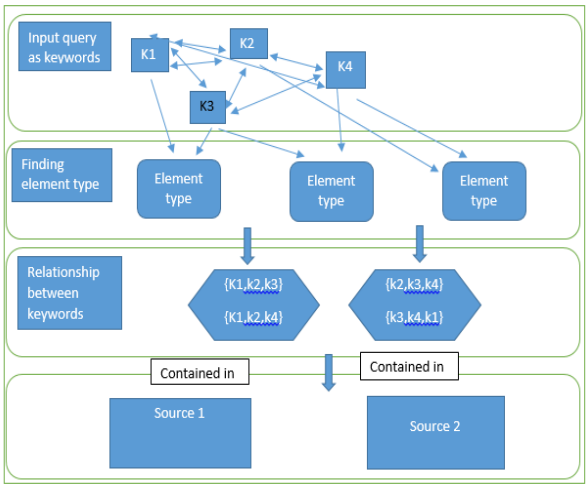

Fig: 3 Execution flow of keyword searching and Relationship

3. Proposed Framework

In this section, we deal with data, mentioning the problem, implement the solution.

3.1 Data Sources:

Data sources means repository of data, the end user can retrieve results from this data sources, in existing system only single data Source can be maintained. In this, we use the linked data scenario.

3.2. Linked Data

It means data source can be linked, it means in web of data there is no single source of repository, and data can be maintained and retrieved from different source, in the same way, multiple databases can be maintained for retrieving results. Initially finding results for individual sources and make relate with other database sources by using foreign key joins.
Non-Linked Data: this means we retrieve exact information from a single data source which satisfies the keyword query.

3.3 Data Scenario

We use linked data scenario which is of multiple databases, connected multiple databases shown in figure 1 and figure 2 we categorizes the querying in two types i.e. Element Level and set level searching.

3.4 Element Level Searching:

It is a single keyword searching which comprise the results from multiple data sources which matches exactly,

it extracts by using foreign key joins, the user use record from displayed results. In this single keyword can be matched from a different source and display all the results. Instead of searching keyword query in databases, GKS approach finds the exact database which satisfies the keyword query.
GKS algorithm specifies that finding exact results based on keyword. That keyword is exactly searched on records on databases and display all the results matched.
Algorithm 1: GKS:

Input: input keyword

Output: database source for input keyword (database location)

Start.

1. Input (keyword).

2. Performing a search for keyword matches to database.

3. Find the database location (keyword, DBLOC).

Stop.

3.5 Set Level Searching:

This is a multiple keywords searching technique, in this KER algorithm and GKS algorithm is used. GKS extracts the results for individual keywords from different sources and KER finds relationships between them and join the relationships over multiple keywords.

Algorithm: 2 KER:
Input: multiple keywords.
Output: keyword relationship between input keywords.

Start.
Step: 1 Take input keywords
Step: 2 find individual database sources of keywords by using **GKS** algorithm
2.1 **GKS** (input keyword, result).
Step: 3 Repeat step 2 until input keywords 1…n,
Step: 4 find the relationship between the keywords by using foreign key joins from different sources.
Step: 5 this relationship is resulting of all input keywords
Step: 6 stop.

3.6 Compute Keyword Relationships:

Based on the input query our application finds the all the relationships satisfied by input keyword query. Suppose for a three keyword query find the relationships for all keywords and find the database source which satisfies input query.

Algorithm 3: Keyword Relationship (keyword, KR)

Input: the query q= {k1, K2 …}

Output: set of Relationships [KR]

Start.

JP -> a join plan that satisfies all keyword in q.

Step: 1 compute the KER (input q, output RP).

Step: 2 find all pairs of keywords -> 2^k.

Step: 3 Show all the relationship pairs in a graphical representation of pie chart.

Stop.

4. IMPLEMENTATION AND TEST RESULTS:

All the proposed strategy is implemented in the system, and its hardware specifications are Processor - core i3, Speed - 2.47 GHz, RAM - 4GB, Hard disk storage - 500 GB, standard IO devices- keyboard and mouse, monitor SVGA. And software specifications is: Operating System is windows 8.1 version, Programing Language is progress 4GL, the version is 10.2b, Database is progress database, tools used - ADM2. ***1000 publicly available Datasets*** can be collected from UCI machine learning repository **https://archive.ics.uci.edu/ml/datasets.html** for testing performance.

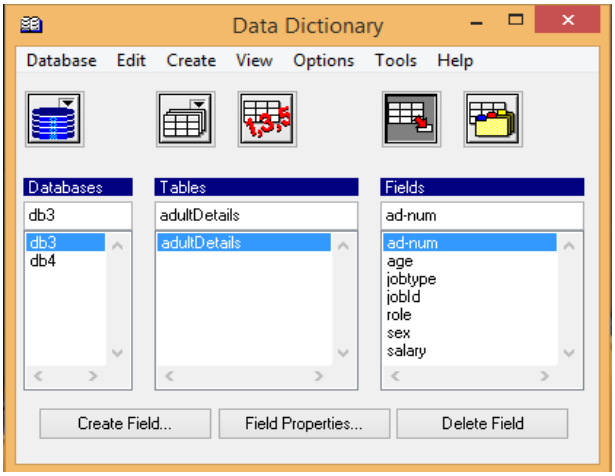

Figure 4: 1st database connected to application

From this 500 datasets can be stored in one database i.e. Database-1 ***Adult dataset***, and remaining 500 datasets can be stored in another database i.e. database-2 ***Car Evaluation dataset.*** And the following test results can show as the below strategies description.

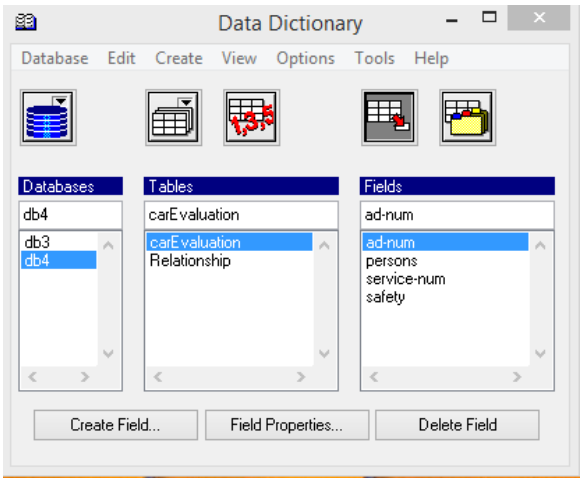

Figure 5: 2nd database connected to application Same time two databases are connected.

4.1 Element Level Searching

It deals with individual elements

Example 3: Input query as single keyword i.e. q_1 = {5004}.

This 5004 can be found which type of keyword belongs to either it is ad-num or service-num or job-id based on that it searches.

5004 is find at the 5th record the result is {2054, 5004, 3, avg}.

Suppose input query is **q2**= {male}

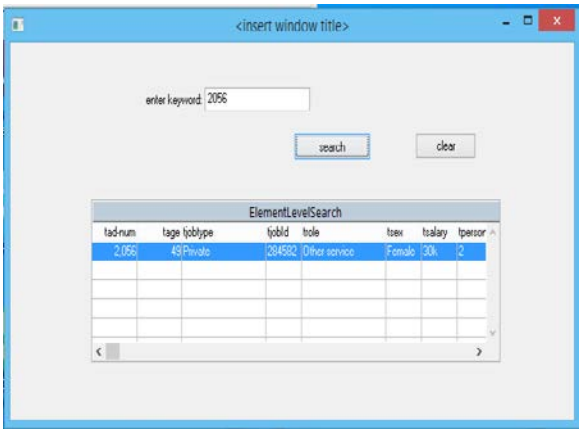

Fig 6: results when searching chooses element level

The output is like **R** => {{2051, self-employed, 77516, Male, 30k}, {2052, Private, 83311, Male, 30k}, {2053, Private, 215646, Male, 30k}, {2057, self-employed, 160187, Male, 60k}}. The Experimental result can be shown in figure 6 and figure 7.

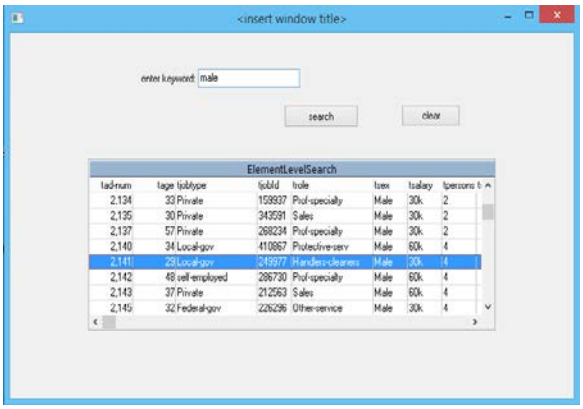

Fig 7: multiple results when searching chooses element level

4.2 Set Level Searching

It deals with a group of elements. In this, we consider the two database sources, by taking input as keywords we can find the exact results based on input taken type either it is element level searching or set level searching.

In element-level, we find the results based on individual keywords in individual databases and display all the results.

In set-level, we find the results based on multiple keywords and make a relationship between them and display most exact results which satisfy the input query keywords. It finds set result for user queried keywords by finding keyword relationships between the keywords and join the foreign key relationships with them.

Example 4: Input query as multiple keywords $\mathbf{q_2}$= {2055, private, 77516}

Here it finds the results for individual keywords by using GKS and make relationship using KER algorithms. **K1**=2055 the result of **k1**= {2055, Private, 338409, Female, 30k}.

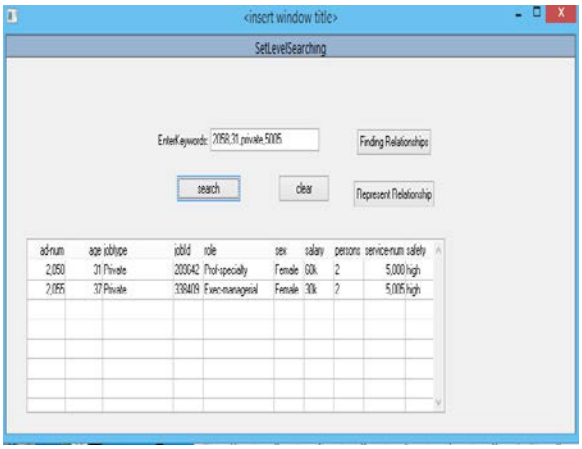

Fig 8: result retrieves when search type is set level searching

For K2= {private} the results of **K2** = {{2052, Private, 83311, Male, 30k}, {2053, Private, 15646, Male, 30k}, {2054, Private, 234721, Female, 30k}, {2055, Private, 338409, female, 30k}, {2056, Private, 284582, Female, 30k}}.

For K3=77516 the result of **k3** = {2051, self-employed, 77516, Male, 30k}.

These are the results for three keywords by using GKS algorithm. By using KER algorithm we find relationship between keywords {2055, private, 77516} the combination of {2055, private} contained records are one record k3 is not matching with this combined result .so we take the {2055, private} for one result and unmatched k3 as one record result.

The final result is:

R= {2055, Private, 338409, female, 30k} {2051, self-employed, 77516, Male, 30k}.

It closely relates to all the keyword query. For input :{ 2058, 31, private, male} the result was shown in figure 8 and figure 9.

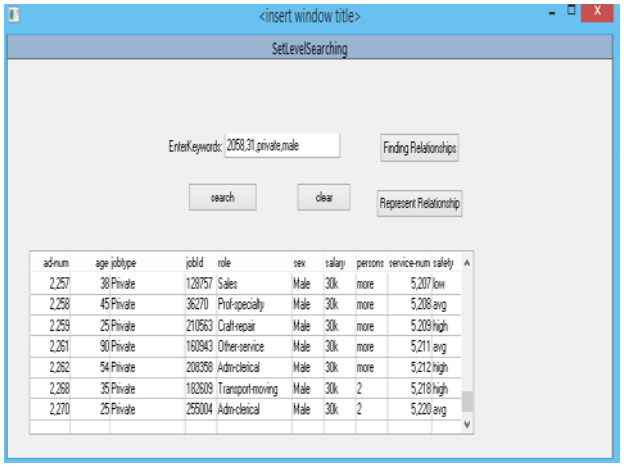

Fig 9: result retrieves when search type is set level searching

4.3 Compute Keyword Relationships:

Example 5: for input q= {2054, private, 234721, male}.the result is {2054, private, 234721} can be taken as one record and {male} cannot be in this record so retrieve {male} results as one group and three combination as one record.

The result is shown in figure 10

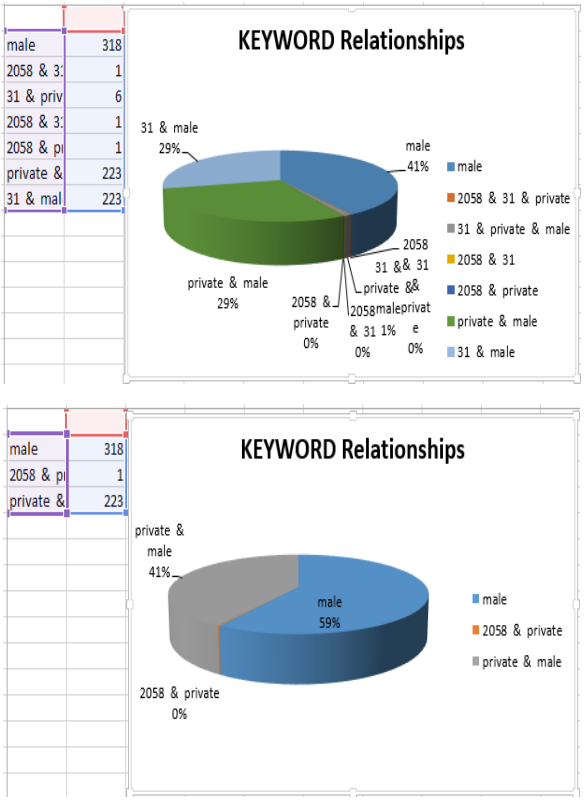

Fig 10: Result for keyword Relationship for a given query

5. Conclusion:

Based on the above observations the performance of the algorithms depends on the datasets taken for testing and choosing the database. The algorithms specified in this paper is briefed about the structured database and linked databases. For structured database, MKS algorithm easily retrieves the results. For linked databases, GKS algorithm used based on the graphs. This paper is provided the information related to data retrieval based on keywords from linked database. This paper is used for researchers and students who are interested in keyword searching in linked database.

REFERENCES:

[1] Effective Keyword-based Selection of Relational Databases..,Bei Yu Guoliang Li, Karen Sollins.
[2] Keyword Query Routing Thanh Tran and Lei Zhang.
[3] A Graph Method for Keyword based Selection of the topK Databases Quang Hieu Vu 1, Beng Chin Ooi 1, Dimitris Papadias 2, Anthony K. H. Tung
[4] 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.
[5] 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.
[6] Interlinking Multimedia How to Apply Linked Data Principles to Multimedia Fragments Michael Hausen blas DERI, Raphaël Troncy CWI Amsterdam.
[7] Linking UK Government Data John Sheridan, Jeni Tennison.
[8] Extracting Medical Information Using Linked Data Jakub Koz´ak⋆, Martin Ne˘cask´y, and Jaroslav Pokorn´y
[9] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional Expansion for Keyword Search on Graph Databases," Proc. 31st Int'l Conf. Very Large Data Bases (VLDB), pp. 505-516, 2005.

SRINIVAS VNVSR He was an M.Tech. Student and researcher at Andhra Loyola Inst of Engg & Technology. And received the bachelor's degree from JNTUK University, A.P. He has involved in some data mining, keyword search strategies projects.

SONY KRISHNA R she was a researcher and Assistant Professor in Andhra Loyola Inst of Engg & Technology. She has been working on data mining and involved in many research projects.