ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

An Improved Performance for Keyword Query Routing for Searching in Linked Databases

Shital Sungare [1] , Sonali Patil [2]

P.G. Student, Department of Computer Engineering, Alard College of Engineering and Management, Marunji, Pune, India[1]

Professor, Department of Computer Engineering, Alard College of Engineering and Management, Marunji, Pune, India[2]

ABSTRACT: Keyword Search is the Information Retrieval Methodology. In which users are not aware of a query language as well as the structure of data. In this paper we focus on the keyword query routing as well as ranking and based on the user preference the top-k results will be retrieved from the database and returned to the user. Information retrieval (IR) is the process of finding relevant to information from a great set of document. This system will provide an effective query processing. The score is depend on ranking function. And it can be calculated by different methods. These scores are the basic for the ranking of the documents. Keyword query search is a technique for searching relevant data sources. It also focuses on how effectively processes the keyword queries. Here we propose to routing keywords only as it reduces the processing keyword search queries over all linked sources. We propose a method for computing top-k routing result using Graph keyword search method. We also propose Graph Keyword Search method. It represents each database by a keyword relationship graph. We are focusing on a keyword-element relationship summary that compactly represents relationships between keywords and the data elements representing them. In this paper, we solve the two problems to improve the efficiency and effectiveness of aggregate keyword by using general ranking scheme and ranking algorithm.

KEYWORDS: Text WebDataGraph, RDF, Data Graph, Graph Structured Data, Keyword Element Relationship Graph, Keyword Relation. SIL, DEINIX

I. INTRODUCTION

In the past decade, extending the keyword search paradigm to relational data has been an active area of research within the database and information retrieval (IR) community. In this research paper some approaches have been proposed and implemented. In this , we focus on the LOD (i.e Linking Open Data). Tremendous amount of data present on the internet i.e. on web e.g. documents, as web is a form of interlinked data sources. Through this project, a large amount of legacy data have been transformed to RDF. It is a metadata model. It contains billions of RDF triples, which are joined by millions of links. The RDF triples are Subject, Predicate and Object. RDF stands for Resource Descriptive Framework representing metadata in XML.

RDF data model is an labelled graph Nodes are resources edge labels are properties. Now a days the existing search engines like Google it produces list of individual pages as a result. It cannot integrate the result from multiple interrelated pages in order to answer the keyword queries. In the meanwhile the keyword search on structured and semi-structured data is challenging task. Because it uses inverted index, processing for keyword queries. There is a classification of data as structured, unstructured, semi structured data. The structured data it consists of relational databases and graph data. Unstructured data contains textual documents. And semi- structured data consists of XML

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

documents. All of these datasets are published according to the Linked Data principles [1] a simple and accessible paradigm that requires data providers to use:

(i) HTTP [2], as data access protocol which is required,

(ii) HTTP-scheme-based URIs [3] as identifiers for entities describe in the data, and

(iii) RDF [4], as the data model to represent data.

Any HTTP-URI in an RDF triple may then be understood as a data link that enables Linked Data-aware software clients to retrieve more data by looking up the URI on the Web. The simplicity and convenience of these principles is one of the drivers behind the proliferation of Linked Data. However, the simple, URI lookup-based data access method that goes along with these principles does not provide the powerful querying capabilities usually associated with data sources in a distributed database system. While some dataset providers offer additional query processing services to overcome this constraint [5].Many datasets are therefore not presented through such services or the services are often unpredictable [6].

Keyword Query Search can be broadly classified into two categories based on how they use schema information:

1) Approach with Schema:

In these approaches, it implemented on top of databases. A keyword query is processed by mapping to the element of the database. Schema based approaches are only applicable to querying on relational data.

2) Approach without Schema:

These approaches are applicable to graph data, not to relational data. It operate directly on data. Graph data consists of node and weights exist on edge, and answers are calculated and ranked on these weights. And it can be represented by Steiner graph. Keyword Query Routing is a process of routing keywords only. It gives only relevant result. In the existing system the Top-K routing plan is the result we get. The existing system uses schema matching approach to find out the links between the sources and find foreign key joins across the sources. We also used a graph based data model. It is defined as, Set-level data graph Which captures the information regarding the group of elements. In this data graph there is a mapping from element level entity with set level element.

- Element level datagraph It shows the relationship between the individual data element. Existing work on keyword search is based on element level model for computation of query result. In the existing system KERG plays important role. KERG stands for Keyword Element relationship Graph. We provide the following contributions.
- Existing work uses the keyword relationships which are collected individually for single databases. We are also showing the relationship between keyword and data element.
- We propose the query routing for keyword search for relevant sources. Here we are searching only relevant sources so it reduces high cost of searching and processing for the results.
- We propose Density Inverted Index Method which reduces the memory storage space hence the pre-processing time is also reduced.

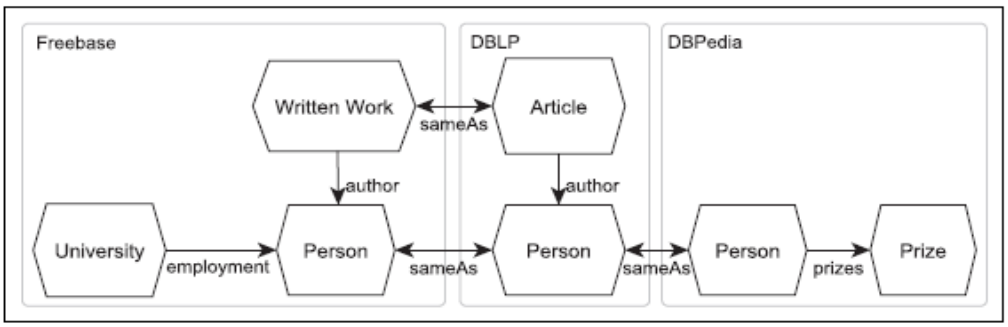

Figure:1. Set Level Web Data Graph

II. PROBLEM STATEMENT

The problem defines of the keyword search over linked database. The exact keyword search for the single database can be done by LIKE predicate which is already predefined in SQL and also by creating a User-Defined function in SQL. These were the two method which works without index. Both the methods describe deals with single database which is the drawback. The research done in this area is confined to search in single databases. So, we can propose a keyword

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

search technique which searches in more than one database. We should keep in mind that introducing the new technique doesn't increases the query processing time. For keyword searching the inverted index table is used to store the keywords but this method will consume more memory space. So, we will remove both the drawbacks by proposing a new approach SIL using DeINIX which will decrease the time complexity as well as the space complexity

III. LITERATURE SURVEY

Vagelis Hristidis [7] DISCOVER operates on relational databases and facilitates information discovery on them by allowing its user to issue keyword queries without any knowledge of the database schema or SQL. DISCOVER proceeds qualified joining of networks of tuples, that is, sets of tuples that are related because they join on their primary and foreign keys and collectively contain all the keywords of the query. Also proves that the selection of the optimal execution plan (way to reuse common sub expressions) is NP-complete.

Guoliang Li Beng Chin Ooi [8] Conventional keyword search engines are restricted to a given data model and cannot easily adapt to unstructured, semi structured or structured data. This paper proposes an efficient and adaptive keyword search method, called EASE, this is for indexing and querying large collections of heterogeneous data. In order to achieve high efficiency in processing keyword queries, we first form unstructured, semi-structured and structured data as graphs, and then sum up the graphs and construct graph indices instead of using traditional inverted indices. We put forward an extended inverted index to facilitate keyword-based search, and present a novel ranking mechanism for enhancing search effectiveness.

Bolin Ding, Yintao Yu, Bo [9] This paper elaborate problem of keyword search in a data cube are called as text cube. The text cube is built on a multidimensional text database, where each row is related with some text data (e.g., a document) and other structural dimensions are called (attributes). so it looks like a Document And Attribute pair. The text cube combine a set of documents with matching attribute values in a subset of dimensions. our intend is to find the top-k relevant cells in the text cube.

Holger Bast Alexandru Chitea [10] ESTER is a modular and highly capable system for combined full-text and ontology search. ESTER build on a query engine that supports two basic operations: prefix search and join. Both operations can be implemented very efficiently with a compact index, in combination it provide powerful querying capabilities. Also shows how ESTER can answer basic SPARQL graph pattern queries on the ontology by reducing them to a small number of these two basic operations. ESTER further supports a natural blend of such semantic queries with ordinary full-text queries. Moreover, the prefix search operation allows for a fully interactive and proactive user interface, which after every keystroke suggests to the user possible semantic interpretations of his or her query, and speculatively executes the most likely of these interpretations.

IV. PROPOSED SYSTEM ARCHITECTURE

A) SYSTEM ARCHITECTURE

The fig 2 .Shows the system architecture. The user will give the query to search in the user interface. The query will be sent to the query evaluation engine. Where the query will be processed by the generating the candidate network and the then it is sent to the searcher. The searcher which will search for the keyword by performing the match. The searched results will be sent to the ranker which will rank the results based on the calculated score. Finally the ranked result will be sent to the user interface for the user to view the result. The results that are obtained are stored in a summary database. When we search the keyword query already used by the system then it checks if similar searches has been performed. And if such search has been performed then the summary database is checked to retrieve the result. And finally integrate it with the updated values in the database. Ranking is a major problem in many applications where information retrieval is performed. When documents are retrieved from RDBMS based on a user query then ranking plays an important part as to deliver the documents in a sorted order based on special parameters. Subsequent to Ranking the data will be sorted and Top-K result will be retrieved.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

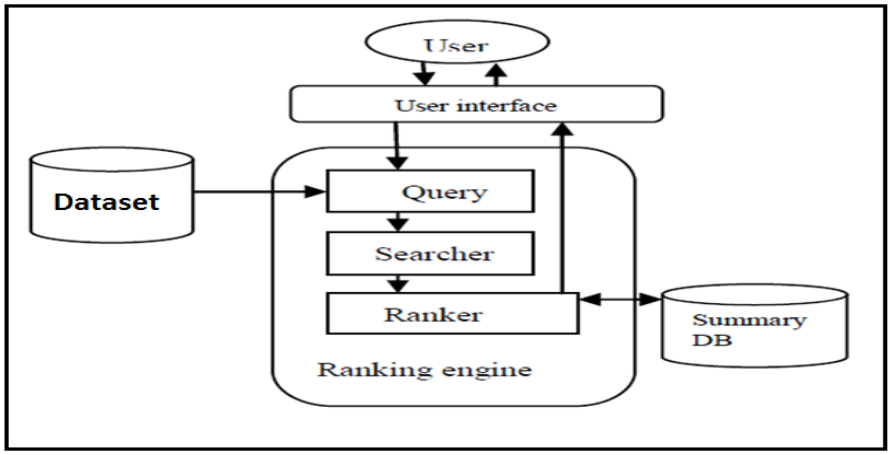

Figure: 2. Query routing Architecture

A) SEARCHING FOR SINGLE DATABASE

1. Index Based Method

In an auxiliary table is constructed for indexing to facilitate the prefix search. The Oracle and SQL have inbuilt facilities use prefix search. But this method is limited to some. databases hence a method introduced in which construct Inverted index and Prefix table for searching more efficiently. For constructing an inverted index table, firstly an unique id is assign to keywords in alphabetic order, which are in **Table T**. Afterwards inverted **index table IT** is created with records in form **<kid,rid>** where kid is the id of keyword and rid is id of record .

For a table T, Prefix table PT is created with **<p, lkid, ukid>** where p represent the prefix of keywords, lkid the largest string id of prefix p and ukid smallest string id of p. Let's take an example. The inverted index has tuple **<k8, r6>** the k8 has keyword "sigmod" is in record r8. The prefix table constructed is **("sig" , k7, k8)** ,now k7 has "sigir" with minimum length and k8 has "sigmod" with maximum length. Hence id will in range **< k7, k8>.** For a given Partial Keyword w, we first extract the range of **prefix table PT** then find the record in the following. The drawback of this it works on single database more over the Keyword Table, Inverted Index and Prefix table which are created consume more space and time too. The second one is that it uses SQL, for that we need to learn an underlying schema to search a Keyword. So we propose an efficient algorithm SIL with DeINIX which will answer the keyword queries without knowing the schema of database on top takes less time and less space compared. The PROPOSED work consists of the two components one is the Pre-Processing and the other is Query Processing. The pre-processing and the query processing are explained in detail. Following Table-I shows a) Keyword Table, b) Inverted Index Table, c) Prefix Table

A) KEYWORD TABLE

Kid	Keyword
k1	icde
k2	icdt
k3	preserving
k4	privacy
k5	publishing
k6	pvldb
k7	sigir
k8	sigmod
k9	vldb
k10	vldbj
.....	.....
.....	.....

B) INVERTED INDEX TABLE

kid	Rid
K2	r10
K5	r6
K5	r8
K5	r10
K6	r1
K7	r9
K8	r3
K8	r6
K9	r8
K10	r4
.....	.....

C) PREFIX TABLE

Prefix	lkid	ukid
ic	k1	k2
p	k3	k6
pr	k3	k4
pri	k4	k4
pu	k5	k5
pv	k6	k6
pvl	k6	k6
sig	k7	k8
v	k9	k10
vl	k9	k10

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

B) Pre-Processing

Firstly the database is collected and then table of Inverted Index is created. The Inverted Index table is created by using DeINIX. In DeINIX traditional Inverted Index technique, keyword occurring in the database has been assigning different IDS (Table-I(A)). Let me explain by an example in (Table II(a)) the word "preserving" is occurring in r1, r2,r3 ,r4 and r5 these index are stored individual in Inverted Index table (Table II (a)) which result in more memory utilization, hence we propose a technique DeINIX to support keyword search using range list. Instead of individual we set them in range intervals <r1,r5> which reduces the memory space of the Inverted Index table and make the performance better and fast. Again a question arises what if a we have keyword like "sigir" is occurring only once in database , the solution for that is we have to write <r9,r9> for occurrence of the keyword once. Other condition is if we can't have continuous interval then we can write as <r5,r5>,<r7,r7> and <r2,r2> for "icde" keyword. After construction of table then we processed towards query processing.

D) Query Processing

After the pre-processing is done then the user enters the keyword then it proceeds towards following steps:

- **Step 1: Keyword Matching**

The basic idea of keyword matching is to find the answer for the keyword query. The user types the and fetch the answer by retrieving form the DeINIX table and apply Searching in Linked Databases.

- **Step 2: Searching In Linked Databases**

The main challenge is to process the keyword over Linked databases. In this paper algorithm is propose to link the databases via foreign keys. In order to join the foreign keys we need to first of all identify the foreign keys in the individual database. The two databases must be sharing approximate common attribute field then only they will be joinable. The next step is to link up two databases with help of foreign keys. Now the databases are link the selection of the keyword is based on DeINIX table . After the searching is completed it displays the result and will go to next step 3.

- **Step 3:** After the query is generated, multiple result tuples are produced. Now the question arises which answer is relevant to the keyword search for the end user. Following Table-II shows working of (a) Traditional Inverted Index And (b) Density Inverted Index

(a) Inverted-Index Table

keyword	Rid
Icde	R2,r5,r7
icdt	r10
preserving	r1,r2,r3,r4,r5
privacy	r1,r2,r3,r4,r5,r6,r7,r8,r9,r10
publishing	r6,r8,r10
pvldb	r1
sigir	r9
sigmod	r3,r6

(b) DeINIX-Table

Keyword	Rid(Range)
icde	[r2,2],[r5,r5],[r7,r7]
icdt	[r10,r10]
preserving	[r1,r5]
privacy	[r1,r10]
publishing	[r6,r6] ,[r8,r10]
pvldb	[r1,r1]
sigir	[r9,r9]
sigmod	[r3,r3],[r6,r6]
. . . . .	. . . . .

V. ALGORITHM WITH MATHEMATICAL MODEL

In this research paper DeINX algorithm is used. The stepwise execution of algorithm is given here with the help of Mathematical Model as follows,

- Let 'I' be the Input sets , I = {RE_{db}, T_{rdf}, KQ} Elements of I are,
  Element, RE_{db} - It represents Repository Dataset of Linked Data
  Element, T_{rdf} - It represents Set of RDF Triples
  Element, KQ - It represents Query Keyword by the User
- Let 'P' be the set of Process, P = { E_x, C_v, R_p, C_o, S_e, P_l }
  E_x = Extraction Process of Linked Data Sources From $RE_{db,}$

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

C_v = Conversion of Linked Data into RDF Triples
R_p = Representation into Graph Based Data Models
C_o = Index Construction Using DeINIX Algorithm
S_e = Query Keyword Searching by the user
P_l = Graph Routing and Planning

- Let 'O' be the set of Outputs, O = {KQ_r }

Element, KQ_r = Set of Query Keyword Result

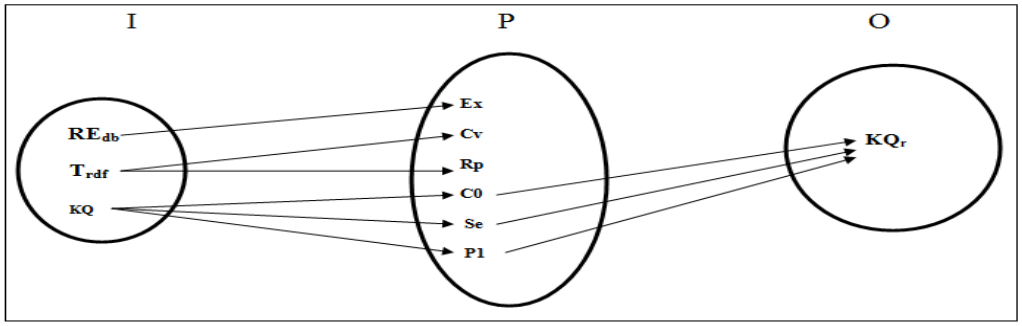

Figure : 3. Query Routing Process

The following Algorithm is a **Density Inverted Index Algorithm** . Which takes input as a keyword query and output is the answer for the query.

```
Input: Query Keyword K
Output: DeINIX Table D; Set of answer of Keyword K
For all K in [1, n] do
Let rk be the first interval list of Rk
Insert lb (rk) and ub (rk) to L and U
While U≠NULL do
Let l be top (max) element in L
Let u be top (min) element in U
If l≤u then Add [l, u] to I the interval list
If l ≠ u
Let i ∈ Ij be consecutive interval Let i' be the next interval (if any) in Ij
Then l (i') and u (i') to Ik
Return D
Determine the set of query keyword K from G
Join the Foreign-Key FK if same for two tables
Return Keywords
```

Figure: 4. Algorithm

Where L is the Lower Bound and U is the Upper Bound of the element

VI. EXPERIMENTAL EVALUATION

The standard configuration required to built the system using Java framework (jdk 6.0) on windows 7 platform. Net beans 7.2.1 is used as a developer tool. Any standard machine is capable of running this application. In proposed system uses any available dataset. The dataset is taken from http://DBpedia.org. BTC dataset is used. It consists of Ontology's of web. It contains about 36 billion things and 128 different languages. Dataset is used for keyword extraction. We have implemented the existing and propose work on real datasets. We have implemented the technique in Java for the proposed technique of keyword searching in Linked databases and DeINIX for constructing Inverted Index. The theoretical time complexity of the existing work and proposed work is On(n+1) and O(n+1) respectively and the space complexity is O(n3) and O(n) respectively.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

VII. RESULTS

According to following chart the DeINIX gives better performance than Inverted Index. So we use The Searching In Linked data with DeINIX answer keyword queries more efficiently. This introduces a novel technique i.e DeINIX(Density Inverted IndeX) which reduce the memory storage space . Major Advantage of using this system is pre-processing time reduced. This method answers queries more precisely, takes less time and memory space to retrieve answer.

The result analysis of the propose technique is shown in fig.5 with help of graph. We compare the result of LIKE method, Index-based method with prefix table and Searching in Linked Databases with DeINIX for keyword search for Housing Dataset, Computer Hardware dataset and Movie dataset. So the evaluation for all three can be seen in graph. All this searching technique works for single database. The new technique proposed works on Linked databases and the indexing method propose saves memory space too as shown in fig.5.

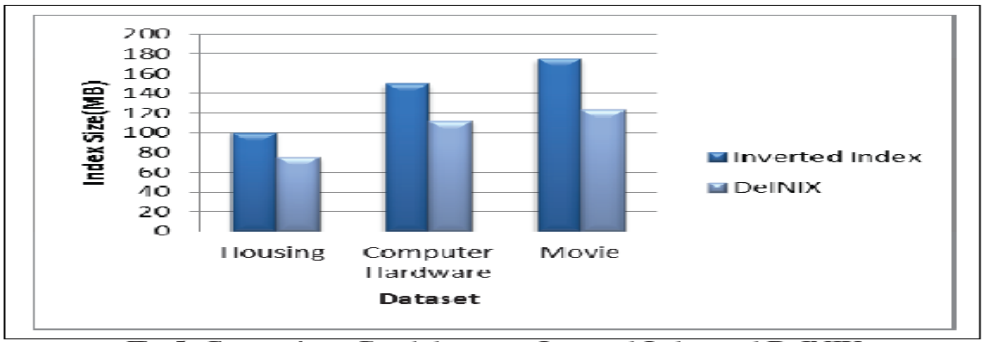

Figure: 5. Comparison Graph between Inverted Index and DeINIX

VIII. CONCLUSION AND FUTUREWORK

The keyword search over relational database works on the single database, we have proposed an algorithm for SIL using DeINIX which answers the keyword queries in Linked databases with linking the two databases via common foreign key. DeINIX is a new inverted index technique which reduces the memory space. In future work, an approach can be developed the ranking technique for Linked databases which can reduce the query execution time. An approach can be developed for an efficient algorithm for fuzzy keyword search. Moreover, one can design an algorithm for such datasets where there is no common attributes.

REFERENCES

1. T.Berners-Lee,"DesignIssues:LinkedData."[Online], http://www.w3.org/DesignIssues/LinkedData.html
2. R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. J. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, Jun. 1999.
3. T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986, Jan. 2005. [Online].
4. G. Klyne and J. J. Carroll, "Resource Description Framework (RDF):Concepts and Abstract-Syntax,"W3CRecommendation,Feb.2004.[Online].Available: http://www.w3.org/TR/rdf-concepts/
5. L. Feigenbaum, G. T. Williams, K. G. Clark, and E. Torres,"SPARQL Protocol," W3CRecommendation,Mar.2013.[Online].Available: http://www.w3.org/TR/sparql11-protocol/
6. L. Feigenbaum, G. T. Williams, K. G. Clark, and E. Torres, "SPARQL 1.1Protocol," W3CRecommendation,Mar.2013.[Online].Available: http://www.w3.org/TR/sparql11-protocol/
7. C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche, "SPARQL Web- Querying Infrastructure: Ready for Action?" in Proc. of the 12th International Semantic Web Conference (ISWC), 2013
8. V. Hristidis and Y. Papakonstantinou, DISCOVER: Keyword search in relational databases, in Proc. of the 28th International Conference on Very Large Databases, Hong Kong, China, 2002, pp. 670-681.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

9. Guoliang Li Beng Chin Ooi, EASE: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-structured and Structured Data in Proc. of the ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 2008, pp. 903-914.
10. Bolin Ding*, Yintao Yu*, Bo*, Cindy Xide Lin*, Jiawei Han*, And Chengxiang Zhai Keyword Search In Text Cube: Finding Top-k Relevant Cells
11. E. Minack , L. Sauermann , G. Grimnes , C. Fluit , J. Broekstra: The Sesame LuceneSail: RDF Queries with Full-text Search. NEPOMUK Technical Report 2008-1, Feb. 2008
12. Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen ,"Dynamic Query Forms for Database Queries",IEEE -2013.
13. Y. Luo, X. Lin, W. Wang, and X. Zhou, "Spark: Top-K Keyword Query in Relational Databases," Proc. ACM SIGMOD Conf.,
14. pp. 115-126, 2007.
15. H. He, H. Wang, J. Yang, and P.S. Yu, "Blinks: Ranked Keyword Searches on Graphs," Proc. ACM SIGMOD Conf., pp. 305-316, 2007.
16. Hao Wu, Guoliang Li, and Lizhu Zhou ,"Ginix: Generalized Inverted Index for Keyword Search", IEEE-2013.