Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Query Expansion for Efficient Keyword Query Routing System

Pawar Prajakta Bhagwat
Department of Computer Engineering,
Matoshri College of Engineering and Research Center, Eklahare, Nashik
Savitribai Phule University Pune, Maharashtra, India

Abstract-- A symbol of operation that looks for twin documents that carry one or more than line specified by the individual is called keyword search. Detect the content we demand. It is for searching linked information sources on the computer network. To route keywords only to the relevant sources to reduce high cost of processing keyword search queries over all sources. Aim is to improve the performance of keyword search, without compromising its result quality. In the proposed system, query expansion takes place using correlated, linguistic and semantic features. The goal of keyword expansion is to improve precision and recall. Relation between Keywords and the elements of data takes place by using a keyword-element relationship. Here, two types of search techniques. One is element level search and another is set level search technique. The proposed system uses routing keyword search for queries having many keywords. This improves the performance of keyword search. This way can greatly reduce time and space costs.

Keywords-- Data mining, keyword search, keyword query, keyword query routing, routing plan, query expansion.

I. INTRODUCTION

Recently web has evolved from linked documents. Linked data is an approach to publishing and sharing data on the web. Data from different domains i.e. companies, people, books, films, scientific publications, television ,music, and radio programs, proteins, genes, clinical trials and drugs, online communities, scientific data and statistical, and reviews. We propose to enquire the difficulty of keyword query routing over a huge number of Linked and structured Data sources for keyword search. To route keywords to appropriate sources reduces the cost of searching. We use a graph-based information model to characterize separate collection sources. In that role model, we describe between an element-level data graph, which represents relationships between separate data elements, and a set-level graph of data, which takes data about group of elements. This set-level graph takes a part of the Linked Data schema that is in RDFS, i.e., relations between classes.

The web is not only text data but also interlinked data. Querying large amount of data is challenging. Linked Data consists of hundreds of sources. Those sources having billions RDF triples, which are connected by millions of links course. While dissimilar forms of links can be formed, the often published are sameAs, which shows that two RDF resources show the same real-world object.

The linked data contains data in different areas, such as ecommerce, the biosciences, and e-government. To search linked data we use this keyword search method which use keyword query routing. To reduce the more cost in searching structured results that span many sources, we propose to route the keywords to the appropriate databases. The aim is to create routing plans, which can be used to calculate results from many different sources. For selecting the true routing plan, we use graphs that are based on the relationships among different keywords which are present in the keyword query. This is considered at the many different levels such as keyword, element, set level etc.

II. RELATED WORK

Keyword Query Search into two directions of process: 1] Keyword search approaches. 2] Solutions for source selection. In keyword searching, we use two methods. That is schema-based and schema-agnostic. Schema-based are implemented on top of off-the-shelf databases. Schema-agnostic methods directly on data. By exploring the underlying graphs the structured results are calculated in these approaches. The aim of this method is to produce structures in the Steiner trees. Different forms of algorithms have been proposed for the efficient keyword search results over data graphs, which might be very huge. Dynamic programming [5] and Bidirectional search [4] are examples. A system Kite extends schema based method to produce candidate networks in the multi source setting [6].

In order to produce the relevant results for keyword search, the selection of the efficient data sources plays a vital role. M-KS [3] takes only binary relationships among keywords. It incurs a huge number of false positives for queries with more than two keywords. G-KS [8] addresses this trouble by considering complex relationships among keywords using a Keyword Relationship Graph (KRG). Each node shows a keyword. Edge between two nodes corresponding to keywords.

For routing the keywords to the appropriate sources and searching the given keyword query, we propose four different methods. They are: 1) Keyword level 2) Element level, 3) Set level models, and 4) Query expansion using semantic and linguistic features. We calculate the keyword query result, keyword routing plan [1] which is the two

© 2015, IJARCSSE All Rights Reserved

crucial factors of keyword routing. In keyword level, we consider the relationship among the keywords in keyword query. This relationship can be represented by Keyword Relationship Graph (KRG) [8].

To address this trouble, tuples units [9] to answer keyword queries. It is a set of highly efficient tuples which containing queries keywords. Thanh Tran and Lei Zhang proposed Keyword Query Routing in Feb, 2014. Here input is keyword query and output is routing plan. Routing keywords returns all the sources which may or may not be the relevant sources. No query expansion takes place [1]. V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar proposed Bidirectional Expansion for Keyword Search on Graph Databases in 2005. Here bidirectional search algorithm introduced, which improves on backward expanding search by allowing forward search from potential roots towards leaves. Ranking not takes place properly [4]. T. Tran, H. Wang, and P. Haase proposed Hermes: Data Web Search on a Pay-as-You-Go Integration Infrastructure in 2009. Hermes is an infrastructure for data web search that address a number of challenges involved in realizing search on the data web. By translating keywords into structured queries, it provides an end-user oriented interface [7].

Jianhua Feng, Guoliang Li, and Jianyong Wang proposed Finding Top-k Answers in Keyword Search over Relational Databases Using Tuple Units in December 2011. To improve search efficiency and to achieve high performance, two indexes are introduced. First is single keyword based structure -aware index and keyword pair-based structure-aware index. But this feature does not support on-line update and does not reduce the index sizes [9]. G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou proposed Ease: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-Structured and Structured Data in 2008. Here, method name for efficient and adaptive keyword search is EASE. This is for indexing and querying large collections of heterogeneous data. An extended inverted index to facilitate keyword based search [10].

R. Goldman and J. Widom proposed DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases in 1997. DataGuides are very important for browsing database structure, query formulation, information storing such as statistics and sample values and enabling query optimization. Dataguide feature supports managing and querying semistructured data [11]. K. Collins-Thompson proposed Reducing the Risk of Query Expansion via Robust Constrained Optimization in 2009. Robust optimization approach provides clean theoretical way to model not only expansion benefit, but also expansion risk, by optimizing over uncertainty sets for the data. Here risk- reward curves also introduced to visualize expansion algorithm performance and analyse parameter sensitivity [12]. Saeedeh Shekarpour, Konrad Höffner, Jens Lehmann and Sören Auer proposed Keyword Query Expansion on Linked Data Using Linguistic and Semantic Features in 2013. Query expansion methods augment the original query of a user with alternative query elements with similar meaning to increase the chance of retrieving appropriate resources. Here, new query expansion features based on semantic and linguistic inferencing over Linked Open Data are introduced. Existing system having some disadvantages. Routing keywords return all the sources which may or may not be the relevant sources.

III. THE PROPOSED SYSTEM

It is for searching linked information sources on the computer network. To route keywords only to the relevant sources to reduce high cost of processing keyword search queries over all sources. Aim is to improve the performance of keyword search, without compromising its result quality. In the proposed system, query expansion takes place using correlated, linguistic and semantic features. The goal of keyword expansion is to improve precision and recall. Relation between Keywords and the elements of data takes place by using a keyword-element relationship. Here, two types of search techniques. One is element level search and another is set level search technique. The proposed system uses routing keyword search for queries having many keywords. This improves the performance of keyword search. This way can greatly reduce time and space costs.

The general architecture of the proposed system is elaborated using given figure.

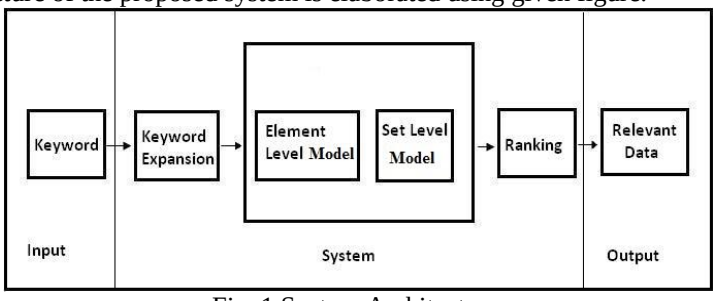

Fig. 1 System Architecture

Basic Blocks:

A. Keyword Expansion
B. Element-level Model
C. Set-level Model
D. Ranking

A. Keyword Expansion

Query Expansion is the process of reformulating a seed query to improve retrieval performance in information retrieval operation. In the context of web search engines, query expansion involves evaluating users input and expanding the search query to match additional documents. The goal of the keyword expansion is to improve precision and recall.

© 2015, IJARCSSE All Rights Reserved

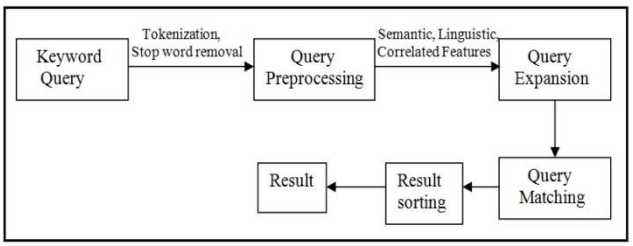

Fig. 2 Query Expansion

Keyword Query: **Input is the keyword query.**

Query Preprocessing:

1) Tokenization: Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. extraction of individual words, ignoring punctuation and case. Here is an example of tokenization:

Input: Friends, Romans, Countrymen, lend me your ears;

Output: Friends Romans Countrymen lend me your ears

Tokens are separated by whitespace characters, such as a space or line break, or by punctuation characters.

2) Stop Word Removal: Removal of common words such as articles and prepositions. i.e. words that appear very often and does not have definite meaning, for example, prepositions and auxiliary words such as the, it, in, etc.

E.g. char text[] = "this is an example of how to use token".

Hence, the output after removal of stop word will be as follows.

Output: this example token

Query Expansion:

Takes place with the help of LOD Cloud. For instance, using DBpedia as knowledge base, the query "Who is married to Barack Obama?" could fail, because the desired property in DBpedia is labeled "spouse" and there is no property labeled "married to". This query expansion is a tried and tested method in web search for tackling the vocabulary problem by adding related words to the search query and thus increase the likelihood that appropriate documents are contained in the result. The query is expanded with features such as synonyms, e.g. "spouse" and "married to" in the example above, or hyponym-hypernym relations, e.g. "red" and "color". Expansion of keyword takes place by semantic and linguistic and correlated features. For the given input k keyword, we define the set of all words related to the keyword k as $X_k = \{x_1, x_2, \ldots x_n\}$. The set X_k is the union of the three sets LE_k, SE_k and CO_k. LE_k is the collection of all words obtained through linguistic features and in the same manner the semantic and correlated features also.

Linguistic features follows:

_ Synonyms: words having same meanings like k.

_ Hyponyms: words represent a specialization of the k.

_ Hypernym: words represent a generalization of the k.

The set SE shows all words semantically from the input k using Linked Data. Semantic features are as the following semantic relations:

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

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

_ property equivalence or class: representing classes or properties providing related de- scriptions for the input resource using owl:equivalentProperty , owl:equivalentClass

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

_ Subclass or -property: representing all sub resources of the input resource ri by following the rdfs:subClassOf / rdfs:subPropertyOf property paths ending with the resource input.

_ broader concept: representing broader concepts related to the input resource ri using the SKOS vocabulary properties skos:broader , skos:broadMatch

_ narrower concepts: representing narrower concepts related to the input resource ri using skos:narrower and skos:narrowMatch.

_ related concepts: representing related concepts to the input resource ri using, skos:mappingRelatioand skos:closeMatch , skos:exactMatch.

Correlation is a mutual relationship or connection between two or more things. Correlated words are having mutual relationship or connection between them.

The set of all related words of k is defined as $X_k = LE_k \cup SE_k \cup CO_k$.

Query Matching:

Identifiers are an important source of domain information. In this step, we match the expanded queries with the identifiers. For each expanded query eQuery and each identifier id we compute the following similarity score:

$$Sim(eQuery, id) = \frac{|\ eQuery \cap id\ |}{max(|\ eQuery\ |, |\ id\ |)}$$

Result Sorting:

Sorting of all results generated by using semantic, linguistic and correlated features.

© 2015, IJARCSSE All Rights Reserved

Result:
List of results relevant to the input keyword query.

B. Element Level Model
In this model, we mainly concentrate on IR technique of data retrieval. This technique allows users to search unstructured information, and do not need users to understand any database schemas. For Element level search, we used LSI(Latent Semantic Indexing)technique.LSI is an indexing and retrieval method that uses a mathematical technique called single value decomposition (SVD)to identify patterns in the relationship between terms and concepts contained in an unstructured collection of text. It called LSI because of its ability to correlate semantically related terms that are latent in a collection of text. LSI begins by constructing Term Document Matrix, to identify occurrence of the m unique terms within a collection of n documents. Each term represent by a row and each document is represented by column.
Algorithm for LSI:
To perform Latent Semantic Indexing on a group of documents, we perform the following steps:
1. First, convert each document in index into a vector of word occurrences. The number of dimensions our vector exists in is equal to the number of unique words in the entire document set. Most document vectors will have large empty patches, some will be quite full. It is recommended that common words (e.g., "this", "him", "that", "the") are removed.
2. Next, scale each vector so that every term reflects the frequency of its occurrence in context.
3. Next, combine these column vectors into a large term-document matrix. Rows represent terms, columns represent documents.
4. Perform Singular Value Decomposition on the term-document matrix. This will result in three matrices commonly called U, S and V. S is of particular interest, it is a diagonal matrix of singular values for our document system.
5. Set all but the k highest singular values to 0. k is a parameter that needs to be tuned based on space. Very low values of k are very lossy, and net poor results. But very high values of k do not change the results much from simple vector search. This makes a new matrix, S'.
6. Recombine the terms to form the original matrix (i.e., U * S' * V (t) = M' where (t) signifies transpose).
7. Break this reduced rank term-document matrix back into column vectors. Associate these with their corresponding documents.
8. Now we have a Latent Semantic Index.

C. Set-level Model
Set-level model perform set level search, which captures information about group of elements. 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.

D. Ranking
Ranking depends on user searching a product. Give the rank number of products depending upon how many time user searching a particular product.

IV. EXPERIMENTAL RESULTS

Data set is nothing but the collection of data used to test the results. Data for this project is fetched online from DBPedia, Freebase and LOD (Linked Open Data) Cloud.

A. Precision:
Precision is the ratio of the number of relevant records retrieved to the total number of
Irrelevant and relevant records retrieved. It is usually expressed as a percentage.

$$\text{Precision} = \frac{\text{Number of relevant results retrieved}}{\text{Number of results retrieved}}$$

The following graph shows precision of direct search (no query expansion) and our approach on 10 search tasks.

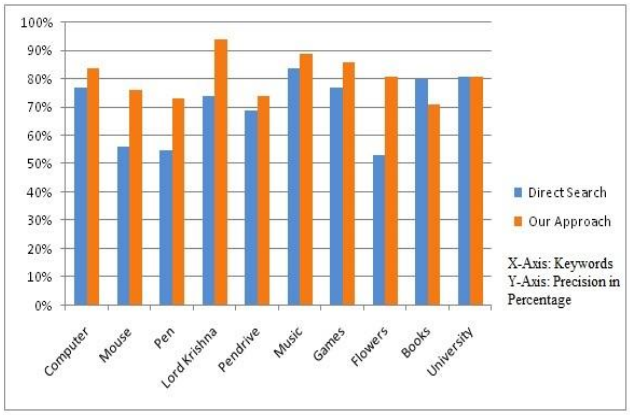

Fig. 3 Precision of direct search (no query expansion) and our approach

© 2015, IJARCSSE All Rights Reserved

Table I

Sr. No.	Keyword Query	Precision of Direct Search	Precision of Our Approach
1	Computer	77%	84%
2	Mouse	56%	76%
3	Pen	55%	73%
4	Lord Krishna	74%	94%
5	Pendrive	69%	74%
6	Music	84%	89%
7	Games	77%	86%
8	Flowers	53%	81%
9	Books	80%	71%
10	University	81%	81%

B. Recall

Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

$$\text{Recall} = \frac{\text{Number of relevant results retrieved}}{\text{Number of relevant docs}}$$

The following graph shows recall of direct search (no query expansion) and our approach on 10 search tasks.

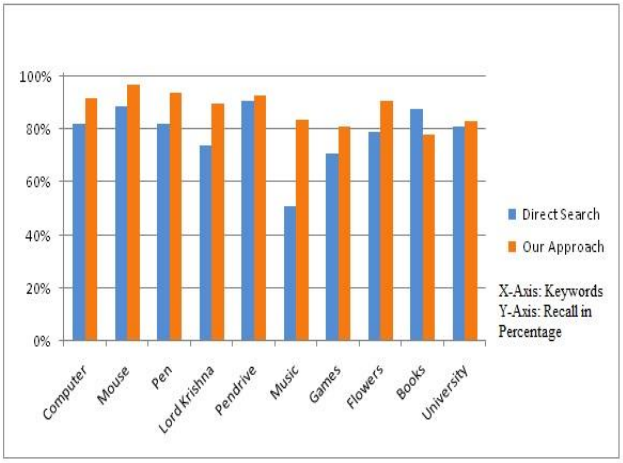

Fig. 4 Recall of direct search (no query expansion) and our approach

Table II

Sr. No.	Keyword Query	Recall of Direct Search	Recall of Our Approach
1	Computer	82%	92%
2	Mouse	89%	97%
3	Pen	82%	94%
4	Lord Krishna	74%	90%
5	Pendrive	91%	93%
6	Music	51%	84%
7	Games	71%	81%
8	Flowers	79%	91%
9	Books	88%	78%
10	University	81%	83%

C. Time Complexity

The following graph shows results on the basis of time it shows the how much time is required for existing system and how much time proposed system takes for Keyword Search.

© 2015, IJARCSSE All Rights Reserved

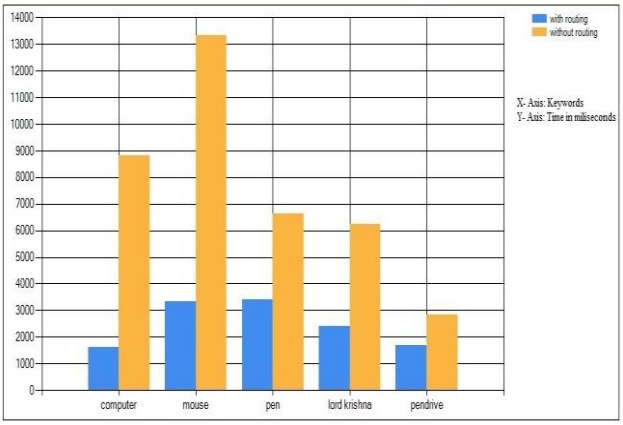

Fig.5 Time comparison between Existing and proposed system

Table III

Sr. No.	Keyword Query	Time required for keyword search with routing (ms)	Time required for keyword search without routing (ms)
1	Computer	1613	8837
2	Mouse	3348	13330
3	Pen	3407	6628
4	Lord Krishna	2414	6235
5	Pendrive	1702	2847

Above tables shows that the proposed system is more effective than existing one as it requires less time to get desired query result. Precision and recall values are also improved.

V. CONCLUSIONS

Efficient keyword query routing system helps to improve the performance of keyword search, without compromising quality of result. Investigate the trouble of keyword query routing for keyword search over a large number of structured and Linked Data sources. Routing keywords only to appropriate sources can reduce the high cost of searching for structured results that span multiple sources. System produced results in minimum time, while not compromising too much on quality. The proposed system uses routing keyword search for the queries having multiple keywords. Here, comparative study between proposed system and existing system is studied. From the above results we conclude that, without routing the keyword search is problematic when the number of keywords is large. In Future, our aim is to route audio video multimedia data and reduce searching time for multimedia data.

REFERENCES

[1] Thanh Tran and Lei Zhang, "Keyword Query Routing" ,IEEE Transactions, VOL.26, NO.2, February 2014.
[2] T. Berners-Lee, "Linked Data Design Issues", 2009;www.w3.org/DesignIssues/LinkedData.html
[3] 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.
[4] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional Expansion for Keyword Search on Graph Databases", Proc. 31st Intl Conf. Very Large Data Bases (VLDB), pp. 505-516, 2005.
[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 Intl Conf. Data Eng. (ICDE), pp. 836845, 2007.
[6] M. Sayyadian, H. LeKhac, A. Doan, and L. Gravano, "Efficient Keyword Search Across Heterogeneous Relational Databases", Proc. IEEE 23rd Intl Conf. Data Eng. (ICDE), pp. 346-355, 2007.
[7] T. Tran, H. Wang, and P. Haase, "Hermes: Data Web Search on a Pay-as-You-Go Integration Infrastructure", J. Web Semantics, vol. 7, no. 3, pp. 189-203, 2009.
[8] 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.
[9] Jianhua Feng, Guoliang Li and Jianyong Wang, "Finding Top-k answers in keyword search over relational databases using tuple units", IEEE transactions, VOL. 23 NO. 12, December 2011.
[10] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou, "Ease: An Effective 3-in-1 Keyword Search Method for Unstructured, SemiStructured and Structured Data", Proc. ACM SIGMOD Conf., pp. 903-914, 2008.
[11] R. Goldman and J. Widom, "DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases",Proc. 23rd Intl Conf. Very Large Data Bases (VLDB), pp. 436-445, 1997.
[12] K. Collins- Thompson, "Reducing the risk of query expansion via robust constrained optimization". In CIKM. ACM, 2009.

© 2015, IJARCSSE All Rights Reserved

[13] H. Deng, G. C. Runger, and E. Tuv. "Bias of importance measures for multi-valued attributes and solutions". In ICANN (2), volume 6792, pages 293300. Springer, 2011.

[14] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling. "Feature selection using linear classifier weights: interaction with classification models". In Proceedings of the 27th Annual International ACM SIGIRConference SIGIR2004. ACM, 2004.

[15] Saeedeh Shekarpour, Jens Lehmann, and Sren Auer, "Keyword Query Expansion on Linked Data Using Linguistic and Semantic Features", IEEE Seventh International Conference on Semantic Computing, 2013.

[16] 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.

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

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

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

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

[21] 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.

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

[23] G. Ladwig and T. Tran, "Index Structures and Top-K Join Algorithms for Native Keyword Search Databases", Proc. 20th ACM Intl Conf. Information and Knowledge Management (CIKM), pp. 1505-1514, 2011.

© 2015, IJARCSSE All Rights Reserved