International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Keyword Query Search Using KERG

Chaitali S. Chaudhari[1], **M. M. Naoghare**[2]

[1, 2]Department of Computer Engineering, Sir Visvesvaraya Institute of Technology, Nashik, Maharashtra, India

Abstract: ***Searching Keyword is a spontaneous method for searching linked data sources on the web. We suggest to route keywords only to related sources to reduce the cost of searching keyword queries over all sources. The new method is proposed for computing top-k routing plans .We employ a keyword-element relationship graph (KERG) that shows relationships between keywords and elements . A multilevel scoring mechanism is proposed for calculating the relevance of routing plans which depends on scores at the level of keywords, data elements, element sets and subgraphs that connect all this elements.***

Keywords: keyword search, query keyword, routing keyword query, RDF, graph –structured data

1. Introduction

The web is a collection of textual documents and also a web of connected data sources .It is somewhat difficult for some web users to extract this web data by means of structured queries using various languages. To this end, searching keyword has proven to be intuitive. As inspite of such structured queries, no special knowledge of the query language, the schema or the underlying data are needed. Query processing over graph-structured data is growing number of applications. A top-k keyword query search on a graph finds the top k answers, where each answer is a substructure of the graph containing all query keywords in database search, solutions have been given, which gives a keyword search, retrieve the most applicable structured results or simply, select the single most relevant databases. However, these approaches are single-source solutions. They are not directly relevant to the web of Linked Data, where results are not tied by a single source but might complete several Linked Data sources. As linked data contains hundreds of sources which further contains billions of RDF triples, which are connected by many links. While various links can be established, the one which is frequently published are sameAs links, which shows that two RDF resources represent the same world object or it is related to each other. As opposed to the source selection problem, which is focusing on computing the most relevant sources, the problem here is to calculate the most relevant combinations of sources. The goal is to produce routing plans, which can be used to compute results from multiple sources.

The intention is to solve the problem of routing keyword query search over a large number of structure and Linked Data sources. Routing the keywords only to relevant sources can diminish the high cost of searching for structured results that span multiple sources. Existing work uses keyword relationships (KR) individually for single databases. We shows the relationships between keywords and data elements. They are build for the entire collection of linked sources, and then grouped together as elements of a compact summary known as the set-level Keyword-Element Relationship Graph (KERG).

The approach will show that when routing is applied to an existing searching keyword system to prune sources, substantial performance gain can be achieved.

2. Literature Survey

V. Hristidis, L. Gravano, and Y. Papakonstantinou, propose *G-KS*, a method for selecting the top-*K* candidates based on their potential to contain results for a given query. *G-KS* summarizes each database by a keyword relationship graph, where nodes represent terms and edges describe relationships between them. Keyword relationship graphs are utilized for computing the similarity between each database and a KS query, so that, during query processing, only the most promising databases are searched.

Y. Luo, X. Lin, W. Wang, and X. Zhou, proposed the study, of the effectiveness and the efficiency issues of answering top-k keyword query in relational database systems. Here, proposed a new ranking formula by adapting existing IR techniques based on a natural notion of virtual document. Compared with previous approaches, the new ranking method is simple and effective, and agrees with human perceptions. Here, also study efficient query processing methods for the new ranking method, and propose algorithms that have minimal accesses to the database.

Hao He, Haixun Wang, Haixun Wang, Philip S. Yu, proposed BLINKS, a bi-level indexing and query processing scheme for top-*k* keyword search on graphs. BLINKS performs a search strategy with provable performance bounds, while additionally exploiting a bi-level index for pruning and accelerating the search. To diminish the index space, BLINKS partitions a data graph into blocks: The bi-level index stores summary information at the block level to initiate and guide search among blocks, and more detailed information for each block to accelerate search within blocks. BLINKS offers orders-of-magnitude performance improvement over existing approaches.

B. Ding, J.X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin, proposed It is widely realized that the integration of database and information retrieval techniques will provide users with a wide range of high quality services. In this paper, we study processing an l keyword query, p1; p2…. pl, against a relational database which can be modeled as a weighted graph, G(V;E). Here V is a set of nodes (tuples) and E is a set of edges representing foreign key references between tuples. Let Vi _ V be a set of nodes that contain the keyword pi. We study finding top-k minimum cost connected trees that contain at least one node in every subset Vi , and denote our problem as GST-k. When k = 1, it is known as a minimum cost group Steiner tree problem which is NPComplete.

3. Proposed System

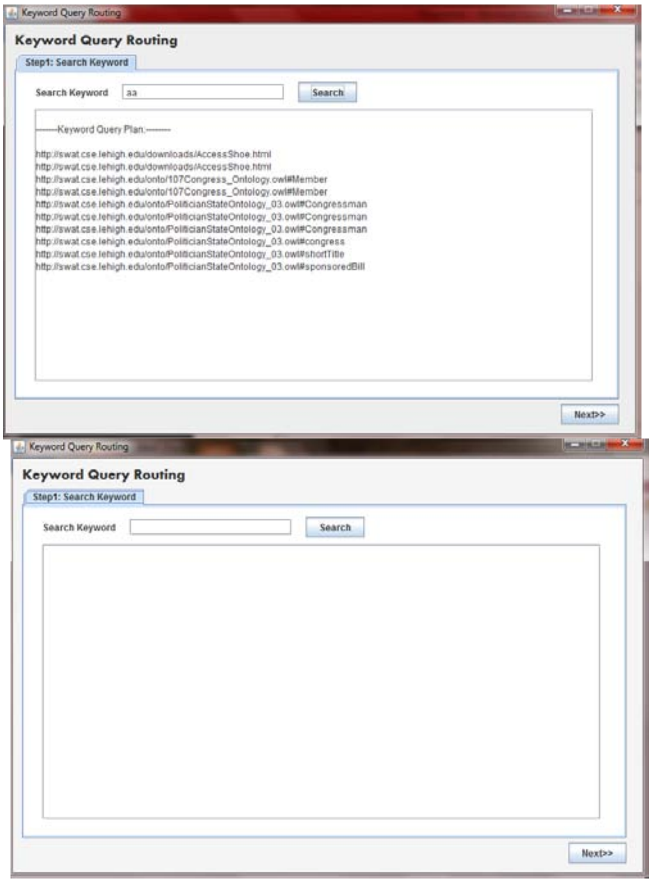
We adopt a graph based model to characterize individual data sources. In this model, we distinguish between an element level data graph showing relationships between individual data elements and a set-level data graph which captures information about group of elements. It mainly consists of following models:

- *Element Level Data Graph*: This model collects Resource Description Framework data where entities stand for some RDF resources, data values stand for Resource Description Framework literals , and relations and attributes correspond to RDF triples.
- *Set-level Data Graph*: This set-level graph actually captures a part of the Linked Data schema on the web that are 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.
- *Routing Keyword Query:* The problem of routing keyword query is to find the top k routing keyword plans based on their relation to a query. A relevant plan should correspond to the information expected by the user.
- The proposed system is divided in following modules:
- *Load Dataset*: The data which we are going to use are drawn from data sets which is prepared from Billion Triple Challenge (BTC).BTC data set are split into chunks of 10M statements each. Normally, this chunk of data contains less than 3K RDF triples.
- *Convert to N-Quad Triplet*: The chunks are converted in RDF triplet form .
- *Index Structure*: During the index building process ,we counted the number of keyword relationship i.e all pairs of keywords that are connected over a maximum distance.
- *KERG*: From the index structure, the KERG counts the element level keyword element relationships. Here, we consider a no. of relationship in Keyword Element Relationship Graph , it is built to capture all valid plans, the scoring mechanism is designed to focus on relevant plans.
- *Routing Plan*: The Routing Plan is computed by using the ComputeRoutingPlan . For obtaining the complete routing plan it is necessary to join all the relevant keywords.

4. Result

4.1 Keyword Search

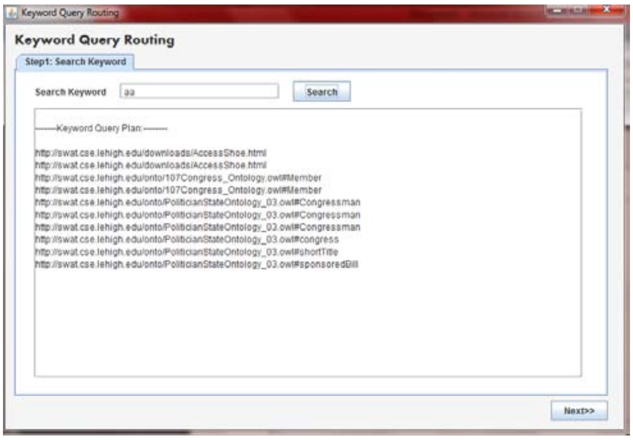
A keyword query is processed by mapping keywords to elements of the database (called keyword elements). Then, using the schema, valid join sequences are derived, which are then employed to join ("connect") the computed keyword elements to form so-called candidate networks representing possible results to the keyword query.



```
--------Keyword Query Plan:--------

http://swat.cse.lehigh.edu/downloads/AccessShoe.html
http://swat.cse.lehigh.edu/downloads/AccessShoe.html
http://swat.cse.lehigh.edu/onto/107Congress_Ontology.owl#Member
http://swat.cse.lehigh.edu/onto/107Congress_Ontology.owl#Member
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#Congressman
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#Congressman
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#Congressman
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#congress
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#shortTitle
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#sponsoredBill
```

Figure 1: Keyword to be searched



```
--------Keyword Query Plan:--------

http://swat.cse.lehigh.edu/downloads/AccessShoe.html
http://swat.cse.lehigh.edu/downloads/AccessShoe.html
http://swat.cse.lehigh.edu/onto/107Congress_Ontology.owl#Member
http://swat.cse.lehigh.edu/onto/107Congress_Ontology.owl#Member
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#Congressman
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#Congressman
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#Congressman
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#congress
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#shortTitle
http://swat.cse.lehigh.edu/onto/PoliticianStateOntology_03.owl#sponsoredBill
```

Figure 2: Relevant Result after searching

Structured results are computed by exploring the underlying data graph. The goal is to find structures in the data called Steiner trees (Steiner graphs in general), which connect keyword element When the keyword is entered for searching ,if that keyword is present in keyword table it will directly display all the keyword routing plans and if it is not present it will show that keyword is not present.

5. Conclusion

We presented a solution to the problem of routing keyword query search. Based on showing a view of the search space as a multilevel inter-relationship graph, it proposes a summary model that groups keyword and element relationships at the level of sets, and developed a multilevel ranking scheme to incorporate relevancy at different dimensions.

References

[1] 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.

[2] 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.

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

[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] G. Ladwig and T. Tran, “Index Structures and Top-K Join Algorithms for Native Keyword Search Databases,” Proc. 20th ACM Int’l Conf. Information and Knowledge Management (CIKM),pp. 1505-1514, 2011.

[6] 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.

[7] S. Chaudhuri and G. Das, “Keyword Querying and Ranking in Databases,” Proceedings of the VLDB Endowment, vol. 2, pp. 1658–1659, August 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687553. 1687622

[8] Y. Chen, W. Wang, Z. Liu, and X. Lin, “Keyword Search on Structured and Semi-Structured Data,” in Proceedings of the 35th SIGMOD International

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] R. Goldman and J. Widom, “DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases,” Proc. 23rd Int’l Conf. Very Large Data Bases (VLDB), pp. 436-445, 1997.

[22] G. Ladwig and T. Tran, “Index Structures and Top-K Join Algorithms for Native Keyword Search Databases,” Proc. 20th ACM Int’l Conf. Information and Knowledge Management (CIKM), pp. 1505-1514, 2011.

Author Profile

Ms. Chaitali S. Chaudhari has completed her B.E in Computer Engineering from Pune University and currently pursuing Master of Engineering from SVIT Chincholi, Nashik, India

Prof. M. M. Naoghare has completed her B.E in Computer Engineering from College of Engineering, Badnera, Amravati University and M.E in Computer Science & Engineering from P.R.M.I.T & R, Badnera, Amravati. She is presently working as an Associate Professor in SVIT Chincholi, Nashik, India