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

Keyword Query Routing using Relationship Graph

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

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

Abstract: ***Keyword searching is an best method for searching linked data sources on the web. We, propose to route keywords only to relevant sources to reduce the processing cost of keyword queries search over all sources of web. We propose a best method for computing top-keyword routing plans based on their potentials to contain results for a given keyword query. We find a keyword-element relationship summary that shows a relationships between keywords and the data elements showing 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 subgraphs that connect these elements. Further, we show routing greatly helps to improve the performance of keyword search, without compromising its result quality.***

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

1. Introduction

The web is a mixture of web of connected data and textual documents .It is somehow difficult for some users to extract this web data by means of structured queries or using other languages. At this end, searching keyword has proven to be best. 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. The 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 provide single-source solutions. They are not directly relevant to the web of Linked Data, where results are not binded 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 intention is to produce routing plans, which can be used to compute results from multiple sources.

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.

3. Proposed System

3.1 Keyword Search

A keyword query is processed by mapping keywords to elements of the database (called keyword elements).

3.2 Triple Table

The data which we used 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. This chunks are converted in RDF triplet form and this triplet data is shown in triple table ,which specify the subject,object and property of data.

3.3 Database Table

This table contains all the data which we have taken as in database for retrieving purpose.

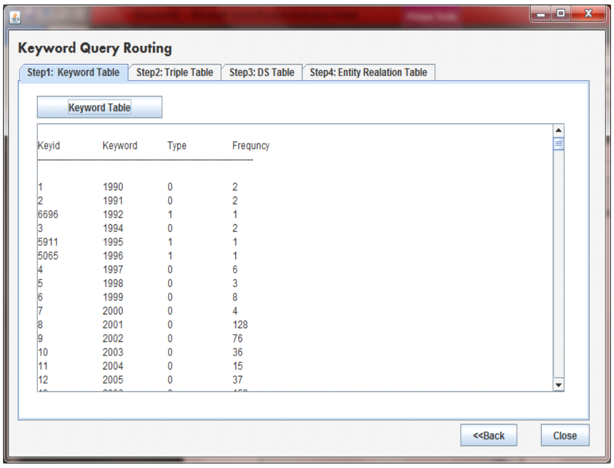
3.4 Entity Relationship Table

E-R table specifies the mapping of data elements from database tables. Whenever web users enters the keyword for searching from database, the mapping with entity relationship table is done and the most relevant result is provided.

4. Result

4.1 Keyword Table

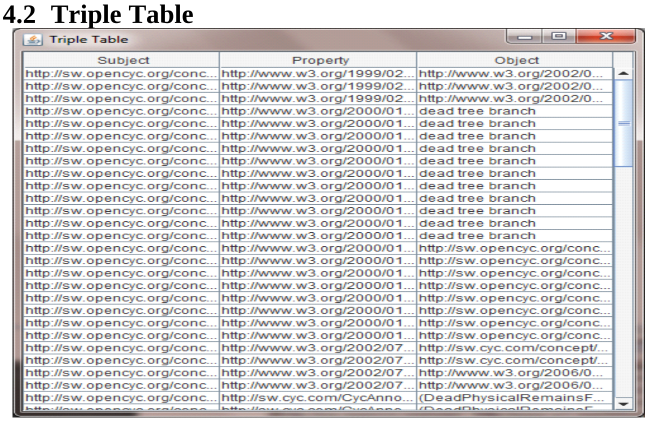
Keyword table shows a keyword id which specifies a id of particular keyword , keyword, type and frequency of its occurrences in database.



Keyid	Keyword	Type	Frequncy
1	1990	0	2
2	1991	0	2
6696	1992	1	1
3	1994	0	2
5911	1995	1	1
5065	1996	1	1
4	1997	0	6
5	1998	0	3
6	1999	0	8
7	2000	0	4
8	2001	0	128
9	2002	0	76
10	2003	0	36
11	2004	0	15
12	2005	0	37

Figure: Keyword Table

4.2 Triple Table



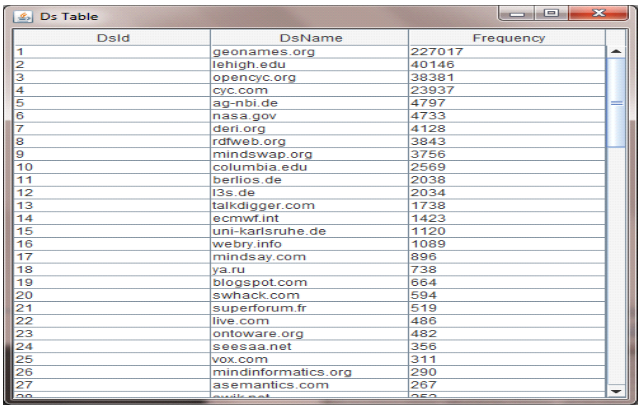
Subject	Property	Object
http://sw.opencyc.org/conc...	http://www.w3.org/1999/02...	http://www.w3.org/2002/0...
http://sw.opencyc.org/conc...	http://www.w3.org/1999/02...	http://www.w3.org/2002/0...
http://sw.opencyc.org/conc...	http://www.w3.org/1999/02...	http://www.w3.org/2002/0...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	dead tree branch
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2000/01...	http://sw.opencyc.org/conc...
http://sw.opencyc.org/conc...	http://www.w3.org/2002/07...	http://sw.cyc.com/concept/...
http://sw.opencyc.org/conc...	http://www.w3.org/2002/07...	http://sw.cyc.com/concept/...
http://sw.opencyc.org/conc...	http://www.w3.org/2002/07...	http://www.w3.org/2006/0...
http://sw.opencyc.org/conc...	http://www.w3.org/2002/07...	http://www.w3.org/2006/0...
http://sw.opencyc.org/conc...	http://sw.cyc.com/CycAnno...	(DeadPhysicalRemainsF...

Figure: Triple Table

Triple table shows a subject of particular site ,its object and its property which its belongs.

4.3 Database Table

It specifies database id, database name and its frequency of its occurrences in database.

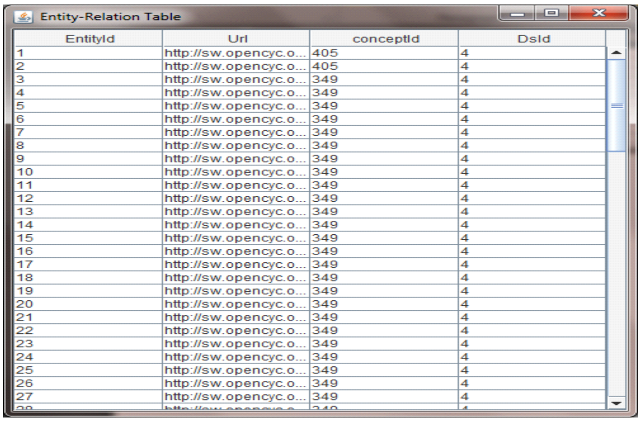


DsId	DsName	Frequency
1	geonames.org	227017
2	lehigh.edu	40146
3	opencyc.org	38381
4	cyc.com	23937
5	ag-nbi.de	4797
6	nasa.gov	4733
7	deri.org	4128
8	rdfweb.org	3843
9	mindswap.org	3756
10	columbia.edu	2569
11	berlios.de	2038
12	l3s.de	2034
13	talkdigger.com	1738
14	ecmwf.int	1423
15	uni-karlsruhe.de	1120
16	webry.info	1089
17	mindsay.com	896
18	ya.ru	738
19	blogspot.com	664
20	swhack.com	594
21	superforum.fr	519
22	live.com	486
23	ontoware.org	482
24	seesaa.net	356
25	vox.com	311
26	mindinformatics.org	290
27	asemantics.com	267

Figure: Database Table

4.4 Entity-Relationship Table

It specifies entity id,its url showing result,concept id and ds id.



EntityId	Url	conceptId	DsId
1	http://sw.opencyc.o...	405	4
2	http://sw.opencyc.o...	405	4
3	http://sw.opencyc.o...	349	4
4	http://sw.opencyc.o...	349	4
5	http://sw.opencyc.o...	349	4
6	http://sw.opencyc.o...	349	4
7	http://sw.opencyc.o...	349	4
8	http://sw.opencyc.o...	349	4
9	http://sw.opencyc.o...	349	4
10	http://sw.opencyc.o...	349	4
11	http://sw.opencyc.o...	349	4
12	http://sw.opencyc.o...	349	4
13	http://sw.opencyc.o...	349	4
14	http://sw.opencyc.o...	349	4
15	http://sw.opencyc.o...	349	4
16	http://sw.opencyc.o...	349	4
17	http://sw.opencyc.o...	349	4
18	http://sw.opencyc.o...	349	4
19	http://sw.opencyc.o...	349	4
20	http://sw.opencyc.o...	349	4
21	http://sw.opencyc.o...	349	4
22	http://sw.opencyc.o...	349	4
23	http://sw.opencyc.o...	349	4
24	http://sw.opencyc.o...	349	4
25	http://sw.opencyc.o...	349	4
26	http://sw.opencyc.o...	349	4
27	http://sw.opencyc.o...	349	4

Figure: Entity Relationship Table

5. Conclusion

We have presented a solution to the problem of keyword query routing. Based on modeling the search space as a multilevel inter-relationship graph, we proposed a model that groups keyword and element relationships at the level of sets, and developed a multilevel ranking scheme to incorporate relevance 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